USING PRIOR KNOWLEDGE IN THE DESIGN OF CLASSIFIERS

A Dissertation

by

MOHAMMAD SHAHROKH ESFAHANI

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Edward Russell Dougherty
Co-Chair of Committee,	Aniruddha Datta
Committee Members,	Byung-Jun Yoon
	Ivan Ivanov
Head of Department,	Chanan Singh

May 2014

Major Subject: Electrical Engineering

ABSTRACT

Small samples are commonplace in genomic/proteomic classification, the result being inadequate classifier design and poor error estimation. A promising approach to alleviate the problem is the use of prior knowledge. On the other hand, it is known that a huge amount of information is encoded and represented by biological signaling pathways. This dissertation is concerned with the problem of classifier design by utilizing both the available prior knowledge and training data. Specifically, this dissertation utilizes the concrete notion of regularization in signal processing and statistics to combine prior knowledge with different data-based or data-ignorant criteria.

In the first part, we address optimal discrete classification where prior knowledge is restricted to an uncertainty class of feature distributions absent a prior distribution on the uncertainty class, a problem that arises directly for biological classification using pathway information: labeling future observations obtained in the steady state by utilizing both the available prior knowledge and the training data. An optimization-based paradigm for utilizing prior knowledge is proposed to design better performing classifiers when sample sizes are limited. We derive approximate expressions for the first and second moments of the true error rate of the proposed classifier under the assumption of two widely used models for the uncertainty classes: ε-contamination and p-point classes. We examine the proposed paradigm on networks containing NF-κB pathways, where it shows significant improvement compared to data-driven methods.

In the second part of this dissertation, we focus on Bayesian classification. Although the problem of designing the optimal Bayesian classifier, assuming some

known prior distributions, has been fully addressed, a critical issue still remains: how to incorporate biological knowledge into the prior distribution. For genomic/proteomic, the most common kind of knowledge is in the form of signaling pathways. Thus, it behooves us to find methods of transforming pathway knowledge into knowledge of the feature-label distribution governing the classification problem. In order to incorporate the available prior knowledge, the interactions in the pathways are first quantified from a Bayesian perspective. Then, we address the problem of prior probability construction by proposing a series of optimization paradigms that utilize the incomplete prior information contained in pathways (both topological and regulatory). The optimization paradigms are derived for both Gaussian case with Normal-inverse-Wishart prior and discrete classification with Dirichlet prior.

Simulation results, using both synthetic and real pathways, show that the proposed paradigms yield improved classifiers that outperform traditional classifiers which use only training data.

DEDICATION

To my parents, Noushin and Mehdi,

my sisters Negar and Marjan,

my brother Majid

and

To my love! My sweetheart wife! Saeideh

ACKNOWLEDGEMENTS

First, I would like to thank my parents, Noushin and Mehdi, for their unconditional love, patience and everyday-support. Without them, I would never have been able to achieve anything, put aside the PhD. I am deeply thankful of my father, Mehdi whose expertise and interest in engineering and math have always influenced my path and directed me to be a "good" engineer.

Next, I would like to thank my lovely wife, Saeideh, for all of her sacrifices during my PhD. Her patience, love, and encouraging attitude have been always driving me to seek for a better quality of work. Moreover, I thank my brother, Majid, my elder sister Marjan, and my little sister, Negar, who has been my real friend, for their love.

I would like to express my deepest gratitude to my advisor, professor Edward Dougherty, who made my dissertation possible. I thank him for his steady guidance, inspirational discussions, and his dedication to high-quality research through my doctorate program at Texas A&M University. The great experiences of working with him will definitely benefit the rest of my career. I would like to thank the co-chair of my committee, Dr. Aniruddha Datta by whom I started to know Genomic Signal Processing during my first year at Texas A&M University. I would also like to thank Dr. Byung-Jun Yoon and Dr. Ivan Ivanov for serving on my committee and for their constructive collaborations and advice.

I thank all students and researchers, past and current, at the GSP lab. Furthermore, I would like to thank all my friends in College Station, specially Behnam Tarrahi, Amin Hassanzadeh, Dr. Amin Rasekh, and Esmaeil Atashpaz-Gargari whose presence and helps have made the life easier and more joyful during my PhD studies.

NOMENCLATURE

ML	Maximum-Likelihood
RML	Regularized Maximum-Likelihood
OBC	Optimal Bayesian Classifier
REML	Regularized Expected Mean-Log-Likelihood
RMEP	Regularized Maximum Entropy Prior
RMDIP	Regularized Maximal Data Information Prior
$\ln(.)$ or $\log(.)$	Natural logarithm of $x \in \mathbb{R}_+$
$\lvert\boldsymbol{A}\rvert$	Determinant of the matrix $\boldsymbol{A}$
$f(X\lvert Y = y)$	Conditional density of X given $Y = y$
$E_x[g(x)]$, or $\mathrm{E}_x[g(x)]$	Expectation of $g(x)$ with respect to x
$\Pr(X\lvert Y = y)$	Conditional probability of X given $Y = y$
$\mathcal{B}in(n, p)$	Binomial distribution
$\mathcal{T}rin(n, p_1, p_2)$	Trinomial distribution
$\mathcal{D}ir(\boldsymbol{\alpha})$	Dirichlet distribution parametrized by $\boldsymbol{\alpha}$
$\mathcal{B}eta(\alpha, \beta)$	Beta distribution parametrized by α and β
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Multivariate normal distribution
$\Gamma(.)$	Gamma function
$\psi(.)$	Digamma function

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION *

A pattern recognition problem is on predicting a random variable, called interchangeably a *label*, *class* or *response variable*, from some other statistically related random variable (vector), called *feature vector* or *prediction vector*. Whenever patterns are distinguished using some intermediate functional derived from existing *labeled observations*, called *training data*, pattern recognition is interchangeably known as *supervised learning*. Depending on the continuity or discreteness of the label, the problem respectively falls into *regression* or *classification*.

1.1 Classification Problem

Let $\boldsymbol{x} \in \mathcal{X}$, where $\mathcal{X} = \mathbb{R}^p$ in the continuous setting or a finite set of numbers (bins) $\mathcal{X} = \{1, \ldots, b\}$ in the discrete scenario, be an event from the sample space of dimension p or bin-size b, coming from one of the subgroups of the overall population. In the *binary classification* problem, the population is divided into two subgroups: $y \in \{0, 1\}$. Then, the problem of *classification* is to design a *classifier* $\psi(\boldsymbol{x}) : \mathcal{X} \to \{0, 1\}$. We will refer to this case, simply as classification throughout this proposal. Further, the relationship between the sample and the subgroup label, y is fully characterized by their joint distribution $f(\boldsymbol{x}, y)$. Hence, the subgroups can be described by probability density functions, called *class (label)-conditional densities* $f(\boldsymbol{x} = X|Y = y);\ y \in \{0, 1\}$.

*Parts of this section are reprinted with permission from M. Shahrokh Esfahani, J. Knight, A. Zollanvari, B.-J. Yoon, and E. R. Dougherty, "Classifier design given an uncertainty class of feature distributions via regularized maximum likelihood and the incorporation of biological pathway knowledge in steady-state phenotype classification," *Pattern Recognition*, vol. 46, no. 10, pp. 2783–2797, 2013.

1.1.1 Complete Knowledge of Underlying Distributions

In this case, it is assumed that the complete knowledge about class-conditional densities and class prior probabilities $c = \Pr(Y = 0)$ (similarly $1 - c = \Pr(Y = 1)$) is known. Using the Bayes theorem, fixing the label $Y = y$, the *likelihood* associated with the $\boldsymbol{x}$ is computed: $f(\boldsymbol{x} = X|Y = y)$, from which the *Bayes (optimal) classifier* (optimal) is defined as follows

$$\psi(\boldsymbol{x} = X) = \begin{cases} 1, \text{ if } \frac{f(\boldsymbol{x}=X|Y=1)}{f(\boldsymbol{x}=X|Y=0)} \geq \frac{c}{1-c} \\ 0, \text{ if } \frac{f(\boldsymbol{x}=X|Y=1)}{f(\boldsymbol{x}=X|Y=0)} < \frac{c}{1-c} \end{cases} \tag{1.1}$$

It should be noticed that in the case of a tie, i.e. $\frac{f(\boldsymbol{x}=X|Y=1)}{f(\boldsymbol{x}=X|Y=0)} = \frac{c}{1-c}$, the class can be either assigned as in (1.1) (arbitrarily) or by randomization (with some predetermined probability). The ratio $\frac{f(\boldsymbol{x}=X|Y=1)}{f(\boldsymbol{x}=X|Y=0)}$, is sometimes called likelihood ratio. Moreover, taking the logarithm from both sides of the inequalities in (1.1), the optimal classifier can be rewritten as follows

$$\psi(\boldsymbol{x} = X) = \begin{cases} 1, \text{ if } \log f(\boldsymbol{x} = X|Y = 1) - \log f(\boldsymbol{x} = X|Y = 0) \geq \log \frac{c}{1-c} \\ 0, \text{ if } \log f(\boldsymbol{x} = X|Y = 1) - \log f(\boldsymbol{x} = X|Y = 0) < \log \frac{c}{1-c}, \end{cases} \tag{1.2}$$

in which, the terms $\log f(\boldsymbol{x} = X|Y = y);\ y \in \{0, 1\}$ are called *log-likelihood* functions.

1.1.2 Classification Using Training Data

The assumption of having complete knowledge of class-conditional densities is not in fact a realistic one. In practice, one would have only access to a limited number of training samples by which a classifier is *trained*. The training samples are usually

denoted by S_n, embedding the sample size, n in the notation. A *classification rule* is defined as a *mapping* from the training samples to a classifier: $\Psi : S_n \rightarrow \psi_n$, or more precisely:

$$\Psi : [\mathcal{X} \times \{0, 1\}]^n \rightarrow \psi_n.$$

In that regard, a classifier is the final product of a training process. The training process is fully characterized by the classification rule. Associated with each of these mappings or functions, a *prediction error*, or simply an *error*, is defined . The most natural definition of the error for a designed classifier is its misclassification rate, namely *true error*: $\epsilon_n = \Pr(\psi_n(X) \neq Y) = E[|Y - \psi(X)|]$. The defined error quantifies the rate of misclassification of future observations X, when the label is assigned using the function ψ. Evaluating a classification rule for a feature-label distribution $f(\boldsymbol{x}|Y = y)$ and class prior probability c, an expectation is taken with respect to the training samples, S_n, each of which leads to different classifiers. In this case, the error is called *expected true error* defined as follows

$$E_{S_n}[\epsilon_n] = E_{S_n}\Big[E[|Y - \psi(X)||S_n]\Big].$$

The main difference between true error and expected true error lies in that the former deals with the actual error of "one designed classifier" obtained by applying a certain classification rule to a fixed training sample S_n. On the other hand, the latter yields the expected performance of a rule on the whole feature-label distribution, rather than one realization of that. Throughout this dissertation, we often deal with expected true error to remove the dependency (randomness) originated from a single realization of S_n.

The classification rules are categorized from modeling perspective into "model-

based" and "model-free" rules. In model-free rules, no specific model for the data generating process is assumed. In particular, no assumptions are made regarding the feature-label distribution (population) from which the sample data have been drawn, and instead a classifier is designed utilizing training data. Examples include $k-$nearest neighbor, support vector machine (SVM), or decision tree. In model-based methods instead, first a model is selected, called *model selection* stage, in which the underlying distributions are usually modeled (parameterized). Then, the assumed model is *trained* via training data. This process is usually done through *model parameter estimation*. The estimated parameters are then plugged in the model yielding the classifier. Examples include linear discriminant analysis (optimal when the underlying model is Gaussian with identical covariance matrices for two classes) or quadratic discriminant analysis (optimal when the underlying model is Gaussian with unequal covaraince matrices). The main advantage of model-based methods is in their tractability for evaluating their performance, e.g. the misclassification error rate can be computed in the assumed model. Assuming a model is in fact adding some information to the problem: *prior information*.

Although almost all the model-free and model-based rules perform admissibly in large sample sizes, the difference between these methods becomes more salient when one deals with *small sample* settings: when n is comparable with p (or b). It turns out that the performance of most of the data-driven (model-free) classification rules, in the sample size range of a typical phenotypic classification problem, significantly degrades.

1.1.3 Phenotype Classification

In general the problem of phenotype classification is to classify between different diseases or between subtypes of a heterogeneous cancer, e.g. BCR1 and BCR2 breast

cancer. The input data here are usually in the form of gene expressions: the main task here is to assign a *phenotype* (perhaps a macroscopic trait) to measured high-dimensional gene expressions (a microscopic observation). Gene expressions can be obtained through several techniques, e.g. microarray or next-generation sequencing technology.

Despite of daily decrease in the cost of genomic data acquisition with the advent of high-throughput sequencing technologies, there still remains a huge gap between the sample size of a phenotypic classification problem and the dimension. There can be tens of thousands of potential features (gene expressions) but the sample sizes tend to be small, typically under 100 and often less than 50. This makes classification problematic. A promising approach to alleviate the problem is the use of *prior knowledge*. For example, the usual procedure for classifier design is to apply a classification rule to a set of features and sample data with the result being a designed classifier that will be applied to the population (all future observations). Prior knowledge can play a role in deciding upon the nature of the data and the original list of features. Knowledge may also be used in choosing a classification rule based on the nature of physical characteristics. The salient point from our perspective herein is that, once the features, sampling procedure, and classification rule are decided upon, from that point on the typical classification rule proceeds without operational knowledge concerning the features. In particular, no assumptions are made regarding the feature-label distribution (population) from which the sample data are drawn, despite the availability of a large amount of information contained in signaling pathways that specify underlying interactions between entities (e.g. genes or proteins), contributed either in normal functioning or malfunctioning (e.g. diseases) cellular states. These pathways are mostly available in the relevant literature or in public databases (e.g. KEGG, BioCarta, Reactome). *If knowledge concerning the*

feature-label distribution is available, then it can be used in classifier design.

1.2 Biological Pathways

Biological pathways are graphical representation of a group of molecules (genes/proteins) in a cell that work together to control one or more cell functions, such as cell division or cell death. For instance, consider the *mammalian cell cycle* network illustrated in Figure 1.1. After the first molecule in a pathway receives a *signal*, after some time unites, it either activates or inhibits (suppresses) its downstream molecules directly connected to it. This process is repeated until the last molecule is influenced by the initial stimulating signal and the cell function is carried out. Abnormal activation of these pathways can lead to cancer or other diseases. For example, in the mammalian cell cycle shown in Figure 1.1, having permanent *down-regulated* Rb, p27 and in the absence of the *extra-cellular signal* CycD, the system undergoes a faulty interconnection, which represents cancerous phenotypes, in which the cell cycles even in the absence of any *growth factor.*

1.2.1 Pathway-Based Classification

Protein-protein interaction (PPI) networks or gene-gene networks have been widely used as *a priori* knowledge, namely called *pathway-based classification*, to improve classification accuracy [6–11], consistency of biomarker discovery [12,13] and targeted therapeutic strategies [14, 15]. For example, to improve classification performance, several studies have proposed to interpret the gene expression data at the level of functional modules (i.e., pathways), instead of at the level of individual genes, by utilizing available pathway knowledge [16, 17]. These pathway-based methods try to infer the activity level of a given pathway by analyzing the expression of its member genes, which is then used as a potential feature. These studies have shown that such "pathway markers" are generally more reproducible compared to "gene mark-

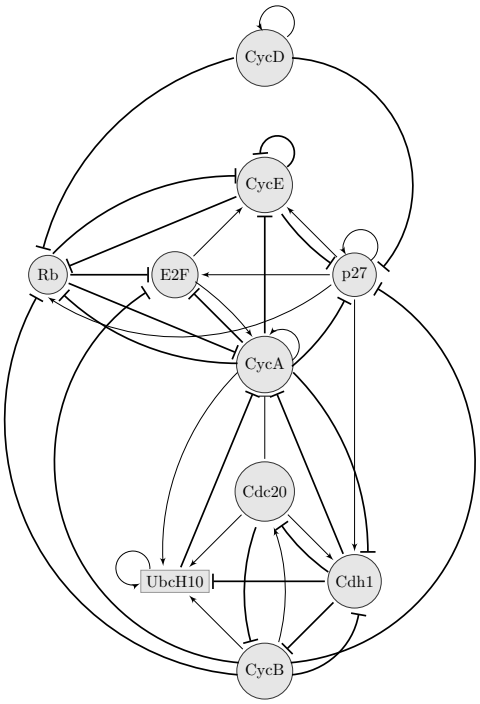


Figure 1.1: Pathways indicating the regulations between different molecules.

ers" and that they lead to better classification performance. Another example is the network-based classification approach [18, 19], which has been gaining interest in recent years. These network-based methods try to identify "subnetwork markers" by overlaying the gene expression data on a large-scale PPI network, where each gene is mapped to the corresponding protein, and searching for differentially expressed subnetwork regions. It has been shown that these subnetwork markers often yield more accurate classification results and have better reproducibility compared to both gene and pathway markers. Nonetheless, the majority of these studies utilize gene expressions corresponding to sub-networks in PPI networks, for instance: mean or median of gene expression values in gene ontology network modules [6], probabilistic inference of pathway activity [9], and producing candidate sub-networks via a

Markov clustering algorithm applied to high quality PPI networks [11, 20]. Considering that pathways are partial representations of the gene regulatory network and that the PPI network provides a skeleton of the biological network underlying cells, the aforementioned methods can be viewed as attempts to construct better classifiers by integrating partial network knowledge with measurement data.

Although recent advances in pathway-based and network-based classification have demonstrated the potential for utilizing prior knowledge to improve genomic classification, currently available methods mostly rely on heuristics. In this dissertation, we propose a general paradigm for classification that incorporates prior knowledge along with the data in the context of an optimization procedure.

1.3 Contributions

In our case, the application in mind is phenotype classification based on gene (or protein) expression measurements in the steady-state of a biological network. This "biomarker problem" is perhaps the most active area of research in genomics owing to the potential for disease diagnosis and prognosis. Rather than depend only on expression data, one can use classical genetic pathway information to provide prior knowledge and augment classifier design. The contribution of this dissertation can be categorized into two parts: (1) Utilizing pathway knowledge through an intermediate step by which the pathways are first transformed to uncertainty class of dynamical networks, being Boolean network with perturbation (BNp), and (2) Incorporation of pathways from a Bayesian perspective.

In this dissertation, we employ *regularization* as an effective approach for integrating prior knowledge with different criteria. Using regularization goes back to 1948 where regularization was introduced in the context of solving integral equation numerically by Andrey Tikhanov [21], known as *Tikhanov regularization*. A simi-

lar approach was proposed by Phillips as "numerical solutions for certain integral equations" in 1962 [22]. Under a different name and context, regularization has been used in statistical problems as *ridge regression* [23]. Both Tikhanov regularization and ridge regression can be seen as effective methods to solve ill-posed problems via incorporating prior knowledge into the problem. Afterwards, there has been an enormous amount of work both in statics [24–26] and more recently in signal processing [27–30] dealing with regularization in a wide range of problems. A brief survey on the use of regularization in statistics can be found in [31].

1.3.1 Regularized Maximum-Log-Likelihood

The example laid out in this approach involves the following chain: {pathways} $\rightarrow$ {class of networks} $\rightarrow$ {class of steady-state distributions}. Prior knowledge in the form of a set of pathways constrains the possible behaviors of the dynamical system to an "uncertainty class" of networks consistent with the pathway information [1]. Each of these possesses a steady-state distribution, thereby yielding an uncertainty class of steady-state distributions. Figure 1.2 shows an illustrative view of this process chain. Detailed description of this figure is given in Section 2.5.

Hence, rather than assume nothing is known about the feature-label distribution than what can be extracted from the data during classifier design, we can impose the constraint that the feature distribution belongs to the uncertainty class of steady-state distributions shown by a box in the middle of Figure 1.2. Put simply, a classifier is designed based on the uncertainty class of steady-state distributions, denoted by Π^0 and Π^1 in Figure 1.2, and the steady-state data.

We emphasize that while the particular application motivating our interest involves the generation of a steady-state uncertainty class from genetic pathway information, the theoretical content of this dissertation lies solely within classification

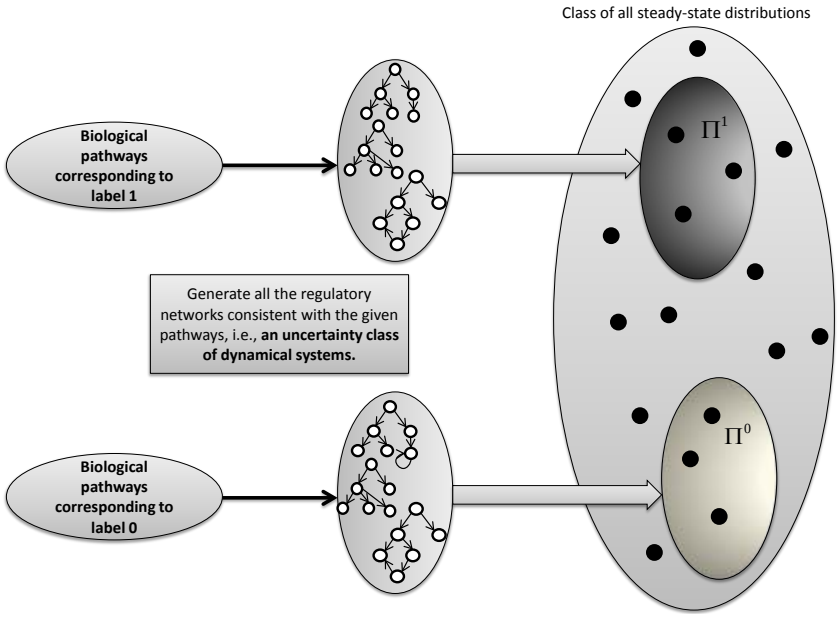


Figure 1.2: An illustrative example of the chain: {pathways} $\rightarrow$ {class of networks} $\rightarrow$ {class of steady-state distributions}. In this schematic view, an intermediate step is applied to construct a class of dynamical systems whose behaviors are consistent with the given pathways, for example, see the methods in [1] and [2]. Two uncertainty classes are shown by Π^0 and Π^1 for labels zero and one, respectively. These classes will be employed as the prior knowledge in the classifier design.

theory – classifier design assuming an uncertainty class of feature distributions. In line with that focus, we provide analytic characterization of the first and second moments of the true error for two well-known uncertainty models, ε-contamination and p-point uncertainty classes, under the assumption of stratified sampling. Characterization of these moments is basic to understanding the behavior of a classification rule and has a long history in pattern recognition, most commonly with stratified sampling [32], [33]- [34]. Recently, the issue of true-error moments has been addressed in the context of the joint distribution of the true and estimated error moments, in this case the most important moment being the second-order mixed moment between the true and estimated errors because this mixed moment is critical to characterizing the accuracy of the error estimate [32, 33, 35, 36]. The proposed method and the associated moments for defined uncertainty class models are explained and extended in detail in Section 2.

1.3.2 Bayesian Framework

In a different approach from above, a Bayesian framework is used for quantifying and incorporating the information in biological pathways. A new sophisticated method is proposed to quantify the knowledge in the pathways. We take a Bayesian view, where everything is translated to expectations with respect to the model parameters found by prior probability. Two types information are extracted: (1) Regulatory set, and (2) Pairwise regulations in which the type of influence, i.e. activation or inhibition, is also taken into account. The Bayesian modeling of biological pathways are described in details in Section 3.

The two-fold problem of pattern recognition has been recently addressed in a Bayesian framework, Bayesian minimum-mean-square error (MMSE) error estimator [37, 38], and optimal Bayesian classifier (OBC) [39, 40]. Considering the focus

of this dissertation, it has been shown that using the Bayesian framework, the designed OBC classifier performs optimally with respect to the *assumed uncertainty model* represented by the *prior probability.*[*] It has been shown, to a great extent, that Bayesian perspective can significantly improve classification. However, this improvement is extensively dependent on the prior chosen to be combined with the data. In order to show the importance of having a "good" prior distribution (in the sense of being centered around the true underlying distribution), we show the results of a small study here, where we focus on discrete OBC with Dirichlet prior and multivariate Gaussian problem with Normal-inverse-Wishart prior on the mean and covariance matrix of the underlying distributions [39].

1.3.2.1 Effect of prior mismatch: a small study

We generate two distributions as the true class-conditional ditributions via Zipf model with $b = 16$ variables and the parameter $a = 1.5$ [41]. As of the hyperparameters, we use

$$\boldsymbol{\alpha}^0_{\text{mis}} = \alpha_0((1-\varepsilon)\boldsymbol{p}^0_{\text{Zipf}} + \varepsilon \boldsymbol{p}_{\text{cont.}}),\ \boldsymbol{\alpha}^1_{\text{mis}} = \alpha_0 \boldsymbol{p}^1_{\text{Zipf}}$$

where vector $\boldsymbol{p}_{\text{cont.}}$ is a random contamination probability mass function, and also the parameter ε is used to control the amount of contamination into our prior distribution. The setting above means that there is no mismatch in the center of prior for class 1 and the contamination only affects class zero's prior. The parameter α_0 determines the concentration of the uncertainty class, i.e., the total variance of the Dirichlet prior. We consider five scenarios for the variance parameter: $\alpha_0 = b/2, b, 2b, 4b, 8b$. Two OBC classifiers are designed; one with the matched centered priors with hyperparameters $\boldsymbol{\alpha}^0 = \alpha_0 \boldsymbol{p}^0_{Zipf}$ and $\boldsymbol{\alpha}^1 = \alpha_0 \boldsymbol{p}^1_{Zipf}$, and the other one using $\boldsymbol{\alpha}^0_{\text{mis}}$ and $\boldsymbol{\alpha}^1_{\text{mis}}$,

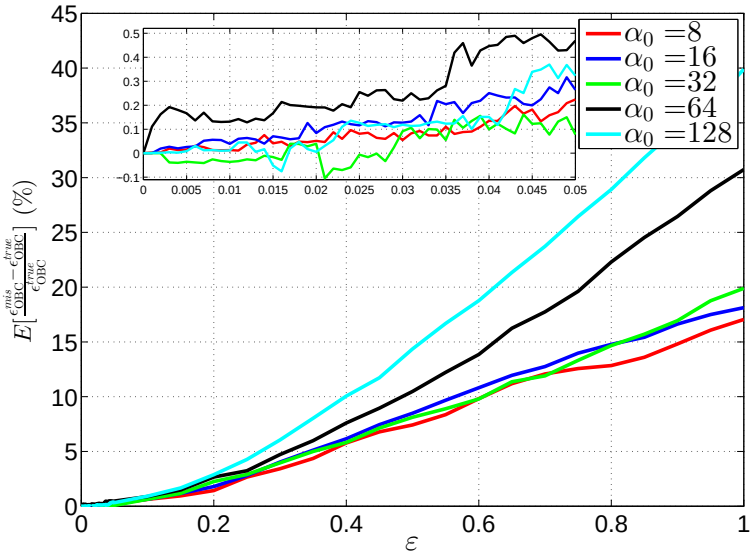
[*]It should be noted that, the OBC classifier is not guaranteed to perform better than traditional methods for *any specific model*.

defined above.

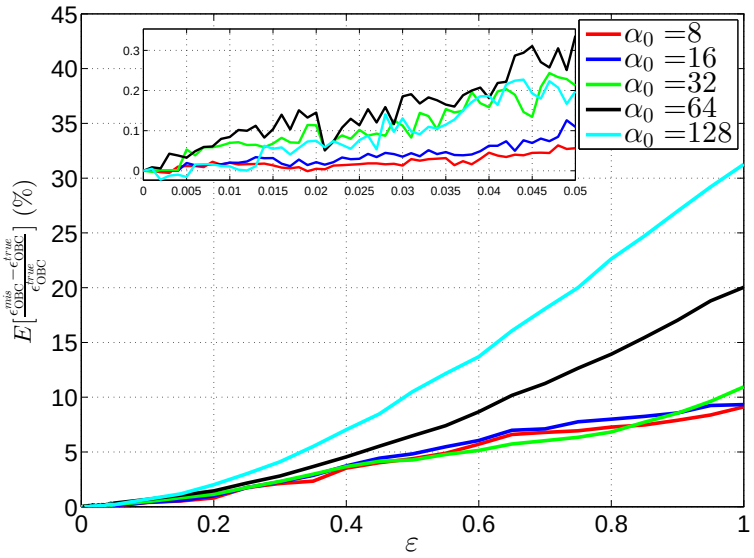
We randomly generated $n = 50$ and $n = 100$ samples according to the true probabilities whereas classes sample sizes are determined randomly based on $c = 0.5$. We computed the expected loss due to the contaminated prior for class 0 by computing $E[\frac{\epsilon^{\text{mis}}_{\text{OBC}} - \epsilon^{\text{true}}_{\text{OBC}}}{\epsilon^{\text{true}}_{\text{OBC}}}]$ where the expectation is taken with respect to random contamination and training sample. The variables $\epsilon^{\text{true}}_{\text{OBC}}$ and $\epsilon^{\text{mis}}_{\text{OBC}}$, respectively, stand for the true error for the OBC classifier when the prior is centered around the true distribution, $\boldsymbol{p}^{y}_{\text{Zipf}};\ y = 0, 1$ and the OBC classifier when the prior's center is shifted.The expectation is approximated via Monte-Carlo with $50,000$ iterations.

Figure 1.3 illustrates the percentage of the performance loss as contamination in the prior increases. It shows that, for a fixed parameter α_0, as the contamination, induced by parameter ε, increases, the performance of the designed OBC using the contaminated prior significantly degrades. Thus, even if one is aware of the true value of α_0. On the other hand, for a fixed contamination, as the prior knowledge concentrates more around an incorrect distribution, the expected loss tremendously increases, where for $\varepsilon = 0.6$, and $\alpha_0 = 4b = 64$ about 15% loss incurs for $n = 50$ which drops down to 10% by increasing sample size to $n = 100$. The latter is the everyday compromise of Bayesian frameworks, if prior is not chosen properly, there would be a need for elevating the sample size to compensate "inaccurate" prior probability.

Next, we perform a small study to show how improper prior can degrade superiority of Bayesian framework when applied in a multivariate Gaussian setting. We consider the problem of classification when the underlying covariance matrices follow the proposed blocked structure [42]. The mean vectors are set to $\boldsymbol{\mu}_0^{true} = 0.3\mathbf{I}_p$ and $\boldsymbol{\mu}_1^{true} = -\boldsymbol{\mu}_0^{ture}$, in which $p = 8$ variables are contributing in the underlying model. We consider unequal covariance matrices $\boldsymbol{\Sigma}_0^{ture} \neq \boldsymbol{\Sigma}_1^{true}$. Two cases are considered: (1) Prior on the mean covariance matrices centered around the truth for



(a) $n = 50$



(b) $n = 100$

Figure 1.3: Effect of mismatch in the prior measured by its effect on the expected true error of the designed optimal Bayesian classifier. The sampling is random with class prior probability $c = 0.5$. The smaller box illustrates a zoomed-in version of the interval $0 \leq \varepsilon \leq 0.1$. One can see that for mismatch values less than 0.1, for $n = 50$, there is a possibility of performing better than the prior centered on the true distributions.

both classes: $\boldsymbol{\mu}_y \sim \mathcal{N}(\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y/\nu_y)$; $\boldsymbol{\Sigma}_y \sim \mathcal{W}^{-1}((\kappa - p - 1)\boldsymbol{\Sigma}_0^{true}, \kappa)$ for $y \in \{0, 1\}$. (2) The case with prior mismatch only on covaraince matrix for class $y = 0$ while the prior for class $y = 1$ is left as case (1). Hence, for the mean vectors we have: $\boldsymbol{\mu}_y^{miss} \sim \mathcal{N}(\boldsymbol{\mu}_y^{true}, \boldsymbol{\Sigma}_y^{miss}/\nu)$, in which for $y = 0$, a convex contamination model is assumed, given by

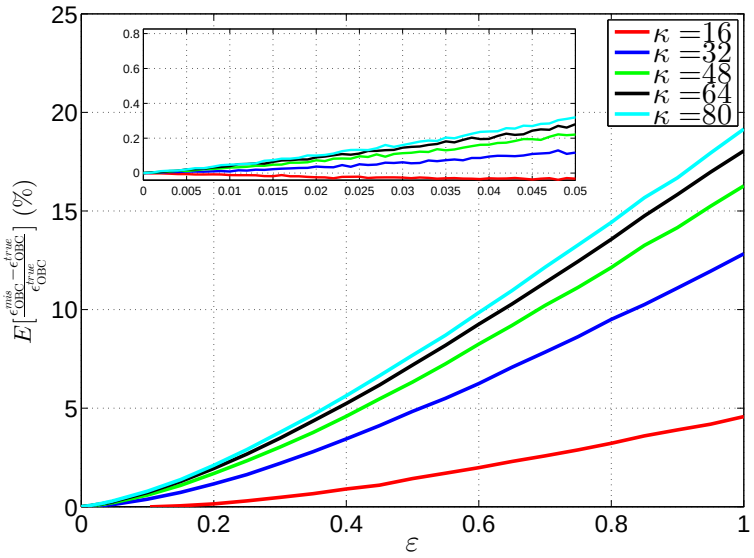
$$\boldsymbol{\Sigma}_0^{miss} \sim \mathcal{W}^{-1}((\kappa - p - 1)[(1 - \varepsilon)\boldsymbol{\Sigma}_0^{true} + \varepsilon \mathbf{I}_p], \kappa).$$

Under this prior, the prior's center, for class $y = 0$, is shifted to

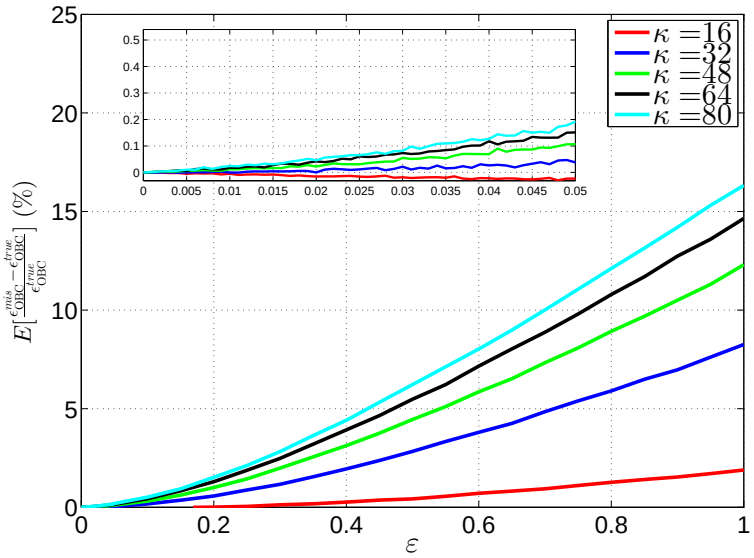
$$(1 - \varepsilon)\boldsymbol{\Sigma}_0^{true} + \varepsilon \mathbf{I}_p.$$

The covariance matrix for $y = 1$ under the mismatched model is identical to that of case (1). In the defined contamination model, as ε goes to 1, the correlation coefficients *shrink* towards 0, i.e. full independence. Similar to the discrete case, the variable $E[\frac{\epsilon_{\text{OBC}}^{\text{mis}} - \epsilon_{\text{OBC}}^{\text{true}}}{\epsilon_{\text{OBC}}^{\text{true}}}]$ is computed through Monte-Carlo simulations with $10,000$ repetitions. The results for two sample sizes $n = 30$ and $n = 60$ with random sampling according to the fixed class prior probability, $c = 0.5$, are shown in Figure 1.4.

One can see that, again, similar to the discrete setting, as the contaminating factor increase the performance of the Bayesian classifier compared to its optimal performance degrades. For instance, with $n = 30$ samples, with $\varepsilon = 0.4$, the expected loss is between 1% to 5% when we increase κ from 16 to 80. It means that as we trust more on a contaminated prior, the degradation become more tangible. Increasing the sample size to $n = 60$ decreases the performance loss. The reason is that we are extracting more knowledge from the data compared to the case $n = 30$, and hence a contaminated prior has a lesser deteriorative effect.



(a) $n = 30$



(b) $n = 60$

Figure 1.4: Effect of mismatch in the prior measured by its effect on the expected true error of the designed optimal Bayesian classifier. The sampling is random with class prior probability $c = 0.5$. The smaller box illustrates a zoomed-in version of the interval $0 \leq \varepsilon \leq 0.05$.

1.3.2.2 Prior Probability Construction: Definition

Let the class-conditional densities, being two distinct measures on $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$, be denoted by $p^y(X)$ for $y \in \{0, 1\}$ *. $\mathcal{B}(\mathcal{X})$ contains the Borel subsets of $\mathcal{X}$. For simplicity, in what follows we drop the sup (sub-)scripts associated with the labels, y. But, one should notice that all the results are applied for both classes. Now, we give a definition to the problem of prior selection:

Given a set of prior information, e.g. biological (signaling) pathways, we aim to find a prior distribution, denoted by $\pi(\boldsymbol{\theta})$, on the space of probability measures on $(\mathcal{X}, \mathcal{B})$, e.g., random multinomial probabilities. In other words, let

$$\mathcal{F} = \{\boldsymbol{p} : \boldsymbol{p} \textit{ is a probability measure on } (\mathcal{X}, \mathcal{B})\},$$

and accordingly, $\mathcal{A}$ denotes some suitable $\sigma-$algebra of subsets of $\mathcal{F}$. Then, we look for measures, $\pi(\boldsymbol{\theta})$ on $(\mathcal{F}, \mathcal{A})$, being consistent with the given prior information. Although any measure on the space $(\mathcal{F}, \mathcal{A})$ can be considered as the prior, it is more desirable that the selected prior $\pi(\boldsymbol{\theta})$ satisfies certain properties. In [43], it is mentioned that a constructed prior probability would need to have some properties to be "desirable" in a Bayesian framework:

- The class Π, of random prior distribution on $\mathcal{F}$ should be analytically tractable in three respects:
  - It should be reasonably easy to determine the posterior distribution on $\mathcal{F}$, given a "sample";

*Although it is possible to give a definition to the conditional distributions using the conditional expectation, here we define our problem of interest through giving a direct definition to the class-conditional probabilities and then, to the prior probabilities.

  - It should be possible to express conveniently the expectations of simple loss functions; and
  - The class Π should be closed, in the sense that if the prior is a member of Π, then the posterior is a member of Π, (i.e. conjugate priors).

- The class Π should be "rich," so that there will exist a member of Π capable of expressing any prior information.
- The class Π should be parametrized in a manner which can be readily interpreted in relation to prior information.

In practice, specifically in the biological problems, the prior information is not a testable piece of information. It is rather a qualitative pairwise illustration of the dependency between the genes/proteins called thus far in this section biological pathways. Hence, the first step before proceeding to any prior probability construction is to transform the knowledge in these pathways to a set of testable information constraints.

1.3.3 Bibliography on Prior Construction

For about 200 years after the Bayes-Laplace uniform prior, Bayesian statistics was based on non-informative priors [44]. After Jeffreys' non-informative prior, which was based on Fisher's information [45], there followed several objective-based methods have been proposed to construct prior probabilities in different contexts [44, 46–54]. Among these, we have: maximal data information priors (MDIP) [51], non-informative priors for integers [48], entropic priors [50], reference (non-informative) priors obtained through maximization of the missing information [44], and least-informative priors [49]. The principle of maximum entropy (MaxEnt) can be seen as a method of constructing least-informative priors [55, 56]. Except for Jeffreys'

prior, almost all methods are based on optimization: maximizing or minimizing an objective function. In [57], several non-informative and informative priors for different problems are found. All of these methods emphasize the separation of prior knowledge and observed sample data. Although these methods are appropriate tools for generating prior probabilities, they are quite general methodologies, i.e., they do not target specific scientific prior information.

1.4 Objective-Based Prior Probability Construction

We introduce the notion of *objective-based priors*, when the prior information is in the form of signaling pathways. Unlike the previously used methods where the prior information is either in the form of known inequalities or equalities, we consider the notion of "slackness." In order to bring the slackness variables, the interactions in the pathways are quantified from a Bayesian perspective, "mapping the signaling pathways to a set of constraints on the hyperparameter space."

In the objective-based approach, the prior construction problem falls into the context of *model selection*. In general, we restrict ourselves to some parametrized model shown in general by Π. For example in the multivariate Gaussian scenario, we have

$$\Pi = \{\mathcal{NW}^{-1}(\boldsymbol{m}, \boldsymbol{\Psi}, \nu, \kappa) : \ \boldsymbol{m} \in \mathbb{R}^p, \ \boldsymbol{\Psi} \in \mathbb{R}_+^{p\times p}\}$$

with known ν and κ. Here, $\mathbb{R}_+^{p\times p}$ stands for the positive definite matrices with dimension p. And in the discrete setting, the model space is restricted to that of

$$\Pi = \{\mathcal{D}(\boldsymbol{\alpha}) : \ \boldsymbol{\alpha} \in S_{b-1}^{\alpha_0}\},$$

where $S_{b-1}^{\alpha_0}$ denotes the $(b-1)-$dimension simplex whereas

$$\alpha_i > 0,\ i \in \{1, \ldots, b\},\ \text{and}\ \sum_{i=1}^{b-1} \alpha_i \leq \alpha_0.$$

Then, the problem reduces to estimating a set of parameters, called *hyperparameters* here, so that not only satisfy the constraints imposed by the pathways, but meet some sound criterion. These criteria can be data-based (similar to *empirical Bayes methods*) or information theoretic functions, e.g. entropy.

1.4.1 Multivariate Gaussian: Normal-Wishart Prior

In Section 4 of this dissertation, the problem of prior probability construction is addressed by proposing a series of optimization paradigms that utilize the incomplete prior information contained in pathways (both topological and regulatory). The optimization paradigms employ the marginal log-likelihood, established using a small number of feature-label realizations (sample points) regularized with the prior pathway information about the variables. In the special case of a Normal-Wishart prior distribution on the mean and inverse covariance matrix (precision matrix) of a Gaussian distribution, these optimization problems become convex.

1.4.2 Discrete Case: Dirichlet Parameter Selection

The problem of Dirichlet prior construction for the discrete classification is addressed in Section 5, where we extend maximum entropy and maximal data information prior to the proposed framework. Moreover, a recently introduced method of prior construction, regularized expected mean log-likelihood, is also revisited. Our problem of interest in this case is discrete classification, and hence we consider the optimal Bayesian classification when the likelihood function results from a multinomial distribution. All the methods are studied for Dirichlet prior families. We examine

the proposed framework on a simplified set of pathways involving the TP53 gene. We show that the Bayesian framework utilizing the informative constructed priors via objective-based priors framework significantly outperforms those rules which do not incorporate prior knowledge.

2. CLASSIFIER DESIGN GIVEN AN UNCERTAINTY CLASS OF FEATURE DISTRIBUTIONS DERIVED FROM BIOLOGICAL PATHWAYS VIA REGULARIZED MAXIMUM-LIKELIHOOD*

Contemporary high-throughput technologies provide measurements of very large numbers of variables but often with very small sample sizes. This section proposes an optimization-based paradigm for utilizing prior knowledge to design better performing classifiers when sample sizes are limited. We derive approximate expressions for the first and second moments of the true error rate of the proposed classifier under the assumption of two widely-used models for the uncertainty classes; ε-contamination and p-point classes. The applicability of the approximate expressions is discussed by defining the problem of finding optimal regularization parameters through minimizing the expected true error. Simulation results using the Zipf model show that the proposed paradigm yields improved classifiers that outperform traditional classifiers that use only training data. Our application of interest involves discrete gene regulatory networks possessing labeled steady-state distributions. Given prior operational knowledge of the process, our goal is to build a classifier that can accurately label future observations obtained in the steady state by utilizing both the available prior knowledge and the training data. We examine the proposed paradigm on networks containing NF-κB pathways, where it shows significant improvement in classifier performance over the classical data-only approach to classifier design.

*Parts of this section are reprinted with permission from "Classifier Design Given an Uncertainty of Feature-Label Distributions via Regularized Maximum-Likelihood and the Incorporation of Biological Pathway Knowledge in the Steady-State Phenotype Classification" by M. Shahrokh Esfahani, J. Knight, A. Zollanvari, B-J Yoon, and E. R. Dougherty, 2013, *Pattern Recognition*, vol. 61, no. 15, pp. 3880–3894, © 2013 Elsevier, and "Designing enhanced classifiers using prior process knowledge: Regularized maximum-likelihood" by M. Shahrokh Esfahani, A. Zollanvari, B-J Yoon, and E. R. Dougherty, 2012, *Proceedings of the International Workshop on Genomic Signal Processing and Statistics*, San Antonio, TX, December 2012, pp 1012–1016, © 2011 IEEE.

This section is organized in the following manner. In Section 2.1, we introduce our proposed paradigm. True error statistics for the stratified sampling case are derived in Section 2.2. Section 2.3 contains a brief discussion on the regularization parameter defined and used throughout the section. Simulation results are shown in Sections 2.4 and 2.5 where we show the improvement of the designed classifier over the histogram rule in synthetic and biologically inspired cases, respectively. Finally, Section 2.6 contains concluding remarks.

We use the following notation and abbreviations. Boldface lower case letters denote column vectors. The cardinality of the set, Π is denoted by $|\Pi|$. $\pi(k)$ and $\boldsymbol{\pi}^T$ denote the $k-$th element and the transpose of the vector $\boldsymbol{\pi}$, respectively . $\Pr(A)$ denotes the probability of event A. The binomial distribution is shown by $bin(n,p)$. $bin(n,p) = x$ is used to denote the binomial random variable having value x. The trinomial distribution is shown by $trin(n,p_1,p_2)$. Thus,

$$\Pr(trin(n,p_1,p_2) = (x_1,x_2)) = \binom{n}{p_1, p_2, 1-p_1-p_2} p_1^{x_1} p_2^{x_2} (1-p_1-p_2)^{n-x_1-x_2}.$$

To show the comparison between two vectors, we use $\boldsymbol{\pi_1} \preceq \boldsymbol{\pi_2}$ meaning that the vector $\boldsymbol{\pi_1}$ is element-wise less than or equal to $\boldsymbol{\pi_2}$. The notation $\mathrm{E}_x(g(x))$ is used to denote taking expectation of $g(x)$ with respect to the subscript x. The indicator function for the event A is shown by I_A.

2.1 Regularized Maximum-Likelihood

In this section, we propose an optimization paradigm for classifier design that utilizes both an uncertainty class (from prior knowledge) and the available training data. Let $\pi_{ac}^y(k) = \Pr(X = k|Y = y)$ be the true conditional distribution of the feature $X = k \in \{1,...,b\}$ given the class label $y \in \{0,1\}$, and let $c_y = \Pr(Y = y)$ be the prior distribution of the class label. We can build a classifier by first finding

label conditional probabilities $\hat{\pi}^y(k)$ that estimate the true probabilities $\pi^y_{ac}(k)$ and then defining

$$\psi(k) = \mathrm{I}_{\{c_1\hat{\pi}^1(k) \geq c_0\hat{\pi}^0(k)\}} = \begin{cases} 1, & \text{if } c_1\hat{\pi}^1(k) \geq c_0\hat{\pi}^0(k) \\ 0, & \text{otherwise} \end{cases} . \tag{2.1}$$

This can be viewed as using the "plug-in rule" in the Bayes classifier $\psi(k) = \mathrm{I}_{\{c_1\pi^1_{ac}(k) \geq c_0\pi^0_{ac}(k)\}}$. In the absence of prior knowledge, the label-conditional distribution $\Pr(X = k|Y = y) = \pi^y_{ac}(k)$ is estimated solely based on the training data by solving the following maximum log-likelihood problem:

$$\min_{\boldsymbol{\pi}^{yT}\mathbf{e}=1, \mathbf{0} \preceq \boldsymbol{\pi}^y} \quad -\sum_{k=1}^{b} u^y_k \log \pi^y(k), \tag{2.2}$$

where u^y_k is the number of sample points at state k with label y and $\mathbf{e}$ is the all-one column vector. The solution to (2.2) is

$$\hat{\pi}^y_{\mathbf{data}}(k) = \frac{u^y_k}{n_y}, \tag{2.3}$$

where n_y is the number of sample points with label y.

We now assume we have *uncertainty classes*, $\Pi^y = \{\boldsymbol{\pi}^y_1, \boldsymbol{\pi}^y_2, ..., \boldsymbol{\pi}^y_{|\Pi^y|}\}, y = 0, 1$, e.g., see Figure 1.2, conveying the prior network knowledge of the label-y conditional distribution, $\boldsymbol{\pi}^y_{ac}$. We adapt (2.2) to form the following weighted-sum optimization problem for the class labels $y = 0, 1$, which includes a term contributed by the uncertainty class:

$$\min_{\boldsymbol{\pi}^{yT}\mathbf{e}=1, \mathbf{0} \preceq \boldsymbol{\pi}^y} \quad -(1-\lambda_y)\sum_{k=1}^{b} u^y_k \log \pi^y(k) + \lambda_y \ell(\boldsymbol{\pi}^y, \Pi^y) \ . \tag{2.4}$$

The *regularization parameter* $\lambda_y \in [0,1]$ reflects the uncertainty of the labeled training data compared to the total amount of uncertainty in our prior knowledge and $\ell : \mathcal{S}_b \times \mathcal{S}_b^{|\Pi^y|} \to [0,\infty)$, where $\mathcal{S}_b$ is the *standard unit* $(b-1)-$*simplex* and $\mathcal{S}_b^{|\Pi^y|}$ is any uncertainty class containing $|\Pi^y|$ $b-$dimensional distributions, is a nonnegative function to measure the *dissimilarity* between a given $\boldsymbol{\pi}^y$ and the uncertainty class.

If the objective function in (2.4) is a convex function, then the optimization problem can be solved efficiently. Since the log-likelihood of the multinomial distribution is concave (i.e., the negative log-likelihood function for $\pi^y(k), k = 1, ..., b$, given the sample, is convex), it is sufficient to use a convex function for ℓ (i.e., the regularizer term) in (2.4) to make it a convex programming problem. We use

$$\ell(\boldsymbol{\pi}^y, \Pi^y) := \frac{1}{|\Pi^y|} \sum_{i=1}^{|\Pi^y|} D(\boldsymbol{\pi}_i^y || \boldsymbol{\pi}^y), \tag{2.5}$$

where $D(\boldsymbol{\pi}_i^y || \boldsymbol{\pi}^y) = \sum_{k=1}^{b} \pi_i^y(k) \log \frac{\pi_i^y(k)}{\pi^y(k)}$ is the *Kullback Leibler (information) distance* (KL-distance).

Lemma 1 (RML Classifier)**.** *Suppose that the dissimilarity function ℓ is defined as* (2.5)*. Then, the solution to the regularized maximum-likelihood (RML) problem in* (2.4) *is obtained bin-wise as*

$$\hat{\pi}^y_{\mathbf{RML}}(k) = \frac{(1-\lambda_y) u_k^y + \lambda_y \overline{\pi}^y(k)}{(1-\lambda_y) n_y + \lambda_y}; y \in \{0,1\}, \forall k = 1, ..., b, \tag{2.6}$$

where $\overline{\pi}^y(k)$ is the probability of the $k-$th bin obtained from the average of $\boldsymbol{\pi}_i^y, i = 1, 2, ..., |\Pi^y|$ in the corresponding uncertainty class $\Pi^y, y \in \{0,1\}$. The corresponding RML classifier can be found by plugging $\hat{\boldsymbol{\pi}}^0_{\mathbf{RML}}$ and $\hat{\boldsymbol{\pi}}^1_{\mathbf{RML}}$ in equation (2.1)*.*

Proof. Plugging (2.5) in (2.4), we obtain

$$\begin{aligned}\hat{\boldsymbol{\pi}}^y_{RML} &= \arg\min -(1-\lambda_y)\sum_{k=1}^{b} u_k^y \log \pi^y(k) + \tfrac{\lambda_y}{|\Pi^y|}\sum_{i=1}^{|\Pi^y|}\sum_{k=1}^{b}\pi_i^y(k)\log\tfrac{\pi_i^y(k)}{\pi^y(k)} \\ &= \arg\min -\Big[(1-\lambda_y)\sum_{k=1}^{b} u_k^y \log \pi^y(k) + \lambda_y \sum_{k=1}^{b}\log\pi^y(k)\{\tfrac{1}{|\Pi^y|}\sum_{i=1}^{|\Pi^y|}\pi_i^y(k)\}\Big] \\ &= \arg\min -\Big[\sum_{k=1}^{b}[(1-\lambda_y)u_k^y + \lambda_y\overline{\pi}^y(k)]\log\pi^y(k)\Big]\end{aligned} \tag{2.7}$$

The solution to this problem can be obtained using a *Lagrangian multiplier* similar to (2.2), which leads to the label conditional probabilities in (2.6). Q.E.D.

Consider the following two special cases:

1. Suppose the uncertainty in the information extracted from the training data is much less than that in the prior network knowledge. In the limiting case, $\lambda_y \to 0$ and

$$\lim_{\lambda_y\to 0}\hat{\pi}^y_{\mathbf{RML}}(k) = \frac{u_k^y}{n_y}, \forall k = 1, ..., b. \tag{2.8}$$

   This is consistent with our expectation: if there is infinite amount of training data (hence no uncertainty therein), the classifier can be perfectly estimated from the data.

2. Suppose we have very good prior network knowledge, so that the uncertainty in this knowledge is much smaller compared to that extracted from the data. In the limiting case, $\lambda_y \to 1$ and

$$\lim_{\lambda_y\to 1}\hat{\pi}^y_{\mathbf{RML}}(k) = \overline{\pi}^y(k), \forall k = 1, ..., b. \tag{2.9}$$

   If we have perfect knowledge of the steady-state distribution, then we do not need training data.

In this dissertation we consider two models having finite uncertainty classes:

2.1.0.1 ε-contamination uncertainty class

The ε-contamination class has been used for modeling uncertainty in a wide range of applications, including robust hypothesis testing [58], robust Wiener filtering (uncertainty about the spectral density) [59, 60], Bayesian robust optimal linear filter design [61], robust decision making problems [62], and minimax robust quickest change detection (with the application in intrusion detection in computer networks and security systems) [63]. In [59]- [61], the ε-contamination class contains all the power spectral densities (PSD) in the vicinity of the nominal PSD. In [58] and [63], the ε-contamination contains all the probability densities in the vicinity of the nominal one.

Here, we use ε-contamination to model the uncertainty about the label-conditional probabilities. We define the ε-contamination class of multinomial distributions associated with each label as the class containing the distributions in the form of

$$\boldsymbol{\pi}^y = (1-\varepsilon_y)\boldsymbol{\pi}^y_{ac} + \varepsilon_y \boldsymbol{\pi}; y \in \{0,1\} \tag{2.10}$$

where $\varepsilon_y \in [0,1)$ is the degree of contamination and $\boldsymbol{\pi}$ is one of a finite number of randomly chosen densities from $\mathcal{S}_b$. Increasing ε_y corresponds to increasing the variance of prior knowledge about the true distribution. We assume a uniform distribution for the contamination part whose domain is the relative interior of the volume under the $(b-1)$-simplex. Since our application of interest is related to steady-state classifiers, we assume that in the simplex the corners and axes have measure zero.

2.1.0.2 p-point uncertainty class

The p-point uncertainty class has been used to model uncertainty in rate distortion problems, detection problems, robust Wiener filter design, and robust non-stationary signal estimation [60], [64]- [65]. In our application of interest, we often only know that the cell, in its steady state, spends a specific portion of time in a subset of states but know nothing about the details of these states individually. Hence, to model this prior knowledge, we can see the problem as a partitioning scenario: if we partition the state space, then the amount of time that the cell spends in each subset in the partition is known. Therefore, we can say that the label-conditional distributions belong to an uncertainty class of distributions satisfying the following constraints:

$$\sum_{k=1}^{b} \pi(k) \mathrm{I}_{\{k \in s_p^y\}} = \sum_{k=1}^{b} \pi_{ac}^y(k) \mathrm{I}_{\{k \in s_p^y\}}; p = 1, ..., m_y, \tag{2.11}$$

where $\boldsymbol{\pi}_{ac}^y$ is the actual steady-state distribution , $s_1^y, ..., s_{m_y}^y$ form a partition of the state space denoted by $\mathcal{P}^y$, and $\boldsymbol{\pi} \in \mathcal{S}_b$ is any density function.

We will use the following notation throughout the section for the probability mass cumulated in each partition:

$$\sum_{k=1}^{b} \pi_{ac}^y(k) \mathrm{I}_{\{k \in s_p^y\}} = \omega_p^y; p = 1, ..., m_y. \tag{2.12}$$

Moreover, we define the following mapping from state space to the partition:

$$P^y : \{1, ..., b\} \to \{1, ..., m_y\}; y = 0, 1. \tag{2.13}$$

In the extreme case, $m_y = 1$ means that we only know that the label-conditional probabilities for the bins sum up to 1, which corresponds to a minimal amount of

prior knowledge. On the other hand, $m_y = b$, i.e. $|s_p^y| = 1$, for any $p \in \{1, ..., m_y\}, y \in \{0, 1\}$, means that we are certain about the label-conditional distributions, because we are given all bin probabilities – hence minimal variance in the uncertainty class (for more details refer to Section 1 of the supplementary materials on the companion website).

2.2 Moments for the True Error

For a classifier ψ_n trained on the sample data S_n, the probability of error is defined as $\epsilon_{\mathbf{data}} = \Pr(\psi_n(X) \neq Y|S_n)$. The overall performance of the classification rule can be evaluated by the expected classification error, $\mathrm{E}(\epsilon_{\mathbf{data}}) = \mathrm{E}_{S_n}\left[\Pr(\psi_n(X) \neq Y|S_n)\right]$, over all samples of size n. When prior knowledge (denoted by "**uc**" for uncertainty class) is incorporated into classifier design, we rewrite the probability of error as

$$\epsilon_{\mathbf{data,uc}} = \Pr(\psi_{n,\Pi^0,\Pi^1}(X) \neq Y|S_n, \Pi^0, \Pi^1).$$

In this section we provide analytic representation of the first and second moments for the error in the ε-contamination and p-point uncertainty models under stratified sampling, in which sampling is performed from classes 0 and 1 in accordance with their prior probabilities. Since we incorporate prior knowledge, the moments are computed relative to all samples of size n and the uncertainty-class space. They take the form

$$\mathrm{E}(\epsilon_{\mathbf{RML}}) = \mathrm{E}(\epsilon_{\mathbf{data,uc}}) = \mathrm{E}_{\Pi^0,\Pi^1}\left[\mathrm{E}_{S_n}\left[\Pr(\psi_{n,\Pi^0,\Pi^1}(X) \neq Y|S_n)\right]|\Pi^0, \Pi^1\right], \tag{2.14}$$

$$\mathrm{E}(\epsilon^2_{\mathbf{RML}}) = \mathrm{E}(\epsilon^2_{\mathbf{data,uc}}) = \mathrm{E}_{\Pi^0,\Pi^1}\left[\mathrm{E}_{S_n}\left[\Pr(\psi_{n,\Pi^0,\Pi^1}(X) \neq Y|S_n)\right]^2|\Pi^0, \Pi^1\right]. \tag{2.15}$$

We derive tight approximations for these moments for $\lambda_y \in (0,1)$. The cases $\lambda_y \in \{0,1\}$ can be handled with a slight modification to the proof.

Theorem 1 (First-Order Moment of the True Error: $\varepsilon-$Contamination Class)**.** *Suppose that the uncertainty classes, Π^0 and Π^1, come from ε_0- and ε_1- contamination classes, respectively. Then, the first-order moment of the true-error of the RML classifier defined in Lemma 1 is given by*

$$\begin{aligned} E(\epsilon_{\mathbf{RML}}) = c_0 \sum_{k=1}^{b} \pi_{ac}^0(k) [\sum_{l_0=0}^{n_0} \sum_{j=0}^{n_1} \sum_{m=j}^{n_1} \Pr(bin(n_0, \pi_{ac}^0(k)) = l_0) \\ \times \Pr(\underline{\zeta}_{k,l_0}^0 = j) \Pr(bin(n_1, \pi_{ac}^1(k)) = m)] \\ + c_1 \sum_{k=1}^{b} \pi_{ac}^1(k) [\sum_{l_1=0}^{n_1} \sum_{j=0}^{n_0} \sum_{m=j}^{n_0} \Pr(bin(n_1, \pi_{ac}^1(k)) = l_1) \\ \times \Pr(\underline{\zeta}_{k,l_1}^1 = j) \Pr(bin(n_0, \pi_{ac}^0(k)) = m)]. \end{aligned} \tag{2.16}$$

where the random variables $\underline{\zeta}_{k,l_y}^y$, $k = 1, ..., b$, $\forall l_y = 0, .., n_y; y \in \{0,1\}$, approximately have the following probability mass function (pmf):

$$\begin{cases} \Pr(\underline{\zeta}_{k,l_y}^y = 0) = \Phi(\frac{-\mu_{k,l_y}^y}{\sigma_{k,y}}) \\ \Pr(\underline{\zeta}_{k,l_y}^y = m) = \Phi(\frac{m-\mu_{k,l_y}^y}{\sigma_{k,y}}) - \Phi(\frac{m-1-\mu_{k,l_y}^y}{\sigma_{k,y}}); m = 1, .., n_y \\ \Pr(\underline{\zeta}_{k,l_y}^y = m) = 0; m \geq n_y + 1 \end{cases}, \tag{2.17}$$

$\Phi(.)$ *being the standard normal distribution. In equation* (2.17) *we have*

$$\begin{aligned} \mu_{k,l_y}^y &= \frac{g_y l_y + (1-\varepsilon_y)\alpha_y \pi_{ac}^y(k) - (1-\varepsilon_{\overline{y}})\alpha_{\overline{y}} \pi_{ac}^{\overline{y}}(k) + \frac{\varepsilon_y \alpha_y - \varepsilon_{\overline{y}} \alpha_{\overline{y}}}{b}}{g_{\overline{y}}} \\ \sigma_{k,y}^2 &= (\frac{\alpha_y^2 \varepsilon_y^2 (b-1)}{b^2 |\Pi^y| (b+1)} + \frac{\alpha_{\overline{y}}^2 \varepsilon_{\overline{y}}^2 (b-1)}{b^2 |\Pi^{\overline{y}}| (b+1)}) / g_{\overline{y}}^2; \forall k = 1, ..., b \end{aligned} \tag{2.18}$$

where $\overline{y}$ *denotes* $1-y$ *and*

$$\begin{aligned} g_y &:= (1-\lambda_y) n_y \Big[n_{\overline{y}} (1-\lambda_{\overline{y}}) + \lambda_{\overline{y}} \Big] \\ \alpha_y &:= \tfrac{g_y \lambda_y}{1-\lambda_y}. \end{aligned} \tag{2.19}$$

Proof. Please refer to A.1. Q.E.D.

Theorem 2 (First-Order Moment of the True Error: $p-$Point Class)**.** *Let the uncertainty classes,* Π^0 *and* Π^1*, be modeled by the* $p-$*point model with partition probabilities* ω_p^0 *and* ω_p^1 *with* $p = 1, ..., m_y$ *for labels* 0 *and* 1*, respectively. Then, the first-order moment of the true-error of the RML classifier defined in Lemma 1 can be written as in equation* (2.16) *in which the random variables* $\underline{\zeta}^y_{k,l_y}$*,* $k = 1, ..., b$*, for any* $l_y = 0, .., n_y$*, approximately have the pmf as defined in equation* (2.17)*, whereas assuming the definitions in equation* (2.19)*, we have*

$$\mu^y_{k,l_y} = \frac{g_y l_y + \alpha_y \frac{\omega^y_{P^y(k)}}{|s^y_{P^y(k)}|} - \alpha_{\overline{y}} \frac{\omega^{\overline{y}}_{P^{\overline{y}}(k)}}{|s^{\overline{y}}_{P^{\overline{y}}(k)}|}}{g_{\overline{y}}}$$

$$\sigma^2_{k,y} = \left[\alpha_y^2 (\omega^y_{P^y(k)})^2 \frac{(|s^y_{P^y(k)}|-1)}{|s^y_{P^y(k)}|^2 (|s^y_{P^y(k)}|+1)|\Pi^y|} + \alpha_{\overline{y}}^2 (\omega^{\overline{y}}_{P^{\overline{y}}(k)})^2 \frac{(|s^{\overline{y}}_{P^{\overline{y}}(k)}|-1)}{|s^{\overline{y}}_{P^{\overline{y}}(k)}|^2 (|s^{\overline{y}}_{P^{\overline{y}}(k)}|+1)|\Pi^{\overline{y}}|} \right] / g^2_{\overline{y}}, \tag{2.20}$$

where the mapping $P^y(.)$ *is defined in equation* (2.13)*.*

Proof. Please refer to A.1. Q.E.D.

Theorem 3 (Second-Order Moment of the True Error)**.** *The second-order moment*

of the true-error of the RML classifier defined in Lemma 1 can be decomposed as

$$\begin{aligned} E(\epsilon^2_{\mathbf{RML}}) &= E_{\Pi^0,\Pi^1}\left[c_0^2\sum_{k=1}^{b}(\pi^0_{ac}(k))^2A^1 + c_1^2\sum_{k=1}^{b}(\pi^1_{ac}(k))^2A^0\right] \\ &+E_{\Pi^0,\Pi^1}\left[c_0^2\sum_{k_1\neq k_2}^{b}\pi^0_{ac}(k_1)\pi^0_{ac}(k_2)B^1 + c_1^2\sum_{k_1\neq k_2}^{b}\pi^1_{ac}(k_1)\pi^1_{ac}(k_2)B^0\right] \\ &+E_{\Pi^0,\Pi^1}\left[c_0c_1\sum_{k_1\neq k_2}^{b}\pi^0_{ac}(k_1)\pi^1_{ac}(k_2)C^1 + c_0c_1\sum_{k_1\neq k_2}^{b}\pi^1_{ac}(k_1)\pi^0_{ac}(k_2)C^0\right]. \end{aligned} \tag{2.21}$$

where $A^0 := E_{S_n}[I_{\{\psi(X=k)=0\}}]$ *and* $A^1 := E_{S_n}[I_{\{\psi(X=k)=1\}}]$ *can be found similarly as in Theorem 1.* B^0*,* B^1*,* C^0*, and* C^1 *are computed as follows:*

$$\begin{aligned} B^0 &:= \sum_{t_1^1,t_2^1}\Big[\sum_{(t_1^0,t_2^0)\succeq(\underline{\zeta}^1_{k_1,t_1^1},\underline{\zeta}^1_{k_2,t_2^1})}\Pr(u^0_{k_1}=t_1^0,u^0_{k_2}=t_2^0)\Pr(u^1_{k_1}=t_1^1,u^1_{k_2}=t_2^1)\Big] \\ B^1 &:= \sum_{t_1^0,t_2^0}\Big[\sum_{(t_1^1,t_2^1)\succeq(\underline{\zeta}^0_{k_1,t_1^0},\underline{\zeta}^0_{k_2,t_2^0})}\Pr(u^1_{k_1}=t_1^1,u^1_{k_2}=t_2^1)\Pr(u^0_{k_1}=t_1^0,u^0_{k_2}=t_2^0)\Big] \\ C^0 &:= \sum_{t_1^1,t_2^1}\Big[\sum_{t_1^0\geq\underline{\zeta}^1_{k_1,t_1^1},t_2^0\leq,\overline{\zeta}^1_{k_2,t_2^1}}\Pr(u^0_{k_1}=t_1^0,u^0_{k_2}=t_2^0)\Pr(u^1_{k_1}=t_1^1,u^1_{k_2}=t_0^1)\Big] \\ C^1 &:= \sum_{t_1^0,t_2^0}\Big[\sum_{t_1^1\geq\underline{\zeta}^0_{k_1,t_1^0},t_2^1\leq\overline{\zeta}^0_{k_2,t_2^0}}\Pr(u^1_{k_1}=t_1^1,u^1_{k_2}=t_2^1)\Pr(u^0_{k_1}=t_1^0,u^0_{k_2}=t_1^1)\Big]. \end{aligned} \tag{2.22}$$

Proof. Please refer to A.2. Q.E.D.

The joint distribution of $\underline{\zeta}^0_{k_1,t_1^0}$ and $\underline{\zeta}^0_{k_2,t_2^0}$ (similarly for $\underline{\zeta}^1_{k_1,t_1^1}$ and $\underline{\zeta}^1_{k_2,t_2^1}$) and the joint distribution of $\underline{\zeta}^0_{k_1,t_1^0}$ and $\overline{\zeta}^0_{k_1,t_2^0}$ (similarly for $\underline{\zeta}^1_{k_1,t_1^1}$ and $\overline{\zeta}^1_{k_2,t_2^1}$), which depend on the uncertainty classes, are given in A.3 for $\varepsilon-$contamination and $p-$point classes.

2.3 The Regularization Parameter

The regularization parameter λ_y in (2.4) should be adjusted based on the relative uncertainty between the training data and the prior knowledge. We propose three approaches for tuning the regularization parameter.

2.3.1 Minimizing the Expected True Error

The optimal value of the regularization parameter, based on expected true error, can be found by solving the following optimization problem:

$$\boldsymbol{\lambda}^* = \arg \min_{\mathbf{0} \preceq \boldsymbol{\lambda} \preceq \mathbf{1}} \mathrm{E}(\epsilon_{\mathbf{RML}}), \tag{2.23}$$

where $\boldsymbol{\lambda} = [\lambda_0, \lambda_1]$, $\mathbf{1} = [1, 1]$, $\mathbf{0} = [0, 0]$ and $\mathrm{E}(\epsilon_{\mathbf{RML}})$ is given in equation (2.16). In (2.16), the only parameters affected by $\boldsymbol{\lambda}$ are $\Pr(\underline{\zeta}^y_{k,l_y} = j), y \in \{0, 1\}$, approximated in Theorems 1 and 2. (2.23) is a constrained non-linear programming problem whose global minimum is not guaranteed to be found by classic gradient-based methods.

2.3.2 SURE-tuning of Regularization Parameter

One way to evaluate the performance of the estimator in Lemma 1 is to use the mean-squared error (MSE) of the estimator. In the problem of multinomial distribution estimation, the MSE can be expanded as follows

$$\mathrm{MSE}^y = \mathrm{E}\Big[\sum_{k=1}^{b} \Big[\hat{\pi}^y_{\lambda_y}(k) - \pi^y_{ac}(k) \Big]^2 \Big], y = 0, 1, \tag{2.24}$$

where we drop the subscript RML and instead use the regularization parameter λ_y to show that the estimate depends on λ_y. One strategy to find the regularization parameter is to minimize MSE^y in (2.24) [27,66]; however, MSE^y depends on the parameter for estimating $\boldsymbol{\pi}^y_{ac}$. We use an approach called SURE (Stein's Unbiased Risk Estimator) [67], proposed for the i.i.d. Gaussian model. Here, an unbiased estimate of the MSE of the designed estimator is found and then one can do optimization to find the required parameters of the estimator. For the sake of simplicity, in the following lemma we omit the superscript y.

Lemma 2. *Let the uncertainty class,* Π*, be given and fixed. Denoting the RML estimator of* $\boldsymbol{\pi_{ac}}$ *in Lemma 1 using* λ *as the regularization parameter by* $\hat{\boldsymbol{\pi}}_\lambda$*, an unbiased estimate of the MSE of the estimate in Lemma 1 is given by*

$$\hat{MSE} = \sum_{k=1}^{b} \Big[\hat{\pi}_\lambda^2(k) + \pi_{ac}^2(k) - 2\{ \frac{\delta_\lambda}{n-1} u_k^2 - u_k (\frac{\delta_\lambda}{n-1} - \frac{\theta_\lambda(k)}{n}) \} \Big] \tag{2.25}$$

where $\delta_\lambda = \frac{1-\lambda}{(1-\lambda)n+\lambda}$ *and* $\theta_\lambda(k) = \frac{\lambda \bar{\pi}(k)}{(1-\lambda)n+\lambda}$.

Proof. Although we took a standard approach to find the unbiased estimator of the MSE, in this part, for simplicity, we only show that $\mathrm{E}(\hat{\mathrm{MSE}}) = \mathrm{MSE}$ (it is sufficient for the proof), where MSE can be expanded as follows

$$\mathrm{MSE} = \mathrm{E}(\sum_{k=1}^{b} \Big[\hat{\pi}_\lambda(k) - \pi_{ac}(k) \Big]^2) = \sum_{k=1}^{b} \mathrm{E} \left[\hat{\pi}_\lambda^2(k) + \pi_{ac}^2(k) - 2\hat{\pi}_\lambda(k) \pi_{ac}(k) \right]$$

The first and the second terms in the right summation do not need any manipulation. Therefore, in the remainder of the proof, we focus on the last term in the right summation. Using the definitions of δ_λ and $\theta_\lambda(k)$, and the fact that $\mathrm{E}(u_k) = n\pi_{ac}(k)$, we have

$$\mathrm{E} \Big[\sum_{k=1}^{b} \hat{\pi}_\lambda(k) \pi_{ac}(k) \Big] = \sum_{k=1}^{b} (\delta_\lambda n \pi_{ac}(k) + \theta_\lambda(k)) \pi_{ac}(k)$$

Now, we return to the $\hat{\mathrm{MSE}}$ in Lemma 2 and take the expectation of the last term in the summation (the term multiplied by 2). We obtain

$$\begin{aligned} \mathrm{E} \left[\sum_{k=1}^{b} \frac{\delta_\lambda}{n-1} u_k^2 - \sum_{k=1}^{b} u_k (\frac{\delta_\lambda}{n-1} - \frac{\theta_\lambda(k)}{n}) \right] &= \sum_{k=1}^{b} \frac{\delta_\lambda}{n-1} [n(n-1)\pi_{ac}^2(k) + n\pi_{ac}(k)] \\ &\quad - \sum_{k=1}^{b} n\pi_{ac}(k) (\frac{\delta_\lambda}{n-1} - \frac{\theta_\lambda(k)}{n}). \end{aligned} \tag{2.26}$$

in which we used the terms for the first and second-moments of the multinomial

distribution. Some simplification completes the proof. Q.E.D.

Minimizing the SURE-estimate of the MSE with respect to the regularization parameter λ yields the following result for case of $n \geq 2$.

Corollary 1 (SURE-Optimal Regularization Parameter)**.** *The SURE-optimal regularization parameter of the estimator defined in Lemma 1 is given by*

$$\lambda^*_{SURE} = \begin{cases} \tilde{\lambda} & 0 \leq \tilde{\lambda} \leq 1 \\ I_{\sum_{k=1}^{b}\left[\overline{\pi}(k)^2 - 2\frac{u_k \overline{\pi}(k)}{n}\right] < \frac{2}{n-1} - \frac{n+1}{n^2(n-1)}\sum_{k=1}^{b} u_k^2} & \textit{otherwise} \end{cases} \tag{2.27}$$

in which we have $\tilde{\lambda} = \dfrac{n\left[1 - \sum_{k=1}^{b}(u_k/n)^2\right]}{(n-1)\left[1 + \sum_{k=1}^{b}\overline{\pi}(k)\left[\overline{\pi}(k) - 2u_k/n\right]\right]}$.

Proof. The corollary results from equating the derivative of (2.25) (with respect to λ) to zero, while considering the boundary of the feasible region of the λ (the SURE estimate in equation (2.25) is continuous in $[0, 1]$). Q.E.D.

Fixing the uncertainty class, as $n \to \infty$, we obtain

$$\lim_{n\to\infty} \lambda^*_{SURE} = \frac{1 - ||\boldsymbol{\pi}_{ac}||_2^2}{1 - ||\boldsymbol{\pi}_{ac}||_2^2 + ||\boldsymbol{\pi}_{ac} - \overline{\boldsymbol{\pi}}||_2^2}, \tag{2.28}$$

in which $||\mathbf{x}||_2$ denotes the ℓ_2-norm of vector $\mathbf{x}$.

To illustrate the effects of different sample sizes and different amounts of uncertainty on λ^*_{SURE}, we have run a simulation assuming an $\varepsilon-$contamination uncertainty class and that the actual distribution follows a Zipf model with parameter $a = 1$ (a detailed description of the Zipf model will be provided in Section 2.4). We observe the behavior of $\overline{\lambda}_{SURE} = \mathrm{E}_{\Pi}\left[\mathrm{E}_{S_n}[\lambda^*_{SURE}|\Pi]\right]$ using Monte-Carlo expectation over $25,000$

pairs of training data sets (for each fixed sample size) and uncertainty classes. We consider different values for $\varepsilon \in [0, 1)$ and sample size n. Figure 2 shows the 3-D figure with n as the x-axis and ε as the y-axis. As $\varepsilon \to 1$ (uncertainty is increased), for a fixed sample size, $\overline{\lambda}_{SURE}$ decreases as in equation (2.28).

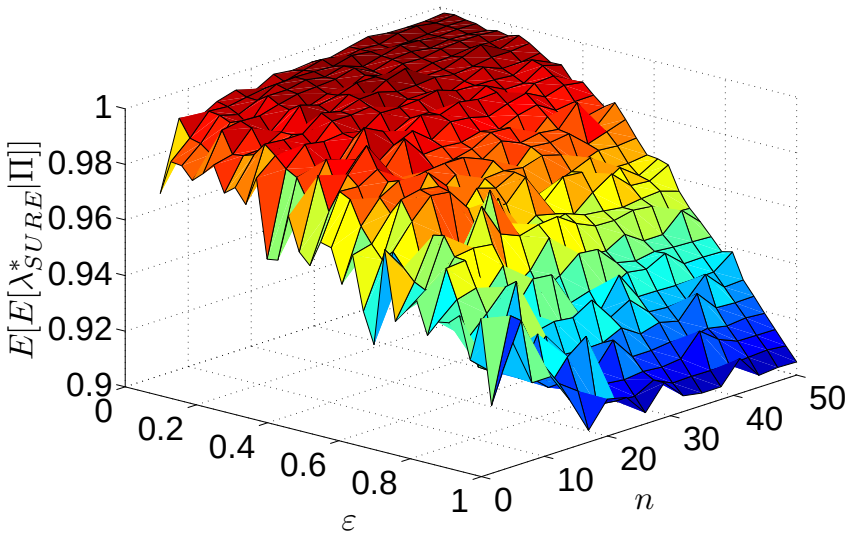


Figure 2.1: Illustrating the expected value of λ^*_{SURE} for different amount of uncertainty and sample sizes. The result is for $\varepsilon-$contamination classes. The uncertainty class size, $|\Pi|$ is set to 50.

In order to remove the effect of sample size on the range of log-likelihood function, we compute the *normalized regularization parameter*: $\mathrm{E}_{\Pi}\Big[\mathrm{E}_{S_n}[\lambda^*_{SURE}|\Pi]\Big]/n$ via Monte-Carlo simulations. The results are summarized for some of the cases in Table 2.1.

Table 2.1 shows that as n increases, removing the actual dependency to n, the SURE-optimal regularization parameter decreases, meaning that the information in the data become more and more reliable. Moreover, as the contamination increases

Table 2.1: The normalized SURE-optimal regularization parameter. The simulation setup is identical to the one used for Figure 2.1.

n \ ε	0.1	0.25	0.40	0.55	0.70	0.85	1
5	0.1948	0.1940	0.1916	0.1923	0.1893	0.1899	0.1889
14	0.0705	0.0706	0.0703	0.0693	0.0685	0.0682	0.0655
23	0.0433	0.0432	0.0430	0.0420	0.0417	0.0407	0.0396
32	0.0311	0.0310	0.0308	0.0304	0.0299	0.0292	0.0283
41	0.0243	0.0242	0.0241	0.0236	0.0233	0.0227	0.0223
50	0.0199	0.0199	0.0197	0.0194	0.0190	0.0186	0.0181

the regularization parameter decreases, though not substantially.

2.3.3 A Heuristic Approach

Although one can use a stochastic algorithm to solve (2.23) (which is not necessarily guaranteed to achieve the global minimum), or use the result in Corollary 1, we can take a heuristic approach for specifying λ_y. Suppose $|\Pi^0|$ and $|\Pi^1|$ are the sizes of the uncertainty classes for labels 0 and 1, respectively. Proceeding heuristically and denoting the ith distribution with label y as $\boldsymbol{\pi}_i^y$, we form a network-based estimate, $\hat{\boldsymbol{\pi}}_{\mathbf{uc}}^y = \overline{\boldsymbol{\pi}}^y$, by averaging the $\boldsymbol{\pi}_i^y$, $i = 1, ..., |\Pi^y|$. A data-based estimate, $\hat{\boldsymbol{\pi}}_{\mathbf{data}}^y$, is obtained from (2.3). Under this setting, we can estimate the relative uncertainty by

$$\lambda_y := \frac{\mathrm{Var}(\hat{\boldsymbol{\pi}}_{\mathbf{data}}^y)}{\mathrm{Var}(\hat{\boldsymbol{\pi}}_{\mathbf{data}}^y) + \mathrm{Var}(\hat{\boldsymbol{\pi}}_{\mathbf{uc}}^y)}. \tag{2.29}$$

where

$$\begin{aligned} \mathrm{Var}(\hat{\boldsymbol{\pi}}_{\mathbf{data}}^y) &= \sum_{k=1}^{b} \mathrm{Var}(\hat{\pi}_{\mathbf{data}}^y(k)) \\ \mathrm{Var}(\hat{\boldsymbol{\pi}}_{\mathbf{uc}}^y) &= \sum_{k=1}^{b} \mathrm{Var}(\hat{\pi}_{\mathbf{uc}}^y(k)). \end{aligned} \tag{2.30}$$

In (2.30), the variance of the training data is independent of the uncertainty class model and can therefore be analytically computed by

$$\mathrm{Var}(\hat{\boldsymbol{\pi}}^y_{\mathbf{data}}) = \sum_{k=1}^{b} \frac{\pi^y_{ac}(k)(1-\pi^y_{ac}(k))}{n_y}. \tag{2.31}$$

The variance of the uncertainty class depends on the underlying model of the uncertainty class. We obtain

$$\mathrm{Var}(\hat{\boldsymbol{\pi}}^y_{\mathbf{uc}}) = \frac{\varepsilon_y^2 b(b-1)}{(b+1)b^2}. \tag{2.32}$$

for a $\varepsilon-$contamination class and

$$\mathrm{Var}(\hat{\boldsymbol{\pi}}^y_{\mathbf{uc}}) = \sum_{p=1}^{m_y} \frac{\omega^y_p |s^y_p|(|s^y_p|-1)}{(|s^y_p|+1)|s^y_p|^2}. \tag{2.33}$$

for a $p-$point uncertainty class (please refer to Section 1 of the supplementary materials on the companion website).

2.4 Numerical Experiments

In this section, we evaluate the performance of the classifiers designed using the proposed optimization paradigm. Let $\epsilon_{\mathbf{RML}}$ denote the error of the RML classifier designed via (2.4) using the estimated probabilities given in Lemma 1. Let $\epsilon_{\mathbf{hist}}$ denote the error of the traditional histogram rule obtained by designing the classifier as in (2.1) using the data-based estimate $\hat{\boldsymbol{\pi}}^y_{\mathbf{data}}$ given in (2.3). The exact expression for $\mathrm{E}(\epsilon_{\mathbf{hist}})$ is given in [35].

We use both the approximation in (2.16) as well as Monte Carlo simulations for assessing $\mathrm{E}(\epsilon_{\mathbf{RML}})$. In the Monte-Carlo estimation, based on the given assumption for the structure of the uncertainty classes, we generate T pairs of uncertainty classes denoted by $(\Pi^0_l, \Pi^1_l), l = 1, ..., T$. Then for each pair, based on the given model for

the true distributions $\boldsymbol{\pi}_{ac}^{y}, y = 0, 1$, we generate M sample sets with size n denoted by $S_n^{l,m}, m = 1, ..., M$. For each sample $S_n^{l,m}$, we estimate the conditional probabilities using Lemma 1. The estimates $\hat{\pi}_{\mathbf{RML}}^{y}(k)$ are then used to construct the classifier, as defined in (2.1). The error of the classifier designed using $S_n^{l,m}$ (i.e., mth sample set generated for the lth pair) is then computed analytically using the actual distribution $\boldsymbol{\pi}_{ac}^{y}$ which was used to generate the sample. We denote this error by $\epsilon_{\mathbf{RML}}^{l,m}$. The first- and the second-order moments of the true error are approximated by

$$\mathrm{E}(\epsilon_{\mathbf{RML}}) \approx \frac{1}{MT} \sum_{l=1}^{T} \sum_{m=1}^{M} \epsilon_{\mathbf{RML}}^{l,m}, \tag{2.34}$$

$$\mathrm{E}(\epsilon_{\mathbf{RML}}^{2}) \approx \frac{1}{MT} \sum_{l=1}^{T} \sum_{m=1}^{M} (\epsilon_{\mathbf{RML}}^{l,m})^2 \tag{2.35}$$

via Monte Carlo simulation. We estimate the variances, $\mathrm{Var}(\hat{\boldsymbol{\pi}}_{\mathbf{data}}^{y})$ and $\mathrm{Var}(\hat{\boldsymbol{\pi}}_{\mathbf{uc}}^{y})$ in (2.29) as

$$\hat{\mathrm{Var}}(\hat{\boldsymbol{\pi}}_{\mathbf{data}}^{y}) = \sum_{k=1}^{b} \frac{\frac{u_k^y}{n_y}(1 - \frac{u_k^y}{n_y})}{n_y}, \tag{2.36}$$

$$\hat{\mathrm{Var}}(\hat{\boldsymbol{\pi}}_{\mathbf{uc}}^{y}) = \frac{1}{|\Pi^y| - 1} \sum_{k=1}^{b} \sum_{i=1}^{|\Pi^y|} (\pi_i^y(k) - \hat{\pi}_{\mathbf{uc}}^{y}(k))^2. \tag{2.37}$$

2.4.1 Performance Assessment Using a Zipf Model

We first assume that the true label-conditional distributions (i.e., $\boldsymbol{\pi}_{ac}^{y}, y = 0, 1$) follow a Zipf model,

$$\pi_{ac}^{0}(k) = \frac{\xi}{k^a}, \quad \pi_{ac}^{1}(k) = \pi_{ac}^{0}(b - k + 1), \tag{2.38}$$

where ξ is a normalizing constant. The Zipf distribution, originally introduced by G.K. Zipf to model the frequency of words in common text [68], is a well-known

power-law discrete distribution, encountered in many applications. In particular, it has been used as a model to study the moments of error estimators for discrete classifiers [35]. As $a \to 0$, both conditional distributions ($y \in \{0, 1\}$) tend to become uniform. Hence the classification problem becomes more difficult, resulting in a larger Bayes error. We examine the performance under two sampling scenarios: stratified sampling (i.e. sampling according to a known $\Pr(Y = 0) = c$), and random sampling. We consider two bin sizes $b \in \{8, 16\}$ (which, respectively correspond to the number of states in a three-gene and four-gene Boolean network when modeling genomic regulatory networks [69]). We evaluate the proposed framework under two different scenarios. First, we examine the accuracy of our approximate expressions by comparing them with the Monte-Carlo simulation while one has access to the exact regularization parameters defined by applying (2.31)-(2.33). The motivation is to test the accuracy of our approximation when the regularization parameters are found off-line, independent of the given sample data. In the second scenario, we assume one has to estimate the regularization parameters based on the given data and uncertainty classes using Corollary 1. Depending on the underlying assumption for the uncertainty classes, for each size n and each set of model parameters (e.g., ε_0, ε_1, or partitions in the $p-$point class), we generate $T = 1000$ different pairs of uncertainty classes, $(\Pi_l^0, \Pi_l^1), l = 1, ..., 1000$, for which we generate $M = 2,000$ samples, $S_n^{l,m}, l = 1, ..., 1000; m = 1, ..., 2000$, for estimating the first- and the second-order moments of the true error, $\mathrm{E}(\epsilon_{\mathbf{RLM}})$ and $\mathrm{E}(\epsilon_{\mathbf{RLM}}^2)$. For the approximate second-order moments, where there are double-integrals, we use the adaptive Simpson algorithm for approximating the integral values. Results for the various experiments are shown in Figures 2.2-2.7. Two Zipf parameters $a = 0.5$ and 1 are examined, which depending on the bin size, yields different Bayes errors. Different parameters along with their counterparts in the figures are summarized in Table 2.2.

Table 2.2: A summary of the parameters used in the simulations. Two class sizes, $|\Pi^0| = |\Pi^1| \in \{10, 250\}$ are examined.

$b = 8$	$a = 0.5$	$c = 0.4$	$\implies$	$\epsilon_{\text{Bayes}} = 0.3442$: Figures 2.2a, 2.2b, 2.3a, 2.3b, 2.4a, 2.4b, 2.5a, 2.5b
$b = 8$	$a = 1.0$	$c = 0.4$	$\implies$	$\epsilon_{\text{Bayes}} = 0.2261$: Figures 2.2c, 2.2d, 2.3c, 2.3d, 2.4c, 2.4d, 2.5c, 2.5d
$b = 16$	$a = 0.5$	$c = 0.4$	$\implies$	$\epsilon_{\text{Bayes}} = 0.3268$: Figures 2.2e, 2.2f, 2.3e, 2.3f, 2.4e, 2.4f, 2.5e, 2.5f
$b = 16$	$a = 1.0$	$c = 0.4$	$\implies$	$\epsilon_{\text{Bayes}} = 0.1903$: Figures 2.2g, 2.2h, 2.3g, 2.3h, 2.4g, 2.4h, 2.5g, 2.5h
$b = 8$	$a = 0.5$	$c = 0.5$	$\implies$	$\epsilon_{\text{Bayes}} = 0.3630$: Figures 2.6a, 2.6b, 2.7a, 2.7b
$b = 8$	$a = 1.0$	$c = 0.5$	$\implies$	$\epsilon_{\text{Bayes}} = 0.2335$: Figures 2.6c, 2.6d, 2.7c, 2.7d
$b = 16$	$a = 0.5$	$c = 0.5$	$\implies$	$\epsilon_{\text{Bayes}} = 0.3440$: Figures 2.6e, 2.6f, 2.7e, 2.7f
$b = 16$	$a = 1.0$	$c = 0.5$	$\implies$	$\epsilon_{\text{Bayes}} = 0.1961$: Figures 2.6g, 2.6h, 2.7g, 2.7h

Table 2.3: Two settings for $p-$point uncertainty classes for any fixed bin size, b.

Label y	bin size (b)	Partition: $\mathcal{P}_1^y$	Partition: $\mathcal{P}_2^y$
0	8	$\{\{1,2,3,4\},\{5,6,7,8\}\}$	$\{\{1,2\},\{3,4\},\{5,6,7,8\}\}$
1	8	$\{\{1,3,5,7\},\{2,4,6,8\}\}$	$\{\{1,3\},\{5,7\},\{2,4,6,8\}\}$
0	16	$\{\{1,2,3,4,5,6,7,8\},$ $\{9,10,11,12,13,14,15,16\}\}$	$\{\{1,2,3,4\},\{5,6,7,8\},$ $\{9,10,11,12\},\{13,14,15,16\}\}$
1	16	$\{\{1,3,5,7,9,11,13,15\},$ $\{2,4,6,8,10,12,14,16\}\}$	$\{\{1,3,5,7\},\{2,4,6,5\},$ $\{9,11,13,15\},\{10,12,14,16\}\}$

We use the algorithm proposed in [70] for generating the contaminating distribution generated uniformly under a unit-simplex. Three cases are considered for the pair: $(\varepsilon_0, \varepsilon_1)$: $(0.3, 0.3)$, $(0.6, 0.6)$, and $(0.8, 0.8)$. The settings for the $p-$point uncertainly class models for which the RML classifier performance is assessed are summarized in Table 2.3. Fixing b, settings are designed such that the uncertainty in the classes corresponding to setting 1, $\mathcal{P}_1^y; y \in \{0, 1\}$, is always greater than that of setting 2. The reason is that there is one subset in setting 1 which contains one of the partition subsets of setting 2. In other words, the information in setting 2 is always greater than that of setting 1.

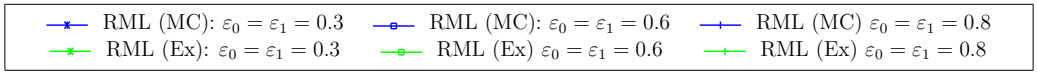
Although we consider $c = 0.5$ in the random sampling scenarios, the stratified case with $c = 0.5$ under separate sampling is not studied. The reason is that, due

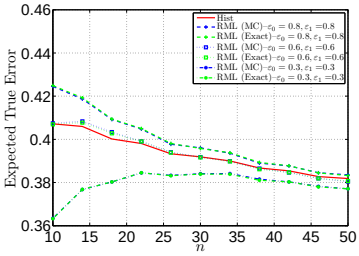
to the symmetry considered for the uncertainty classes, described above, the final classifiers become trivial and identical to that of histogram rule. Therefore, we fix the class prior probability $c = 0.4$ in the following whenever we deal with stratified sampling. It should be noted that for the SURE-optimal regularization parameter case, the classifiers would not be identical to the histogram. Nonetheless, for the sake of fair comparison, we fix $c = 0.4$ for all the separate sampling settings.

Figures 2.2-2.3 show the results for the first scenario, under the separate sampling scenario, for the $\varepsilon-$contamination and $p-$point uncertainty classes, respectively. The expected true error of the proposed scheme is smaller than that of the histogram rule in almost all the cases. Nonetheless, since the regularization parameters, in these two figures, are set prior to observing data or uncertainty classes, increasing the contamination factor leads to poor performance, even compared to the histogram rule. Moreover, the results from the Monte-Carlo simulations are very close to those obtained from Theorems 1 and 2, shown by "Ex" in the legends of plots.

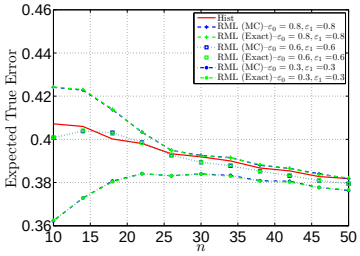
Next, we examine the performance when the regularization parameter is chosen using Corollary 1 and the sampling is stratified based on the true class prior probability, i.e. $c = 0.4$. The results are shown in Figures 2.4 and 2.5 for $\varepsilon-$contamination and $p-$point classes, respectively. Comparing Figures 2.4 with 2.2, and similarly 2.5 with 2.3, one can see that selecting data-uncertainty-class-dependent regularization parameters significantly improves the classification accuracy using the RML rule.

Moreover, Figure 2.6 shows that, by increasing the contamination factor from 0.3 to 0.8, the expected true error of the RML classifier increases, being a direct result of "inaccurate" prior information. Nonetheless, despite of having this degradation, owing to adaptive selection of the regularization parameters, the RML classifier still outperforms the histogram rule in most of the cases. Figure 2.7 shows similar behavior for the $p-$point classes. Again, it can be inferred, in all the cases, the RML

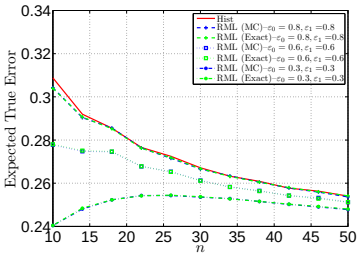




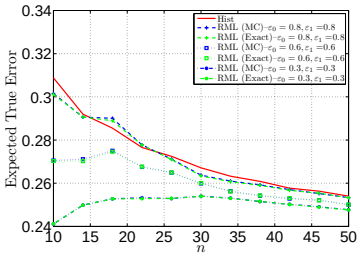
(a) $b = 8$, $a = 0.5$, $|\Pi^0| = |\Pi^1| = 10$



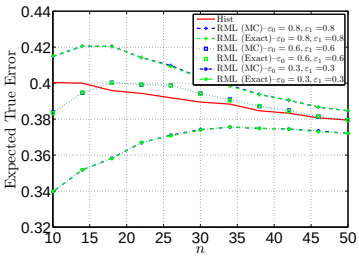
(b) $b = 8$, $a = 0.5$, $|\Pi^0| = |\Pi^1| = 250$



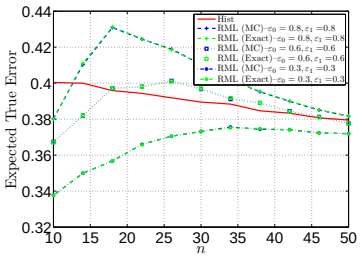
(c) $b = 8$, $a = 1$, $|\Pi^0| = |\Pi^1| = 10$



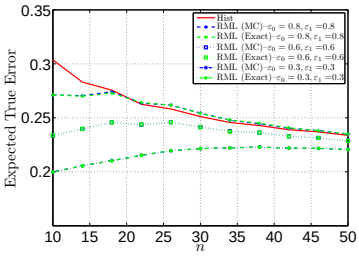
(d) $b = 8$, $a = 1$, $|\Pi^0| = |\Pi^1| = 250$



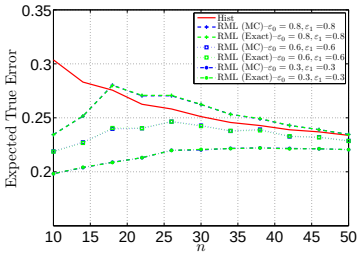
(e) $b = 16$, $a = 0.5$, $|\Pi^0| = |\Pi^1| = 10$



(f) $b = 16$, $a = 0.5$, $|\Pi^0| = |\Pi^1| = 250$

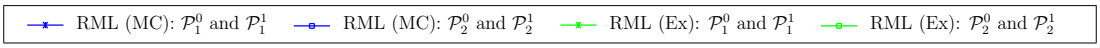


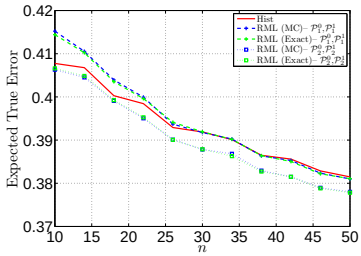
(g) $b = 16$, $a = 1$, $|\Pi^0| = |\Pi^1| = 10$



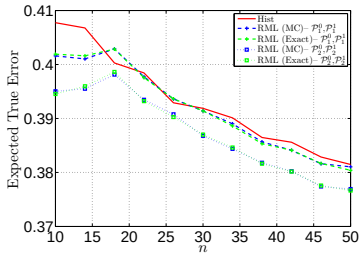
(h) $b = 16$, $a = 1$, $|\Pi^0| = |\Pi^1| = 250$

Figure 2.2: Results for the histogram and RML rules as a function of n, with $c = 0.4$, under stratified sampling, for $\varepsilon-$contamination classes, and fixed regularization parameter computed as in (2.29).

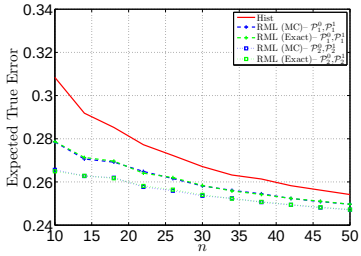




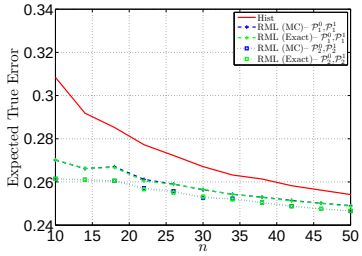
(a) $b = 8$, $a = 0.5$, $|\Pi^0| = |\Pi^1| = 10$



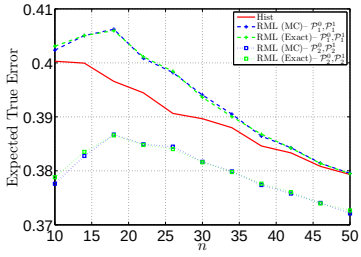
(b) $b = 8$, $a = 0.5$, $|\Pi^0| = |\Pi^1| = 250$



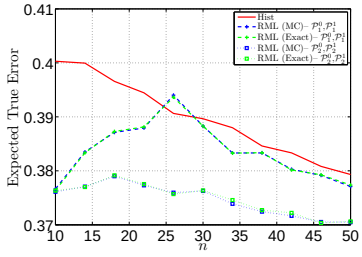
(c) $b = 8$, $a = 1$, $|\Pi^0| = |\Pi^1| = 10$



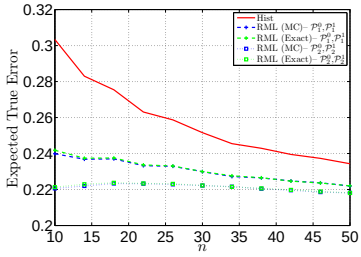
(d) $b = 8$, $a = 1$, $|\Pi^0| = |\Pi^1| = 250$



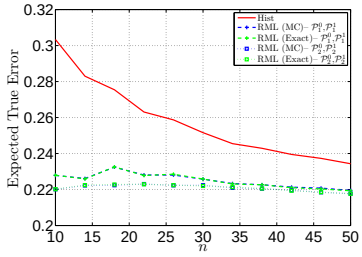
(e) $b = 16$, $a = 0.5$, $|\Pi^0| = |\Pi^1| = 10$



(f) $b = 16$, $a = 0.5$, $|\Pi^0| = |\Pi^1| = 250$



(g) $b = 16$, $a = 1$, $|\Pi^0| = |\Pi^1| = 10$



(h) $b = 16$, $a = 1$, $|\Pi^0| = |\Pi^1| = 250$

Figure 2.3: Results for the histogram and RML rules as a function of n, with $c = 0.4$, under stratified sampling, for $p-$point classes, and fixed regularization parameter computed as in (2.29).

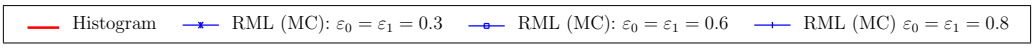
classifier outperforms the histogram.

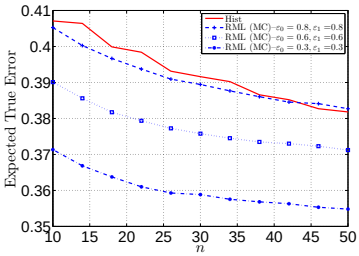
All the simulations so far have been resulted from stratified sampling with true class prior probability, $c = 0.4$ leading to $n_0 = n_1$. While the focus in Figures 2.3-2.3 were mainly to validate our analytical results, given by Theorems 1-2, now we examine the performance when the sampling is random, with $c = 0.5$, and all regularization parameters are chosen using Corollary 1. The results are shown in Figures 2.6-2.7 for $\varepsilon-$contamination and $p-$point classes, respectively.

2.5 Performance Assessment Using Networks Containing NF-κB pathways

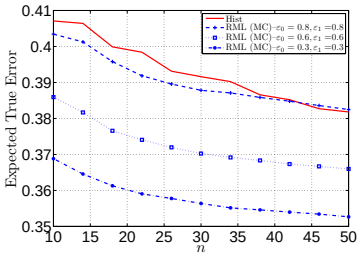
While the theoretical development of this chapter pertains to uncertainty classes of distributions for classification, as stated at the outset, our original motivation for the theory comes from our desire to apply prior pathway knowledge in biological network steady-state classification.

In this section, we use prior pathway knowledge and an associated cellular context in order to improve the performance of a classifier which discriminates between biologically relevant states of a biological system. More specifically, a biological system can be modeled by a discrete, dynamical system that is subject to external stimuli and behaves according to interactions amongst its constitutive components. These interactions between components are often referred to as pathways and are time invariant in most biological processes. It is instead the varying cellular context that activates or deactivates pathways in order for a cell to respond to the demands of life. For many classification problems of interest and this example here, these pathways will be identical in each class and it is the cellular context of available nutrients, signaling proteins, or other agents that are of interest. However, the general method can be used with differing pathways if the goal is to discriminate against such things as the presence of mutations, separate organisms, or cancer. In all of these exam-

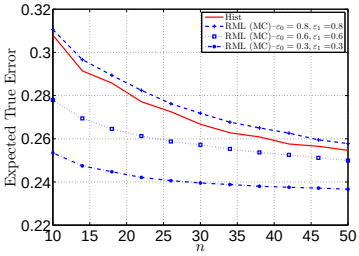




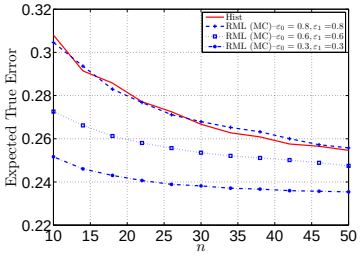
(a) $b = 8$, $a = 0.5$, $|\Pi^0| = |\Pi^1| = 10$



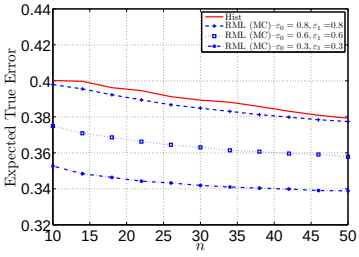
(b) $b = 8$, $a = 0.5$, $|\Pi^0| = |\Pi^1| = 250$



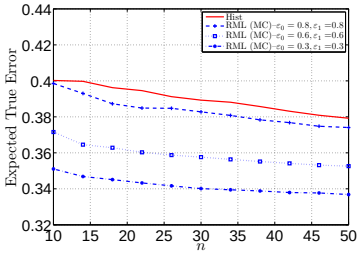
(c) $b = 8$, $a = 1$, $|\Pi^0| = |\Pi^1| = 10$



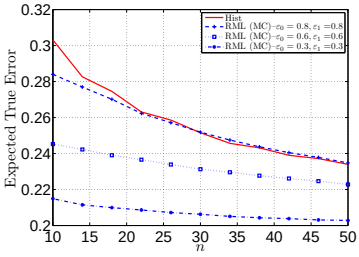
(d) $b = 8$, $a = 1$, $|\Pi^0| = |\Pi^1| = 250$



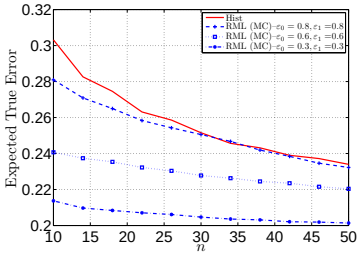
(e) $b = 16$, $a = 0.5$, $|\Pi^0| = |\Pi^1| = 10$



(f) $b = 16$, $a = 0.5$, $|\Pi^0| = |\Pi^1| = 250$

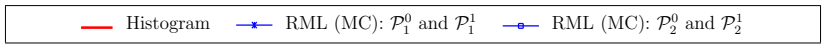


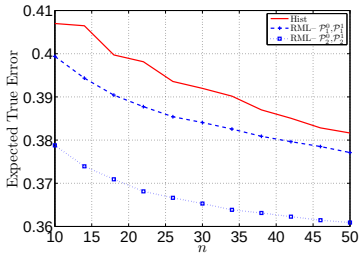
(g) $b = 16$, $a = 1$, $|\Pi^0| = |\Pi^1| = 10$



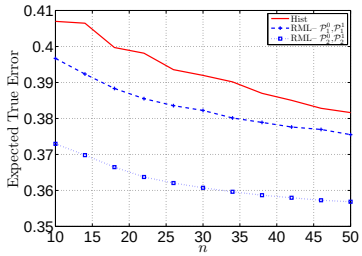
(h) $b = 16$, $a = 1$, $|\Pi^0| = |\Pi^1| = 250$

Figure 2.4: Results for the histogram and RML rules as a function of n, with $c = 0.4$, under stratified sampling, for $\varepsilon-$contamination classes, and SURE-optimal regularization parameters.

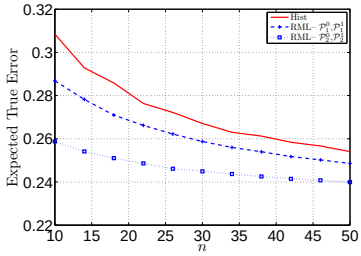




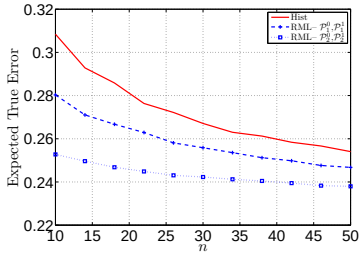
(a) $b = 8$, $a = 0.5$, $|\Pi^0| = |\Pi^1| = 10$



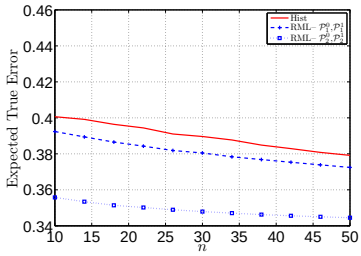
(b) $b = 8$, $a = 0.5$, $|\Pi^0| = |\Pi^1| = 250$



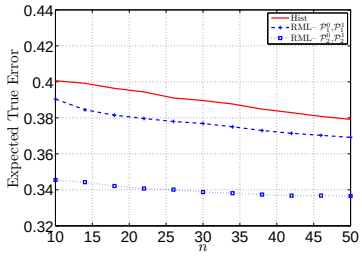
(c) $b = 8$, $a = 1$, $|\Pi^0| = |\Pi^1| = 10$



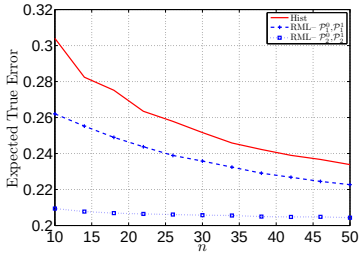
(d) $b = 8$, $a = 1$, $|\Pi^0| = |\Pi^1| = 250$



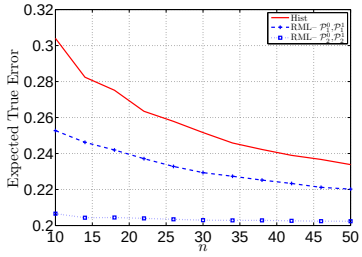
(e) $b = 16$, $a = 0.5$, $|\Pi^0| = |\Pi^1| = 10$



(f) $b = 16$, $a = 0.5$, $|\Pi^0| = |\Pi^1| = 250$

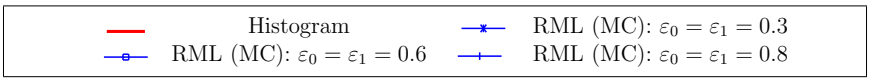


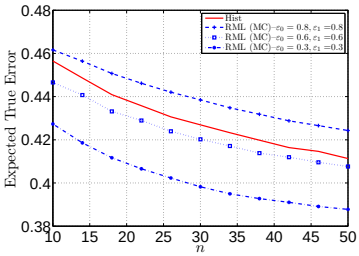
(g) $b = 16$, $a = 1$, $|\Pi^0| = |\Pi^1| = 10$



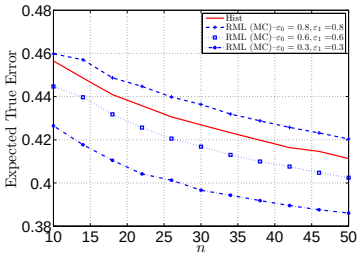
(h) $b = 16$, $a = 1$, $|\Pi^0| = |\Pi^1| = 250$

Figure 2.5: Results for the histogram and RML rules as a function of n, with $c = 0.4$, under stratified sampling, for $p-$point classes, and SURE-optimal regularization parameters.

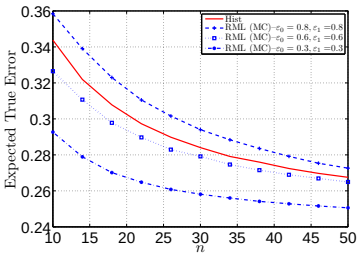




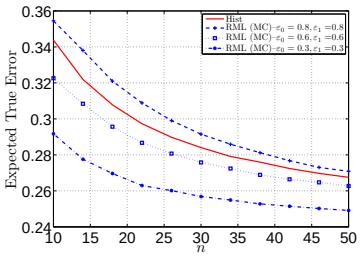
(a) $b = 8$, $a = 0.5$, $|\Pi^0| = |\Pi^1| = 10$



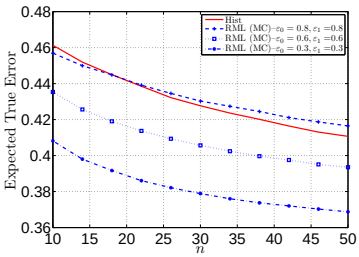
(b) $b = 8$, $a = 0.5$, $|\Pi^0| = |\Pi^1| = 250$



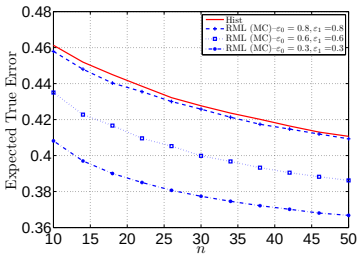
(c) $b = 8$, $a = 1$, $|\Pi^0| = |\Pi^1| = 10$



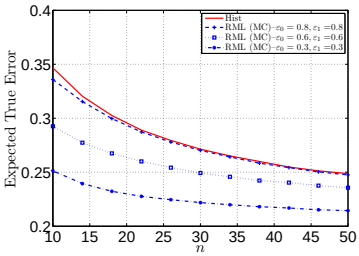
(d) $b = 8$, $a = 1$, $|\Pi^0| = |\Pi^1| = 250$



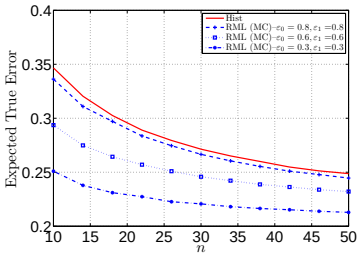
(e) $b = 16$, $a = 0.5$, $|\Pi^0| = |\Pi^1| = 10$



(f) $b = 16$, $a = 0.5$, $|\Pi^0| = |\Pi^1| = 250$

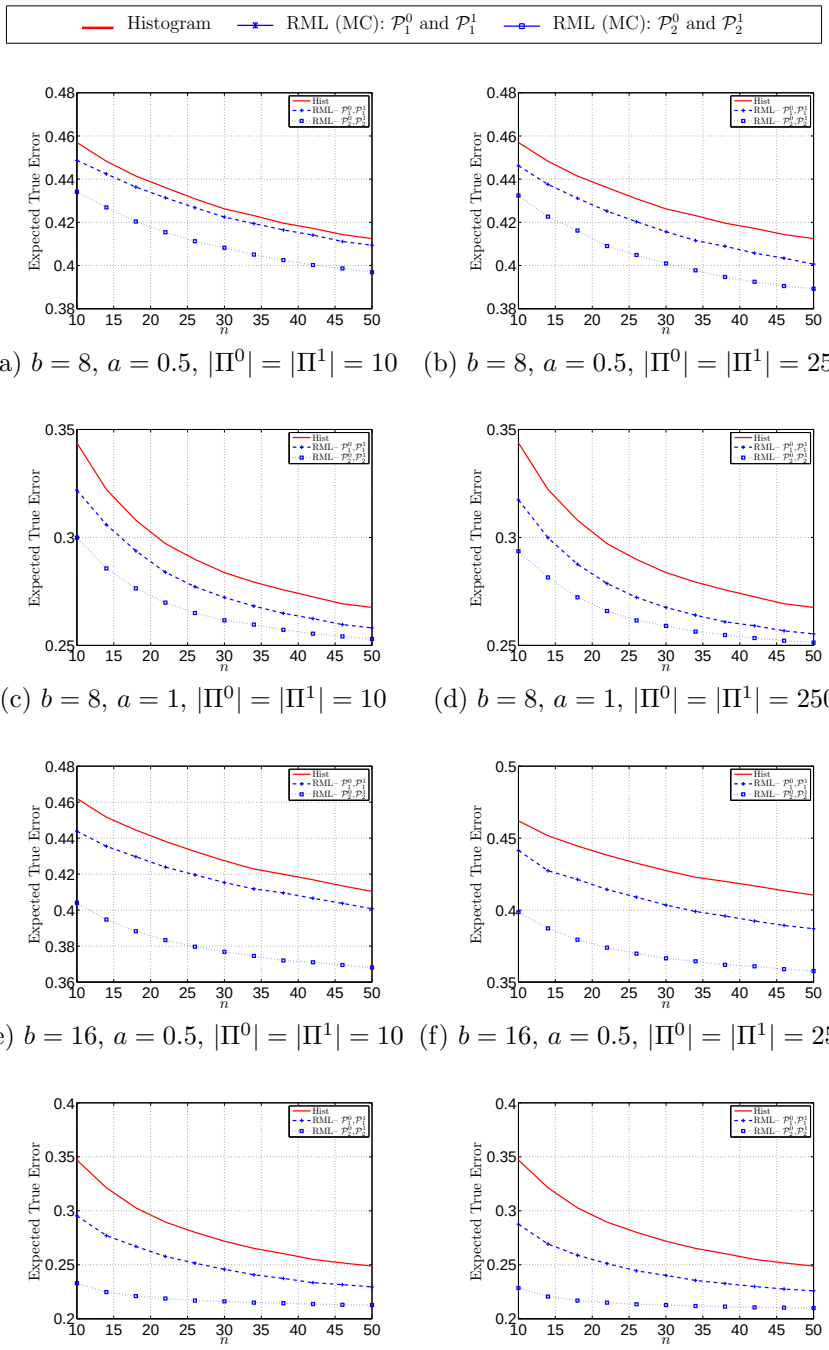


(g) $b = 16$, $a = 1$, $|\Pi^0| = |\Pi^1| = 10$



(h) $b = 16$, $a = 1$, $|\Pi^0| = |\Pi^1| = 250$

Figure 2.6: Expected true error of the histogram and RML rules as a function of total sample size, n, with $c = 0.5$, under random sampling, for $\varepsilon-$contamination classes.



(a) $b = 8$, $a = 0.5$, $|\Pi^0| = |\Pi^1| = 10$ (b) $b = 8$, $a = 0.5$, $|\Pi^0| = |\Pi^1| = 250$

(c) $b = 8$, $a = 1$, $|\Pi^0| = |\Pi^1| = 10$ (d) $b = 8$, $a = 1$, $|\Pi^0| = |\Pi^1| = 250$

(e) $b = 16$, $a = 0.5$, $|\Pi^0| = |\Pi^1| = 10$ (f) $b = 16$, $a = 0.5$, $|\Pi^0| = |\Pi^1| = 250$

(g) $b = 16$, $a = 1$, $|\Pi^0| = |\Pi^1| = 10$ (h) $b = 16$, $a = 1$, $|\Pi^0| = |\Pi^1| = 250$

Figure 2.7: Expected true error of the histogram and RML rules as a function of total sample size, n, with $c = 0.5$, under random sampling, for $p-$point classes.

ples, we would expect the two classes to have different pathways through differing genetics.

To set up the classification example, we use a single set of pathways describing our biological system of interest, and choose two different cellular contexts which describe the biological phenomena we are interested in classifying. Then for each (context, pathways) tuple we generate an intermediate class of dynamical systems that have behavior described by the the biological pathways under this context. These classes represent all possible dynamical systems that can behave according to the constraints of the pathways and cellular context. Each dynamical system in these two classes possesses a unique steady-state distribution, and we can therefore obtain two classes of steady state distributions from our two tuples of (context, pathways).

2.5.1 The NF-κB System

Nuclear factor-κB (NF-κB) is a family of transcription factors that control the expression of over 100 genes. Its primary role is in the immune system as a central regulator of inflammation. This makes it important in cancer research as inflammation contributes to the reduction of apoptosis and increased angiogenesis in the tumor microenvironment [71].

Biologically the NF-κB transcription factor can be activated through several parallel signaling pathways. In this section we use a model containing three stimulating external inputs which are shaded in Figure 2.8. When a bacterial infection occurs, the lipopolysaccharide (LPS) molecule present in the cell wall of the bacteria binds to TLR4 receptors in immune cell membranes and initiates a strong NF-κB response [72]. Tumor necrosis factor α (TNFα) is a cytokine produced primarily by macrophages to induce an endogenous inflammatory response by binding to the TNFR receptor. And finally, NF-κB responses can be initiated through the 'alter-

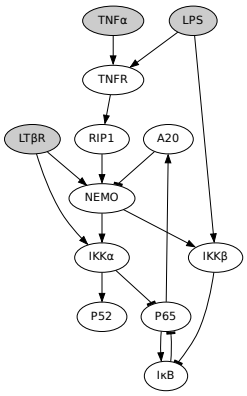


Figure 2.8: The interactions between members of this model are shown using directed edges where an edge from species A to species B indicates that species A regulates species B. Pointed edges represent promoting influences while tee edges represent down regulating influences. LPS, TNFα, and LTβR are shaded indicating their role as external stimuli to the cell. These three inputs provide the cellular context for the model as described in [2].

native pathway' with the lymphotoxin β receptor (LTβR). Once activated, each of these inputs initiates a downstream signaling cascade activating the NF-κB system. As there is no feedback from the system back onto these three external signaling molecules, their state is constant once chosen and helps determine the behavior of the other nine genes.

2.5.2 NF-κB Classification

In a biological system, we are often unable to directly measure or quantify the cellular context which controls the behavior of some cells of interest. We consider such a scenario as a classification problem. Given two possible cellular contexts and some data samples of the 9 proteins whose behaviors are constrained by the context, determine which context the samples were taken from. In Figure 2.9 we graphically depict the two contexts (or classes) in three such classification problems (or configurations). The presence of an input indicates activation, absence indicates

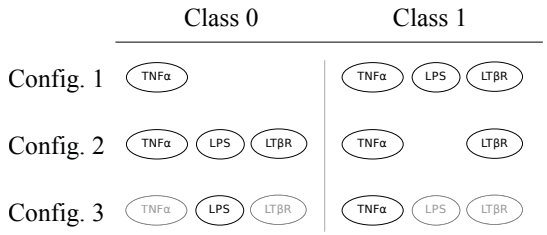


Figure 2.9: The three classification problems (configurations) considered in this section are defined by a pair of biologically interesting cellular contexts. For each configuration we attempt to classify samples as coming from class 0 or class 1 given measurements of the 9 downstream signaling proteins. The presence of an input indicates activation, absence indicates inactivation, and a shaded input indicates the input may either be active or inactive.

inactivation, and a shaded input indicates the input may either be active or inactive.

Qualitatively the three configurations in Figure 2.9 can be described in the following manner: configuration 1 considers an endogenous macrophage induced inflammatory insult in class 0 versus inflammation as a result of bacteria and the response of immune cells in class 1 [72]. Configuration 2 considers an inflammatory insult resulting from bacteria and immune cells in class 0 versus an endogenous inflammatory insult arising from many types of immune cells signaling in class 1. Configuration 3 compares inflammation resulting from a bacterial infection (either in the early stage with no immune cells present or late stage after immune cells have arrived) in class 0 versus an inflammatory injury with immune cells present (possibly resulting from a bacterial infection in class 1).

In these three configurations we measure the ability for the classifier to distinguish the underlying context for an inflammatory response. The classification problem is of significant medical and translational science import.

2.5.3 Modeling the NF-κB System

Previously, we have used pathways collected from the literature to develop and validate a discrete-time, finite-state Markov chain model of the NF-κB system [2]. This method was then generalized in [73] to generate a parameterized class of Markov chains from the pathway knowledge instead of a single Markov chain.

The pathways which define the NF-κB model (which can be seen in [74]) constrain the possible behaviors and interactions of the nine genes. As these pathways are incomplete and sometimes conflicting, the evolution of the Markov chain in some states is often uncertain. We model these uncertainties as independent Bernoulli random variables in the state transition graph with unknown parameters. We then consider the collection of these parameters in the vector $\theta = \{\theta_1, \theta_2, \ldots, \theta_n\}$, where $\theta_i \in [0, 1]$, in order to parameterize the uncertainty class of system behavior.

In the NF-κB model, there are only three uncertainties that arise from the pathways. These determine the parameterization of the uncertainty class via the vector $\theta \in [0, 1]^3$. Choosing θ gives a single well-defined Markov chain from the uncertainty class. For a small example see the companion website (Section 3 of the supplementary materials) and for more details we refer to [2] and [73]. For the true network, we choose a network from [2]. It is at the center of the parameter space, $\theta_{ac} = (0.5, 0.5, 0.5)$. From the standpoint of classification this network is unknown; it is chosen here to generate samples. *A priori* we only know that the true network exists inside our uncertainty class.

2.5.4 Results

To utilize this modeling technique with the proposed RML framework we define two uncertainty classes of models for each configuration by fixing the inputs according to Figure 2.9. Since the RML framework requires finite uncertainty classes, we

discretize the continuous $[0,1]^3$ space as explained in the companion website. Then, adding a perturbation probability $p = 10^{-3}$ in our simulations to each network, we obtain a class of ergodic irreducible Markov chains and, accordingly, a class of steady-state distributions [69]. The perturbation probability for the true model is set to $p = 10^{-5}$. We generate data from the true network in each class. These two data sets along with the two uncertainty classes allow us to compare the RML classification framework with the classical histogram rule.

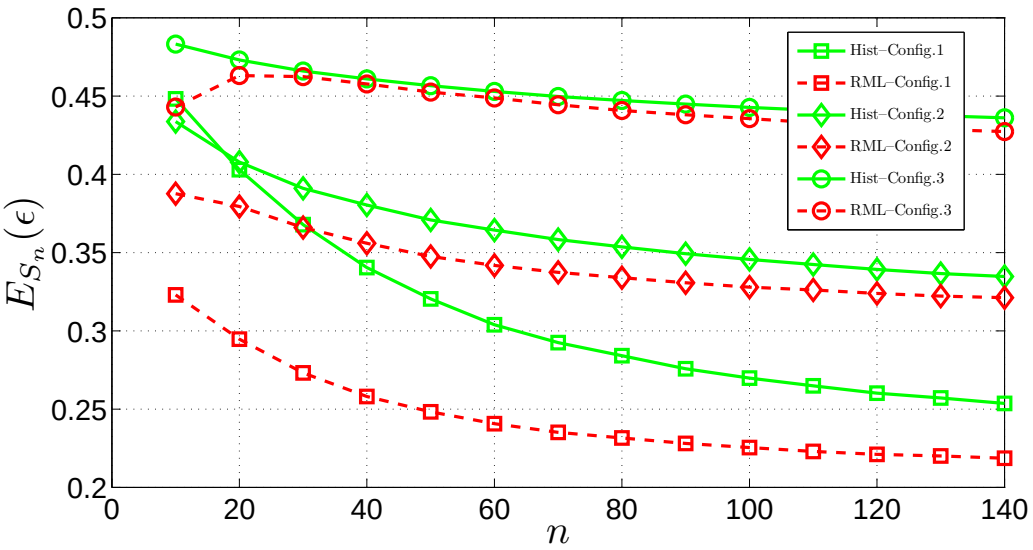


Figure 2.10: Performance comparison between the Histogram-rule and the RML framework. The x axis shows the number of samples n, with $n = n_0 + n_1, n_0 = n_1$. We have $\epsilon_{Bayes} = 0.193$, $\epsilon_{Bayes} = 0.299$, and $\epsilon_{Bayes} = 0.371$ for Configurations 1, 2, and 3, respectively.

Figure 2.10 shows the results for the histogram-rule and proposed method for different configurations. In configuration 3, the error of the classifier briefly increases as

a function of the sample size at the beginning. The regularization parameter is set according to Corollary 1, denoted by λ_{SURE}. Both the histogram and RML classifiers converge to the Bayes errors as $n \to \infty$. In all cases, the RML approach outperforms the histogram-rule, illustrating the benefit of prior knowledge, if available.

2.5.4.1 Comparison to MAP

Designing the RML classifier begins with the assumption of having finite uncertainty classes of feature distributions, in the absence of a prior distribution governing these classes, i.e., no prioritization of any uncertainty class member in favor of the others. Nonetheless, one would still solve the maximum *a posteriori* (MAP) to find the most likely multinomial distribution existing in the uncertainty class and build the "plug-in rule" classifier according to equation (2.1). Hence, using the log-likelihood function in equation (2.2), we define the MAP distribution as

$$\hat{\boldsymbol{\pi}}^y_{\mathbf{MAP}} := \arg\max_{\boldsymbol{\pi}^y \in \Pi^y} \sum_{k=1}^{b} u^y_k \log \pi^y(k). \tag{2.39}$$

Thereafter, we define the MAP classifier by plugging the estimates $\hat{\boldsymbol{\pi}}^y_{\mathbf{MAP}}$ in equation (2.1). In Figure 2.11, we compare performance of the RML given in Lemma 1 with that of MAP given in equation (2.39) by plotting the difference between the corresponding expected true errors, i.e., $\mathrm{E}_{S_n}[\epsilon_{\mathbf{MAP}} - \epsilon_{\mathbf{RML}}]$ as a function of sample size for the three configurations considered in Figure 2.10.

Figure 2.11 illustrates that for configurations 1 and 2 the RML classifier performs always better than the MAP. For category 3, the MAP classifier performs better than the RML in some range, but then, the RML classifier outperforms the MAP after increasing the sample size.

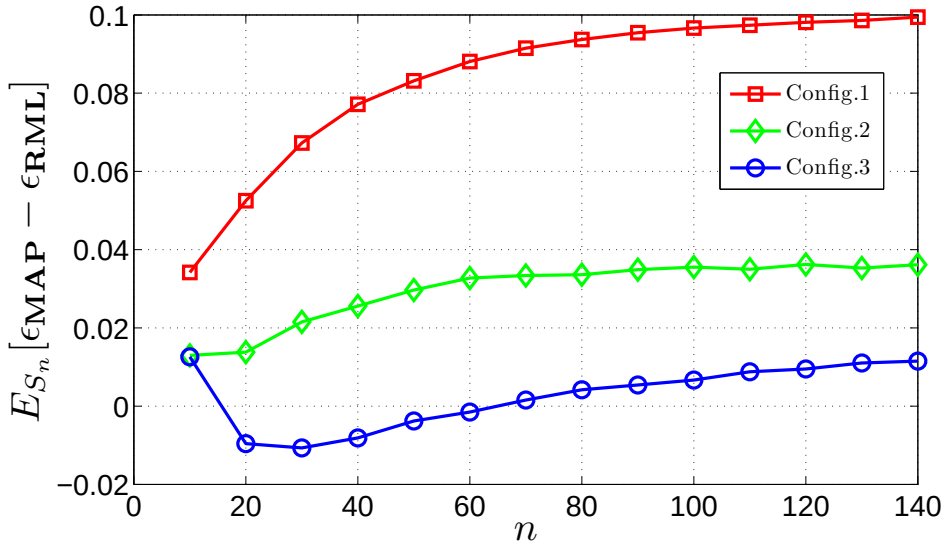


Figure 2.11: Performance comparison between the RML and MAP classifier defined in Lemma 1 and the one designed using estimates in equation (2.39), respectively. The x axis shows the number of samples n.

2.6 Discussion

We have proposed a novel classifier design paradigm that allows us to design enhanced classifiers by incorporating available prior knowledge of the process generating the observation data. As shown in our simulations, such knowledge can significantly improve the performance of the designed classifier, especially, when the sample size is small. Having laid the theoretical groundwork for enhancing steady-state classifier design via the use of prior process knowledge, our plan is to apply the methodology to developing better biomedical classifiers in the presence of partial knowledge of the underlying genetic regulatory network. More generally, given the ubiquity of large feature sets and relatively small sample sizes now common in many disciplines, including medicine, material science, environmental science, and transportation, there will no doubt be an increasing number of methods proposed for using prior knowledge in classifier design. We believe it is important to provide

analytic performance characterization of the classifiers on standard models, as we have done in this work, so that their behavior can be understood.

3. BAYESIAN INFORMATION QUANTIFICATION OF BIOLOGICAL PATHWAYS*

The main task ahead of any pathway knowledge utilization is *knowledge transformation* or *information quantification*. Biological pathways are graphical representations of dependency between molecules. These dependencies are represented by adding *directed* and *regulated* edges. Although these pathways are generated from several *quantitative measurements* (through several *experiments*), the final representation is almost *qualitative*: (1) There is no *timing* associated with them, and (2) Most of the interactions are tested using pairwise experiments. In other words, the experiments' conditions have not necessarily kept the same among different experiments, bringing uncertainty to the problem. On the other hand, modeling a small subset of molecules, is another source of uncertainty: the role of *latent variables*; those which are not taken into account in the modeling.

In this section, we give a formal definition of prior knowledge in the form of pathways. Then, based on the given interpretation of these pathways, to the best of our knowledge, for the first time, a Bayesian information quantification framework is proposed. Doing so, the two-layer uncertainty in these pathways is characterized and transformed to the *hyperparameter space*: the space of prior's parameters.

For the sake of integrity, we denote the entities (e.g. gene or protein), in a discrete setting (i.e., Boolean modeling), contributed in a given set of pathways by x_i (as the i-th element of the feature vector $\mathbf{x}$). In the following section, the continuous case is introduced in which instead of subscript we use $x(i)$ to denote the $i-$th element

*Parts of this section are reprinted with permission from "Incorporation of Biological Pathway Knowledge in the Construction of Priors for Optimal Bayesian Classification" by M. Shahrokh Esfahani and E. R. Dougherty, 2014, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, in press, © 2014 IEEE.

of the vector $\mathbf{x}$. In the following, we consider two cases separately: continuous and discrete case, corresponding to discrete and continuous classifier design problems, respectively. It will be noted that, the discrete case considers a more complete modeling of pathways. The underpinning reason is the difficulty of continuous (Gaussian) case for capturing all the information, while maintaining problem simplicity, to some extent.

3.1 Continuous Model

Define the term "activating pathway segment" (APS) $x(i) \longrightarrow x(j)$ to mean that, if $x(i)$ is "up-regulated" (UR), then $x(j)$ becomes UR (in some time steps). Similarly, the term "repressing pathway segment" (RPS) $x(i) \dashv\!\!\!\rightarrow x(j)$ means that, if $x(i)$ is UR, then $x(j)$ becomes "down-regulated" (DR). A pathway is defined to be an APS/RPS sequence, for instance, $x(1) \longrightarrow x(2) \dashv\!\!\!\rightarrow x(3)$. In this pathway, there are two pathway segments, one APS $x(1) \longrightarrow x(2)$ and one RPS $x(2) \dashv\!\!\!\rightarrow x(3)$. A set of pathways used as the prior knowledge is denoted by $\mathcal{G}$. We define $\mathcal{G}_{\mathcal{A}}$ and $\mathcal{G}_{\mathcal{R}}$ to include all the APS and RPS segments in $\mathcal{G}$, respectively. We refer to regulations of the form $x(i) \longrightarrow x(j)$ and $x(i) \dashv\!\!\!\rightarrow x(j)$ as "pairwise regulations." We denote the set of genes involved in $\mathcal{G}$ by G and, without loss of generality, we fix an order to the genes in G and denote this vector of genes by $\mathbf{g}$.

In addition to pairwise regulations, one can consider a subset of pathways. The *regulatory set* for gene x is the set of genes affected by x, i.e., regulated by x through some APS/RPS. We denote this set by $R_{x(i)}$ for gene $x(i)$. We denote the union of a gene, $x(i)$, with its regulatory set, $R_{x(i)}$, by $\bar{R}_{x(i)}$. As an example, for the pathways

shown in Figure 3.2,

$$R_{x(1)} = \{x(3), x(4)\}, R_{x(2)} = \{x(4), x(5)\}, R_{x(3)} = \{x(1)\},$$
$$R_{x(4)} = \{x(5)\}, R_{x(5)} = \{x(6)\}, R_{x(6)} = \emptyset.$$

$\mathbf{r}_{x(i)}$ and $\bar{\mathbf{r}}_{x(i)}$ denote the vectors of genes in $R_{x(i)}$ and $\bar{R}_{x(i)}$ given the order induced from vector $\mathbf{g}$.

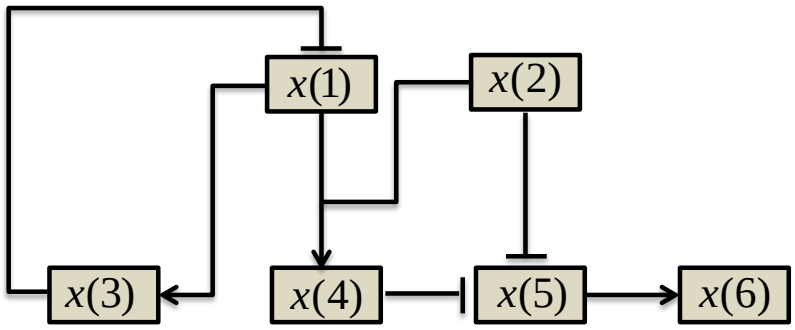


Figure 3.1: An example of pathways with "feedback" containing 6 genes. This contains 3 RPS's and 4 APS's.

Pathway information is not regulatory (in a functional sense) and is understood to be marginal and incomplete [75]. Moreover, these pathways provide no "testable piece of information" [55]. Nevertheless, we can introduce a way of quantifying them objectively. For the moment, assume that these pathways convey "complete information," that is, they are not affected by unspecified crosstalk or conflicting interaction therein. Under this assumption and recognizing the way in which the pathways are built from different experimental settings (conditions) in different cell lines, in a manner analogous to [75], we quantify the pairwise regulations in a conditional

probabilistic manner:

$$\begin{aligned} &\text{APS: } \Pr(x(j) = \text{UR} | x(i) = \text{UR}) \geq 1 - \varepsilon_{ij}; \text{for some small } \varepsilon_{ij} > 0 \\ &\text{RPS: } \Pr(x(j) = \text{DR} | x(i) = \text{UR}) \geq 1 - \varepsilon_{ij}; \text{for some small } \varepsilon_{ij} > 0. \end{aligned} \tag{3.1}$$

For Gaussian joint distributions, we change the inequalities to simpler ones involving correlation:

$$\begin{aligned} &\text{APS: } \rho_{x(i),x(j)} \geq 1 - \varepsilon_{ij}; \text{for some small } \varepsilon_{ij} > 0 \\ &\text{RPS: } \rho_{x(i),x(j)} \leq -1 + \varepsilon_{ij}; \text{for some small } \varepsilon_{ij} > 0, \end{aligned} \tag{3.2}$$

where the notation $\rho_{x(i),x(j)}$ is used to denote the correlation coefficient between two entities $x(i)$ and $x(j)$. The definitions in equation (3.1) are directional and asymmetric, so that the flow of influence is preserved; on the other hand, the definitions in equation (3.2) are symmetric but not directional. Moreover, the interpretation of equation (3.1) as correlations in equation (3.2) is not always be appropriate. Specifically, in case of a cycle (directed loop regardless of type of regulation), this two-way interpretation is inapplicable. As an example, see Figure 3.2, where there is an APS from $x(1)$ to $x(3)$ while an RPS from $x(3)$ to $x(1)$. Hence, when using equation (3.2) for the Gaussian case, we only apply it for acyclic pathways.

We also employ the conditional Shannon entropy of a gene given its regulatory set via the constraint

$$\text{H}_{\boldsymbol{\theta}}(x(i) | R_{x(i)}) \leq \xi_i; \forall x(i) \in \mathcal{G}, \text{for some small } \xi_i > 0, \tag{3.3}$$

where $\text{H}_{\boldsymbol{\theta}}(v_1|v_2)$ is the conditional Shannon entropy, obtained by a $\boldsymbol{\theta}$-parameterized distribution and computed with respect to the uniform measure. $\text{H}_{\boldsymbol{\theta}}(x(i)|R_{x(i)})$ is the amount of information needed to describe the outcome of $x(i)$ given $R_{x(i)}$. Note that the regulatory set information does not take regulation type (activation, inhibition)

into account. Hence, we consider them as two separate pieces of information.

The assumption of having complete pathways is unrealistic and there are many sources of uncertainty impeding us from constructing a single distribution on the features. Nonetheless, the available information can be utilized to impose a probability measure (prior probability) on an uncertainty class of distributions – that is, a prior distribution, $\pi(\boldsymbol{\theta})$, over the $\boldsymbol{\theta}$-parameterized feature-distribution. We extend the quantification in (3.1) and (3.2) to this prior probability by

$$\begin{aligned} &\text{APS: } \mathrm{E}_{\boldsymbol{\theta}}[\Pr(x(j) = \mathrm{UR} | x(i) = \mathrm{UR})] \geq 1 - \varepsilon_{ij}; \text{for some small } \varepsilon_{ij} > 0 \\ &\text{RPS: } \mathrm{E}_{\boldsymbol{\theta}}[\Pr(x(j) = \mathrm{DR} | x(i) = \mathrm{UR})] \geq 1 - \varepsilon_{ij}; \text{for some small } \varepsilon_{ij} > 0. \end{aligned} \tag{3.4}$$

Table 3.1: Continuous representation: regulations in a segment view of signaling pathways when instead of ON and OFF, we respectively insert UR (up-regulated) and DR (down-regulated).

Pathway segment	Interaction type	A sample logic	Bayesian information quantification
$x_i \longrightarrow x_j$	APS	$x(j) = x(i)$	$E_{\boldsymbol{\theta}}[\Pr(x(j) = \mathrm{UR} \| x(i) = \mathrm{UR})] \geq 1 - \xi^a_{i,j}$; for some small $\xi^a_{i,j} > 0$
$x_i \longrightarrow\!\!\!\!\!\!\mid\ x_j$	RPS	$x(j) = \bar{x}(i)$	$E_{\boldsymbol{\theta}}[\Pr(x(j) = \mathrm{DR} \| x(i) = \mathrm{UR})] \geq 1 - \xi^r_{i,j}$; for some small $\xi^r_{i,j} > 0$

Upon relaxation to the correlation coefficients, we have

$$\begin{aligned} &\text{APS: } \mathrm{E}_{\boldsymbol{\theta}}[\rho_{x(i),x(j)}] \geq 1 - \varepsilon_{ij}; \text{for some small } \varepsilon_{ij} > 0 \\ &\text{RPS: } \mathrm{E}_{\boldsymbol{\theta}}[\rho_{x(i),x(j)}] \leq -1 + \varepsilon_{ij}; \text{for some small } \varepsilon_{ij} > 0. \end{aligned} \tag{3.5}$$

Furthermore, for the conditional entropy,

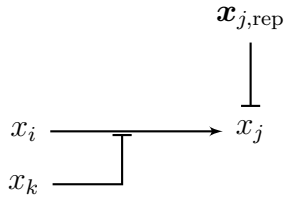
$$\mathrm{E}_{\boldsymbol{\theta}}[\mathrm{H}_{\boldsymbol{\theta}}(x(i)|R_{x(i)})] \leq \xi_i; \forall x(i) \in \mathcal{C}, \text{for some small } \xi_i > 0, \tag{3.6}$$

where $\mathcal{C}$ contains the constraints in the form of regulatory sets.

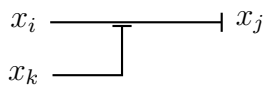
3.2 Discrete Model

Defined in Section 3.1, an "activating pathway segment" (APS), denoted by $x_i \longrightarrow x_j$, means that if species x_i is ON then species x_j becomes ON. Similarly, a "repressing pathway segment" (RPS), denoted by $x_i \dashv\!\!\!\rightarrow x_j$, means that if species x_i is ON then species x_j becomes OFF. Suggested in [76], here, we modify our interpretation about the APS and update it as follows: if x_i is ON and *all the repressing elements connected directly to x_j, denoted by $\boldsymbol{x}_{j,\mathrm{rep}}$ are OFF, then x_j is ON.*

Furthermore, in this thesis, we also define the term "conditional APS" (CAPS) denoted by



to mean that provided that species x_k is OFF, then the APS $x_i \longrightarrow x_j$ is active. Similarly, we define the term "conditional RPS" (CRPS) denoted by



to mean that provided that species x_k is OFF, then the RPS $x_i \dashv\!\!\!\rightarrow x_j$ is active. The set of all the triple (i, j, k) in the form of a CAPS and CRPS is denoted by $\mathcal{G}_{ca}$ and $\mathcal{G}_{cr}$, respectively. These regulations are summarized in Table 3.2 where we give

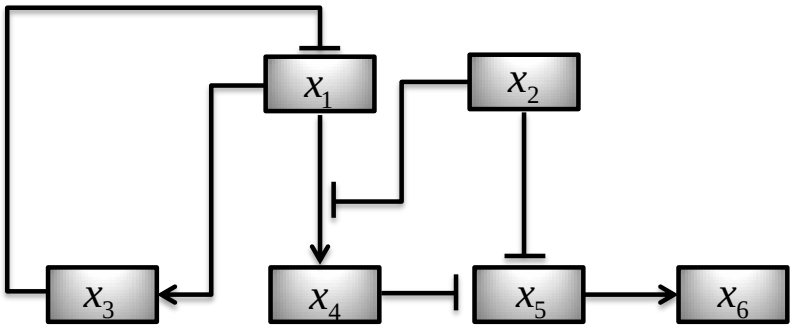


Figure 3.2: An example of pathways with "feedback" containing 6 genes. This contains 3 RPS's and 4 APS's.

an example for a very basic logic between interacting species using the basis, AND, OR, and NOT, denoted by $\Omega = \{\wedge, \vee, -\}$, respectively. Then, a pathway is defined to be a sequence of APS/RPS/CAPS/CRPS, e.g., $x_1 \longrightarrow x_2 \dashv x_3$. In this pathway, there are two pathway segments, one APS $x_1 \longrightarrow x_2$ and one RPS $x_2 \dashv x_3$. A set of pathways used as the prior knowledge throughout is denoted by $\mathcal{G}$. We define $\mathcal{G}_a$ and $\mathcal{G}_r$ to include all the segments in $\mathcal{G}$ in form of APS and RPS, respectively. We call the regulations in above the "segment-wise regulations."

Similar to the continuous modeling, we use the term "regulatory set for gene x" as the set of genes affected by x, i.e., regulated by x through some APS/RPS. We denote this set by R_{x_i} for gene x_i. We denote the union of a gene, x_i, with its regulatory set, R_{x_i} with $\overline{R}_{x_i}$. As an example, consider the pathways shown in Figure 3.2.

According to the definitions above, one may write as in the bottom of the page.

$$R_{x_1} = \{x_3, x_4\}, R_{x_2} = \{x_4, x_5\}, R_{x_3} = \{x_1\},$$
$$R_{x_4} = \{x_5\}, R_{x_5} = \{x_6\}, R_{x_6} = \emptyset.$$

Pathway information s not a regulatory (in a functional sense) and is understood to be marginal and incomplete [41]. Hence its information is quantified in a conditional probabilistic manner in [41]. In this part, we extend it to the two newly defined pathway segments as follows:

$$\text{CAPS: } \Pr(x_j = 1 | x_i = 1, x_k = 0, \boldsymbol{x}_{\text{rep},j} = \mathbf{0}) \geq 1 - \xi^{ca}_{ijk}; \text{for some small } \xi^{ca}_{ijk} > 0 \tag{3.7}$$

$$\text{CRPS: } \Pr(x_j = 0 | x_i = 1, x_k = 0) \geq 1 - \xi^{cr}_{ijk}; \text{for some small } \xi^{cr}_{ijk} > 0. \tag{3.8}$$

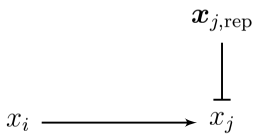
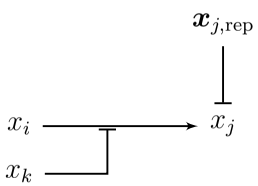
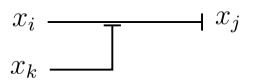
The assumption of having complete pathways is unrealistic and there are many sources of uncertainty impeding us from constructing a single distribution on features. Nonetheless, the available information can be utilized to impose a probability measure (prior probability) on an uncertainty class of distributions- that is a prior distribution o, $\pi(\boldsymbol{\theta})$, over the $\boldsymbol{\theta}-$parameterized feature-distribution. Hence, as proposed in [41], we summarize the Bayesian quantification of the information contained in the pathways in Table 3.2. Furthermore, for the conditional entropy,

$$H_{\boldsymbol{\theta}}[x_i | R_{x_i}] \leq \xi^{reg}_i; \forall x_i \in C, \text{for some small } \xi^{reg}_i > 0. \tag{3.9}$$

where $H_{\boldsymbol{\theta}}[.|.]$ is the conditional Shannon's entropy, obtained by a $\boldsymbol{\theta}-$parameterized distribution, computed with respect to the uniform measure. In (3.9), C denotes all the genes whose regulatory set R_x is nonempty.

One should notice that, prior information may not be limited to what we list in Table 3.2, meaning that as more experiments are done, other types of information would be available.

Table 3.2: Discrete representation: possible regulations in a segment view of signaling pathways when instead of ON and OFF, respectively we have 0 and 1, i.e. the binary representation.

Pathway segment	Interaction type	A sample logic	Bayesian information quantification
	APS	$x_j = x_i$	$E_{\boldsymbol{\theta}}[\Pr(x_j = 1 \vert x_i = 1, \boldsymbol{x}_{\text{rep},j} = \mathbf{0})] \geq 1 - \xi^a_{i,j}$; for some small $\xi^a_{i,j} > 0$
	RPS	$x_j = \bar{x}_i$	$E_{\boldsymbol{\theta}}[\Pr(x_j = 0 \vert x_i = 1)] \geq 1 - \xi^r_{i,j}$; for some small $\xi^r_{i,j} > 0$
	CAPS	$x_j = \bar{x}_k \wedge x_i$	$E_{\boldsymbol{\theta}}[\Pr(x_j = 1 \vert x_i = 1, x_k = 0, \boldsymbol{x}_{\text{rep},j} = \mathbf{0})] \geq 1 - \xi^{ca}_{i,j,k}$; for some small $\xi^{ca}_{i,j,k} > 0$
	CRPS	$x_j = \bar{x}_k \wedge \bar{x}_i$	$E_{\boldsymbol{\theta}}[\Pr(x_j = 0 \vert x_i = 1, x_k = 0)] \geq 1 - \xi^{cr}_{i,j,k}$; for some small $\xi^{cr}_{i,j,k} > 0$

4. NORMAL-WISHART PRIOR CONSTRUCTION ON MULTIVARIATE GAUSSIAN*

If knowledge concerning the feature-label distribution is available, then it can be used in classifier design. For instance, in [77], prior information in the form of a finite uncertainty class of feature-label distributions is incorporated to design a discrete steady-state classifier. One can employ minimum-mean-square-error (MMSE) error estimation based on a prior distribution over an uncertainty class of feature-label distributions [78, 79]. An optimal Bayesian classifier (OBC) is introduced in [39, 40] by minimizing the corresponding MMSE error estimator. Together, the MMSE error estimator and the Bayesian classifier significantly improve the two-fold goal of pattern classification, classifier design and error estimation.

The application we have in mind is phenotype classification based on gene (or protein) expression measurements. Rather than depend only on expression data, one can use genetic pathway information to provide prior knowledge and augment classifier design. The procedure involves the following chain:

$$\{\text{pathways}\} \longrightarrow \{\text{prior probability}\} \longrightarrow \{\text{optimal Bayesian classifier}\}.$$

Prior knowledge in the form of a set of pathways is employed to constrain the space of all the measures on the feature-label distribution in accordance with the assumption that the constructed prior probability should be consistent with the pathway information. For instance a simplified illustration of the pathways that are highly

*Parts of this section are reprinted with permission from "Incorporation of Biological Pathway Knowledge in the Construction of Priors for Optimal Bayesian Classification" by M. Shahrokh Esfahani and E. R. Dougherty, 2014, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, in press, © 2014 IEEE.

influential in colon cancer is shown in Figure 4.1.

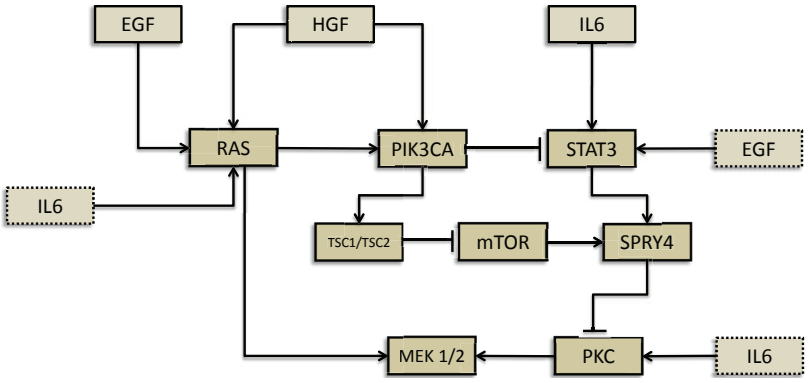


Figure 4.1: A simplified wiring diagram showing the key components of the colon cancer pathways used in [3] and in Section 4.5. Dashed boxes are used to simplify the representation indicating identical components of their counterparts in the solid boxes.

Given the prior distribution governing the uncertainty class of feature-label distributions, a classifier can be constructed that performs optimally relative to the prior distribution and new data [39].

Here we aim to construct a prior distribution on an uncertainty class of covariance matrices utilizing a framework consisting of three steps: (1) Pathway information quantification: information in the biological pathways is quantified via an information-theoretic perspective and translated into a set of "testable" quantities [55]. (2) Data split: data are split into two sets; one for prior construction and the rest for prior update (constructing the posterior). (3) Optimization: combining the portion of data for prior construction with prior knowledge, we build an objective function that is shown to be convex for a Normal-Wishart prior on an unknown mean and precision matrix. In this objective function, the expected mean log-likelihood is

regularized by the quantified information in step 1.

Since some sample data are incorporated to help build a more accurate and informative prior, it is not a pure pre-existing prior, i.e., before any experimental observations [80]. In this regard, one may call the constructed prior a "sample-based prior" or a "reference posterior" [44]. This distribution will be utilized in future analyses as a prior probability which can be updated by observing new data points.

Because the current section makes use of incomplete regulatory knowledge to form a prior distribution to be utilized in conjunction with data to form a classifier that is optimal relative to the prior distribution and the data, it fits into the general paradigm of using incomplete regulatory knowledge to infer networks and feature-label distributions. For instance, in [1], a procedure, which resolves pathway inconsistencies by relaxing pathway timings, is proposed to infer deterministic dynamical models from Boolean pathway knowledge. In the case of incomplete knowledge the procedure outputs an uncertainty class of deterministic models. In [74], inconsistencies and incompleteness are incorporated into a single stochastic dynamical model that can cope with underlying pathway inconsistencies stemming from timing overlaps, different cellular contexts, and incomplete knowledge regarding pathways. In Section 2, the RML classification rule utilizes uncertainty classes constructed via an intermediate step in which pathway information is transformed to a finite number of dynamical systems, each possessing a steady-state distribution. Two class-conditional distributions are estimated using a regularization between the likelihood function and a distance to the uncertainty classes. The RML classifier is then built from these. A basic problem in all uncertainty-based methods is to quantify the uncertainty in the knowledge relative to achieving the objective, which in our case would be classification. In [81], an *objective cost of uncertainty* is proposed that provides such a measure based on the performance difference between the actual

optimal operator (which one would know without the need for data if there were no uncertainty) and the optimal operator relative to the prior distribution and the data.

This section is organized as follows. Section 4.1 is devoted to a review of the optimal Bayesian classifier and then a methodology for quantifying the information in biological pathways. The proposed sample-based prior constructing framework is introduced in Section 4.2. In Section 4.3, the optimization framework is developed for the Gaussian distribution with unknown mean and precision matrix governed by a Normal-Wishart prior. Simulation results on the synthetically generated pathways are provided in Section 4.4. We test the proposed framework on real pathways containing genes associated with colon cancer in Section 4.5. Finally, Section 4.6 contains concluding remarks.

We summarize some notation used in this section. Boldface lower case letters denote column vectors. Concatenation of several vectors is also denoted by a boldface lower case letter. The k-th element of the vector $\boldsymbol{\pi}$ is denoted by $\pi(k)$. Boldface upper case letters are used to denote matrices. $\mathrm{tr}(.)$, $(.)^T$, and $|.|$ denote the trace, transpose, and determinant operators, respectively; however, when the argument is not a matrix, the notation $|.|$ stands for the cardinality of a set. For a matrix $\mathbf{W}$, if A and B consist of a set of rows and a set of columns in $\mathbf{W}$, respectively, then the sub-matrix corresponding to the rows in A and columns in B is denoted by $\mathbf{W}_{A,B}$. If $A = B$, then we simply write $\mathbf{W}_A$. $\Pr(E)$ denotes the probability of event E. $\mathrm{E}_x[g(x)]$ denotes taking the expectation of $g(x)$ with respect to x. Finally, $\log(.)$ denotes the natural logarithm. We use the terms: feature, variable, and entity interchangeably.

4.1 Background

4.1.1 Optimal Bayesian Classifier

Given a binary classification problem with classes $y \in \{0, 1\}$, we observe a collection of n sample points, S_n, in a sample space $\mathcal{X}$, with n_y i.i.d. points from each class. Call c the *a priori* probability that an individual sample point $\mathbf{x} \in \mathbb{R}^p$ is from class 0 and let the class-conditional distribution for class y, denoted $f_{\boldsymbol{\theta}_y}(\mathbf{x}|y)$, be parameterized by $\boldsymbol{\theta}_y$. The feature-label distribution is completely specified by the modeling parameters $\boldsymbol{\theta} = [c, \boldsymbol{\theta}_0, \boldsymbol{\theta}_1]$. In [78], it is assumed that c, $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_1$ are all independent prior to observing the data. Denoting the prior for $\boldsymbol{\theta}_y$ by $\pi(\boldsymbol{\theta}_y)$, we have $\pi(\boldsymbol{\theta}) = \pi(c)\pi(\boldsymbol{\theta}_0)\pi(\boldsymbol{\theta}_1)$. The posterior preserves independence. Denoting it by $\pi^*(\boldsymbol{\theta}_y)$ and letting $\mathbf{x}_i^y$ be the i-th sample point in class y [39],

$$\pi^*(\boldsymbol{\theta}_y) \propto \pi(\boldsymbol{\theta}_y) \prod_{i=1}^{n_y} f_{\boldsymbol{\theta}_y}(\mathbf{x}_i^y|y). \tag{4.1}$$

Priors quantify the known information about the distribution before observing data. We have the option of using diffuse (non-informative) priors, as long as the posterior (conditioned on the sample) is a valid density function. Alternatively, informative priors can supplement the classification problem with additional information. In the Bayesian framework, we characterize the initial uncertainty in the "actual distribution" through the prior. As data are observed, this uncertainty should converge to a certainty on the true distribution. The Bayesian framework for the problem of pattern classification has been widely studied in the Bayesian network view [82–85]. The Bayesian framework for the pattern classification in which two prior distributions on the feature-label distributions are assumed was developed more recently with the introduction of the Bayesian MMSE error estimator [78]. The optimal Bayesian

classifier (OBC) is obtained by minimizing this error estimator. This classifier is given by [39]

$$\psi_{\text{OBC}} = \begin{cases} 0, \text{ if } \mathrm{E}_{\pi^*}[c] f(\mathbf{x}|0) \geq (1 - \mathrm{E}_{\pi^*}[c]) f(\mathbf{x}|1) \\ 1, \text{ otherwise} \end{cases}, \tag{4.2}$$

where $f(\mathbf{x}|y), y \in \{0, 1\}$, called the "effective class-conditional densities" (ECCD), are defined by

$$f(\mathbf{x}|y) = \int f_{\boldsymbol{\theta}_y}(\mathbf{x}|y) \pi^*(\boldsymbol{\theta}_y) d\boldsymbol{\theta}_y.$$

Henceforth, we drop the sub (sup)-script denoting the dependency on the label, y, but one should recognize that the prior knowledge is assumed to be available for both classes, separately.

4.2 Regularized Expected Mean Log-Likelihood Prior

Using prior knowledge in the form of signaling pathways, we propose a regularized expected mean-log-likelihood (REML) framework in which the expectation is taken to marginalize the dependency of the mean-log-likelihood to the actual feature-label distribution parameters (e.g. mean and covariance matrix in a Gaussian setting). The regularization is performed to apply prior information as soft constraints. The final objective function is a function of the hyperparameters of interest to determine the prior distribution.

To this end, we first split the given sample, S_n, into two parts for each class $y \in \{0, 1\}$: $S_{n_y^p}^{prior,y}$ and $S_{n_y^t}^{train,y}$, with $n_y = n_y^p + n_y^t$ and $n = n_0 + n_1$. Assume that the sample set (consisting of $n_p = n_0^p + n_1^p$ sample points) used for prior construction is denoted by $S_{n_p}^{prior}$. Moreover, assume that $\mathcal{C}$ contains the constraints in the form of regulatory sets. Henceforth, for notational ease, we drop the index y.

We state the proposed optimization framework with multiple constraints in $\mathcal{C}$:

$$\begin{aligned}\pi_{\text{REML}}(\boldsymbol{\theta}) := \quad & \arg\min \quad -(1-\lambda_1-\lambda_2)\mathrm{E}_{\boldsymbol{\theta}}\Big[\ell_{n_p}(\boldsymbol{\theta})\Big] \\ & \pi(\boldsymbol{\theta}) \in \Pi, \xi_i \geq 0 \\ & \varepsilon_{i_a j_a} \geq 0, \varepsilon_{i_r j_r} \geq 0 \\ & \qquad +\lambda_1 \sum_{i=1}^{|\mathcal{C}|} \xi_i + \lambda_2 \Big[\sum_{(i_a,j_a)\in\mathcal{G}_{\mathcal{A}}} \varepsilon_{i_a j_a} + \sum_{(i_r,j_r)\in\mathcal{G}_{\mathcal{R}}} \varepsilon_{i_r j_r} \Big]\end{aligned} \tag{4.3}$$

subject to the following constraints:

$$\mathrm{E}_{\boldsymbol{\theta}}\Big[\mathrm{H}_{\boldsymbol{\theta}}(x(i)|R_{x(i)})\Big] \leq \xi_i, x(i) \in \mathcal{G} \tag{4.4}$$

$$\mathrm{E}_{\boldsymbol{\theta}}\Big[\Pr(x(j_a) = \mathrm{UR}|x(i_a) = \mathrm{UR})\Big] \geq 1 - \varepsilon_{i_a j_a}, (i_a, j_a) \in \mathcal{G}_{\mathcal{A}} \tag{4.5}$$

$$\mathrm{E}_{\boldsymbol{\theta}}\Big[\Pr(x(j_r) = \mathrm{DR}|x(i_r) = \mathrm{UR})\Big] \geq 1 - \varepsilon_{i_r j_r}, (i_r, j_r) \in \mathcal{G}_{\mathcal{R}} \tag{4.6}$$

where Π is the feasible region to which the prior distribution belongs and $\ell_{n_p}(\boldsymbol{\theta}) := \frac{1}{n_p}\ell(\boldsymbol{\theta}; S_{n_p}^{prior})$, in which $\ell(\boldsymbol{\theta}; S_{n_p}^{prior})$ is the log-likelihood function. $\ell_{n_p}(\boldsymbol{\theta})$ can be interpreted as an estimator of [86–88]

$$\int_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}|\boldsymbol{\theta}_{true}) \log f(\mathbf{x}|\boldsymbol{\theta}) d\mathbf{x},$$

which is a measure of "similarity" between the true model, governed by $\boldsymbol{\theta}_{true}$, and the one governed by the parameter $\boldsymbol{\theta}$. This estimate is also employed in the Akaike's information criterion for model selection [89]. In (4.3), the parameters λ_1 and λ_2, for which we have $\lambda_1, \lambda_2 \geq 0$ and $\lambda_1 + \lambda_2 \leq 1$, are the regularization parameters (or the design parameter) depending on the relative importance of the prior sources and likelihood. The Shannon entropy, $\mathrm{H}_{\boldsymbol{\theta}}(.)$, is computed with respect to the uniform measure.

In equation (4.3), the term $\mathrm{E}_{\boldsymbol{\theta}}\Big[\ell_{n_p}(\boldsymbol{\theta})\Big]$ reflects the "expected similarity between the observed data and the true model." Prior averaging performs marginalization with respect to the model parametrization making us depend only on the hyperparameters.

Assuming Gaussian distributions, equations (4.5)-(4.6) become

$$\mathrm{E}_{\boldsymbol{\theta}}\Big[\rho_{x(i_a),x(j_a)}\Big] \geq 1 - \varepsilon_{i_a j_a}, (i_a, j_a) \in \mathcal{G}_{\mathcal{A}} \tag{4.7}$$

$$\mathrm{E}_{\boldsymbol{\theta}}\Big[\rho_{x(i_r),x(j_r)}\Big] \leq -1 + \varepsilon_{i_r j_r}, (i_r, j_r) \in \mathcal{G}_{\mathcal{R}}. \tag{4.8}$$

4.3 Multivariate Gaussian with Normal-Wishart Prior

For the multivariate Gaussian distribution, $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1})$, we have $\boldsymbol{\theta} = [\boldsymbol{\mu}, \boldsymbol{\Lambda}]$. Define the feasible region, Π, for given ν and κ, for the prior probability as $\Pi = \{\mathcal{NW}(\mathbf{m}, \nu, \mathbf{W}, \kappa) : \mathbf{m} \in \mathbb{R}^p, \mathbf{W} > 0\}$, the set of all Normal-Wishart distributions (hence, an inverse Wishart distribution for the covariance matrix). The Normal-Wishart distribution is determined fully by four parameters, $[\mathbf{m}_{p\times 1}, \nu, \mathbf{W}_{p\times p}, \kappa]$, via

$$\begin{aligned} \boldsymbol{\mu}|\boldsymbol{\Lambda} &\sim \mathcal{N}(\mathbf{m}, (\nu\boldsymbol{\Lambda})^{-1}) \\ \boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1} &\sim \mathcal{W}(\mathbf{W}, \kappa) = B(\mathbf{W}, \kappa)|\boldsymbol{\Lambda}|^{(\kappa-p-1)/2} \exp\{-\tfrac{1}{2}\mathrm{tr}(\mathbf{W}^{-1}\boldsymbol{\Lambda})\}, \end{aligned} \tag{4.9}$$

where $B(\mathbf{W}, \kappa) \propto |\mathbf{W}|^{-\kappa/2}$ [90]. In order to have a proper prior, we should have $\mathbf{W} > 0$ and $\kappa > p - 1$. As $\nu \to 0$, the prior probability for the mean vector tends to be more non-informative (flatter).

The general optimization framework proposed in (4.3)-(4.6) does not yield a convex programming for which a guaranteed converging algorithm exists. Therefore, to facilitate convergence to the global optimum, we decompose the full procedure into two optimization problems. The main advantage of this decomposition is tractability

of existing algorithms for solving convex problems. In particular, although the final solution is different from that of the initial problem, the effect of prior knowledge can be assessed by deriving analytical expressions for the gradient and the hessian of the cost functions. First, we assume $\lambda_2 = 0$ by utilizing only the regulatory set constraints: $\lambda_2 = 0 \rightarrow$ solve optimization in equations (4.3)-(4.4). Then, the second optimization problem treats the regulation types according to the constraints simplified to correlations in (4.7)-(4.8). This procedure is outlined in Figure 4.2. The second optimization will be discussed in detail in Section 4.3.2.

4.3.1 Regulatory Set Constraints: $\lambda_2 = 0$

Setting the regularization parameter $\lambda_2 = 0$, the general REML optimization reduces to

$$\min_{\mathbf{m},\mathbf{W}>0,\xi\geq 0} -(1-\lambda_1)\int_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \ell_{n_p}(\boldsymbol{\theta})\pi(\boldsymbol{\theta})d\boldsymbol{\theta} + \lambda_1\xi, \tag{4.10}$$

subject to constraints of the form

$$\int_{\boldsymbol{\Sigma}>0} \mathrm{H}_{\boldsymbol{\theta}}(x|R_x)\pi(\boldsymbol{\theta})d\boldsymbol{\theta} \leq \xi.$$

Removing the additive constant parts, for the log-likelihood of the Gaussian distribution we have

$$2\ell_{n_p}(\boldsymbol{\mu},\boldsymbol{\Lambda}) = \log|\boldsymbol{\Lambda}| - \frac{1}{n_p}\sum_{i=1}^{n_p}\mathrm{tr}[\boldsymbol{\Lambda}(\mathbf{x}_i-\boldsymbol{\mu})(\mathbf{x}_i-\boldsymbol{\mu})^T]. \tag{4.11}$$

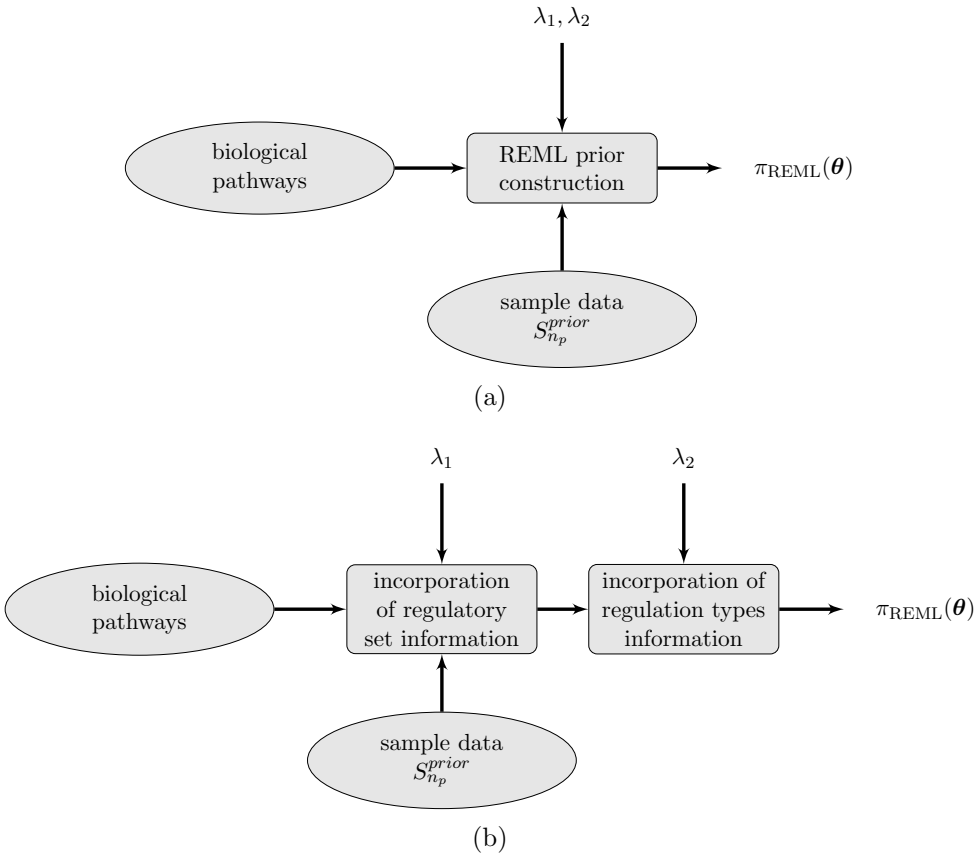


Figure 4.2: Panel (a) is an illustrative view of the general REML approach. Relaxing the framework for the Gaussian scenario, the figure in panel (b) demonstrates a schematic view of the methodology of breaking the original REML optimization problem.

Taking the expectation with respect to the mean and covariance matrix, i.e., $\mathrm{E}_{\boldsymbol{\Lambda}}[\mathrm{E}_{\boldsymbol{\mu}|\boldsymbol{\Lambda}}[.]]$, yields

$$\begin{aligned} 2\mathrm{E}_{\boldsymbol{\theta}}[\ell_{n_p}(\boldsymbol{\mu}, \boldsymbol{\Lambda})] &= \mathrm{E}\log|\boldsymbol{\Lambda}| - \tfrac{\kappa}{n_p}\sum_{i=1}^{n_p}\mathrm{tr}(\mathbf{W}(\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T) + \tfrac{p}{\nu} \\ &\equiv \log|\mathbf{W}| - \kappa\mathrm{tr}(\mathbf{W}\mathbf{V}_{\mathbf{m}}) + \sum_{d=1}^{p}\psi(\tfrac{\kappa+1-d}{2}), \end{aligned} \tag{4.12}$$

where the last equivalency is due to the assumption of predetermined ν (and κ), $\psi(.)$ is the digamma function defined as $\psi(t) = \frac{d}{dt}\log\Gamma(t)$ [90], and

$$\mathbf{V}_{\mathbf{m}} = \frac{1}{n_p}\sum_{i=1}^{n_p}(\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T.$$

Solving the optimization problem in equation (4.10) with respect to $\mathbf{m}$ gives $\hat{\mathbf{m}}_{\text{REML}} = \frac{1}{n_p}\sum_{i=1}^{n_p}\mathbf{x}_i$. If $\lambda_1 = 0$ (no regularization), then the optimization problem in (4.11) implies $\hat{\mathbf{W}} = (\kappa\mathbf{V})^{-1}$, provided that $n_p \geq p+1$, where $\mathbf{V}$ is the matrix $\mathbf{V}_{\mathbf{m}}$ with $\mathbf{m}$ replaced by its estimate $\hat{\mathbf{m}}_{\text{REML}}$ whereby

$$(\boldsymbol{\mu}, \boldsymbol{\Lambda}) \sim \mathcal{N}(\boldsymbol{\mu}|\hat{\mathbf{m}}_{\text{REML}}, (\nu\boldsymbol{\Lambda})^{-1})\mathcal{W}(\boldsymbol{\Lambda}|(\kappa\mathbf{V})^{-1}, \kappa).$$

From this distribution we obtain $\text{E}[\boldsymbol{\Lambda}] = \mathbf{V}^{-1}$ (see [91] for the moments of the Wishart distribution).

We will consider two cases for the covariance matrix, and consequently for $\mathbf{W}$. Throughout, we assume $x \notin R_x$ (no self-regulation) and have $x \sim \mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x)$, $\mathbf{r}_x \sim \mathcal{N}(\boldsymbol{\mu}_{R_x}, \boldsymbol{\Sigma}_{R_x})$, and $\bar{\mathbf{r}}_x \sim \mathcal{N}(\boldsymbol{\mu}_{\bar{\boldsymbol{R}}_{\boldsymbol{x}}}, \boldsymbol{\Sigma}_{\bar{R}_x})$. $\boldsymbol{\Lambda}$ denotes the precision matrix. We use $\boldsymbol{\Sigma}_x$ to denote the variance of the single variable x.

4.3.1.1 Covariance Matrix Containing Only $\bar{R}_{x(i)}$

Suppose the precision matrix contains only those entities contributed in the constraint: the constrained entity and the elements of its regulatory set. Omitting the gene index and simply denoting a gene by x, if we write the precision matrix $\boldsymbol{\Lambda}_{\bar{R}_x} = \boldsymbol{\Sigma}^{-1}$ in blocks as

$$\boldsymbol{\Lambda}_{\bar{R}_x} = \begin{bmatrix} \boldsymbol{\Lambda}_{R_x} & \boldsymbol{\Lambda}_{12} \\ \boldsymbol{\Lambda}_{21} & \boldsymbol{\Lambda}_x \end{bmatrix},$$

knowing that $\mathbf{\Lambda}_{\bar{R}_x} \sim \mathcal{W}(\mathbf{W}_{\bar{R}_x}, \kappa)$, we have $\mathbf{\Lambda}_{R_x} \sim \mathcal{W}(\mathbf{W}_{R_x}, \kappa)$, where

$$\mathbf{W}_{\bar{R}_x} = \begin{bmatrix} \mathbf{W}_{R_x} & \mathbf{W}_{12} \\ \mathbf{W}_{21} & \mathbf{W}_x \end{bmatrix}.$$

Before restating the optimization problem for this special case, we first find the constraint reflected from the Wishart prior distribution. Using existing formulas for the mutual information and the entropy of Gaussian distributions, we obtain (see Appendix C.1.1 on the companion website)

$$\mathrm{H}_{\mathbf{\Lambda}}(x|R_x) \propto \log 2\pi e - \log|\mathbf{\Lambda}_{\bar{R}_x}| - \log|(\mathbf{\Lambda}^{-1}_{\bar{R}_x})_{R_x}| = \log 2\pi e - \log|\mathbf{\Lambda}_x|. \tag{4.13}$$

From the properties of the Wishart distribution, $\mathbf{\Lambda}_x \sim \mathcal{W}(\mathbf{W}_x, \kappa)$. Hence, the constraint can be written as

$$\mathrm{E}_{\mathbf{\Lambda}}[\mathrm{H}_{\mathbf{\Lambda}}(x|R_x)] \propto \log \pi e - \log|\mathbf{W}_x| - \psi(\frac{\kappa}{2}). \tag{4.14}$$

Plugging the preceding results into the optimization framework yields

$$\mathrm{CP}_1(\kappa): \begin{array}{ll} \min\limits_{\mathbf{W}_{\bar{R}_x}>0, \xi \geq 0} & -\frac{1}{2}(1-\lambda_1)\Big[\log|\mathbf{W}_{\bar{R}_x}| - \kappa \mathrm{tr}(\mathbf{W}_{\bar{R}_x}\mathbf{V})\Big] + \lambda_1 \xi \\ \text{Subject to} & -\log|\mathbf{W}_x| - \psi(\frac{\kappa}{2}) \leq \xi; \xi \geq \underline{\xi}, \end{array} \tag{4.15}$$

where $\underline{\xi} = -\log(\pi e)$. From the inequalities in [92], one can see that the parts containing $\log|\mathbf{W}_{\bar{R}_x}|$ are concave, thereby making the optimization problem (i.e, the objective function and constraints) in (4.15) convex in the matrix $\mathbf{W}_{\bar{R}_x}$.

4.3.1.2 Covariance Matrix Containing $\bar{R}_x$ along with other entities in $\mathcal{G}$

In this subsection, we assume that the covariance matrix which needs to be estimated has more genes than that of $\bar{R}_x$. We denote the covariance matrix and its inverse by $\mathbf{\Sigma}$ and $\mathbf{\Lambda}$, respectively, and the parameters of the Wishart distribution governing the precision matrix by $\mathbf{W}$ and κ. The precision matrix and its prior are represented in a block format by

$$\mathbf{\Lambda} = \begin{bmatrix} \mathbf{\Lambda}_{R_x} & \mathbf{\Lambda}_{12} & \mathbf{\Lambda}_{13} \\ \mathbf{\Lambda}_{21} & \mathbf{\Lambda}_{x} & \mathbf{\Lambda}_{23} \\ \mathbf{\Lambda}_{31} & \mathbf{\Lambda}_{32} & \mathbf{\Lambda}_{33} \end{bmatrix}; \mathbf{W} = \begin{bmatrix} \mathbf{W}_{R_x} & \mathbf{W}_{12} & \mathbf{W}_{13} \\ \mathbf{W}_{21} & \mathbf{W}_{x} & \mathbf{W}_{23} \\ \mathbf{W}_{31} & \mathbf{W}_{32} & \mathbf{W}_{33} \end{bmatrix}. \tag{4.16}$$

Then, knowing that $\mathbf{\Lambda}_x - \mathbf{\Lambda}_{23}\mathbf{\Lambda}_{33}^{-1}\mathbf{\Lambda}_{32} \sim \mathcal{W}(\mathbf{W}_x - \mathbf{W}_{23}\mathbf{W}_{33}^{-1}\mathbf{W}_{32}, \kappa - \dim(\mathbf{W}_{33})))$, where dim(.) returns the dimension of a matrix, the optimization problem in (4.10) can now be restated as (for the conditional entropy constraint please refer to Appendix C)

$$\text{CP}_2(\kappa): \begin{array}{ll} \min\limits_{\mathbf{W}>0,\xi\geq 0} & -\frac{1}{2}(1-\lambda_1)\Big[\log|\mathbf{W}| - \kappa\text{tr}(\mathbf{W}\mathbf{V})\Big] + \lambda_1\xi \\ \text{Subject to} & -\log|\mathbf{W}_x - \mathbf{W}_{23}\mathbf{W}_{33}^{-1}\mathbf{W}_{32}| - \psi(\frac{\kappa-(p-|R_x|-1)}{2}) \leq \xi; \\ & \xi \geq \underline{\xi}; x(i) \in \mathcal{C} \end{array} \tag{4.17}$$

Lemma 3. *The programming, $CP_2(\kappa)$ is a convex programming.*

Proof. Please refer to Appendix C.2. Q.E.D.

Corollary 2. *The optimization problems $CP_1(\kappa)$ and $CP_2(\kappa)$ satisfy Slater's condition.*

Proof. It can be readily seen from the constraints and considering the relative interior of the feasible region of the problem, by choosing $\mathbf{W}$, a scaled identity matrix. Q.E.D.

The optimization problem in (4.17) can be readily extended to multiple constraints, i.e., the situation where we incorporate all the entities' information simultaneously, by considering the corresponding submatrix for a gene and its regulatory set. This is given, for any $\xi_i \geq \underline{\xi}$, by

$$\begin{aligned} \mathrm{CP}_2(\kappa): \quad & \min_{\mathbf{W}>0, \xi_i \geq 0} && -\tfrac{1}{2}(1-\lambda_1)\Big[\log|\mathbf{W}| - \kappa \mathrm{tr}(\mathbf{W}\mathbf{V})\Big] + \lambda_1 \sum_{i=1} \xi_i \\ & \text{Subject to} && -\log|\overline{\mathbf{W}}_{x(i)}| - \psi(\tfrac{\kappa-(p-|R_{x(i)}|-1)}{2}) \leq \xi_i, \end{aligned} \tag{4.18}$$

where

$$\overline{\mathbf{W}}_{x(i)} := \mathbf{W}_{x(i)} - \mathbf{W}_{x(i),\mathbf{g}\setminus\bar{R}_{x(i)}} \mathbf{W}^{-1}_{\mathbf{g}\setminus\bar{R}_{x(i)}} \mathbf{W}^{T}_{x(i),\mathbf{g}\setminus\bar{R}_{x(i)}}. \tag{4.19}$$

4.3.2 Incorporating Regulation Types

Biological signaling pathways not only contain dependency information between variables, they also illustrate the type of regulation between entities. This can help decrease the uncertainty and modify our estimation of the matrix $\mathbf{W}$. Since, we are assuming that the underlying feature distribution is a joint Gaussian, we incorporate the APS and RPS effects using equations (4.7)-(4.8). Therefore, similar to our interpretation indicated in $\mathrm{CP}_1(\kappa)$ or $\mathrm{CP}_2(\kappa)$, we try to manipulate the expected correlation coefficients. However, instead of taking the expectation of the correlation coefficient, which ends up with a non-convex function, we fix the variances according to what we get from $\mathrm{CP}_2(\kappa)$.

From the properties of the Wishart distribution, $\mathbf{\Sigma} \sim \mathcal{W}^{-1}(\mathbf{\Psi}, \kappa)$, where $\mathbf{\Psi} = \mathbf{W}^{-1}$. Define $\mathbf{\Psi}^* = \mathbf{W}^{*-1}$, where $\mathbf{W}^*$ is the optimal solution of $\mathrm{CP}_2(\kappa)$. The first moments of the elements of the covariance matrix distributed according to an inverse Wishart distribution, i.e., $\mathbf{\Sigma} = [\sigma_{ij}]_{p\times p} \sim \mathcal{W}^{-1}(\mathbf{\Psi}, \kappa)$, are $\mathrm{E}[\sigma_{ij}] = \frac{1}{k-p-1}\psi_{ij}$,

$i, j \in \{1, ..., p\}$ [91], from which we approximately write

$$\mathrm{E}[\rho_{ij} = \rho_{x(i),x(j)}] = \mathrm{E}\left[\frac{\sigma_{ij}}{\sqrt{\sigma_{ii}\sigma_{jj}}}\right] \approx \frac{\mathrm{E}[\sigma_{ij}]}{\frac{1}{k-p-1}\sqrt{\psi^*_{ii}\psi^*_{jj}}} = \frac{\psi_{ij}}{\sqrt{\psi^*_{ii}\psi^*_{jj}}}.$$

The goal of the second optimization paradigm is twofold: while satisfying the correlation-coefficient constraints according to the regulation types, we wish to be as close to the $\mathrm{CP}_2(\kappa)$ solution as possible. Thus, we introduce a penalty term based on the distance from the solution of $\mathrm{CP}_2(\kappa)$ and aim to find the closest, in the sense of the Frobenius norm, positive definite matrix to the matrix $\mathbf{\Psi}^*$. Hence, we introduce the following optimization problem, with optimization parameter $\mathbf{\Psi} = [\psi_{i,j}]_{p\times p}$:

$$\mathrm{CP}_3 : \min_{\mathbf{\Psi}>0, \varepsilon_{ij}\geq 0} \ (1-\lambda_2)||\mathbf{\Psi} - \mathbf{\Psi}^*||_F^2 + \lambda_2\Big[\sum_{(i_a,j_a)\in\mathcal{G}_\mathcal{A}} \varepsilon_{i_a j_a} + \sum_{(i_r,j_r)\in\mathcal{G}_\mathcal{R}} \varepsilon_{i_r j_r}\Big], \quad (4.20)$$

subject to the constraints

$$\begin{cases} 1-\varepsilon_{i_a j_a} \leq \frac{\psi_{i_a j_a}}{\sqrt{{\psi^*}_{i_a i_a}{\psi^*}_{j_a j_a}}} \leq 1; (i_a, j_a) \in \mathcal{G}_\mathcal{A} \\ 1-\varepsilon_{i_r j_r} \leq \frac{-\psi_{i_r j_r}}{\sqrt{{\psi^*}_{i_r i_r}{\psi^*}_{j_r j_r}}} \leq 1; (i_r, j_r) \in \mathcal{G}_\mathcal{R} \\ \psi_{ij} = \psi_{ji}, \ \forall i, j \in \{1, \ldots, p\} \end{cases} \quad (4.21)$$

The parameter $\lambda_2 \in (0, 1)$ is again the regularization factor making the balance between two functions. It can be readily shown that the optimization problem in equations (4.20)-(4.21) is convex.

In sum, we break the general REML problem in equation (4.3)-(4.6) into two sequential problems: (1) the optimization in equations (4.18)-(4.19) ($\mathrm{CP}_2(\kappa)$), and then (2) the optimization in equations (4.20)-(4.21) (CP_3).

4.3.3 Algorithm for Solving $CP_2(\kappa)$

For the sake of simplicity in the formula below, we only consider the one-constraint problem. The multiple constraint problem can be treated similarly. Being a nonlinear inequality constrained programming, we choose the log-barrier interior point method for solving the optimization problem $\mathrm{CP}_2(\kappa)$. The basic idea of the log-barrier interior point method is to replace an inequality constrained nonlinear optimization with a sequential equality constrained problems whose total number of iterations depends on the barrier parameter, some tolerance parameter, number of constraints, and the convergence criterion for the centering problem solved via Newton's method [93].

From Corollary 2, a local optimum for $\mathrm{CP}_2(\kappa)$, which also satisfies the KKT (Karush-Kuhn-Tucker) conditions, corresponds to the global optimum of the optimization problem [93] (refer to Section 5.5 in [93] for more details). Hence, the solution to the KKT system of equations will provide the optimal solution to the problem of interest. Our proposed strategy is a mixture of existing strategies to solve nonlinear convex and log-determinant problems [94, 95]. The core of the algorithm is the log-barrier type of interior point method. Writing the first-order conditions for optimality (KKT system of equations), we approximate these equations by their quadratic approximation [93]. We change some notation to use existing results for log-determinant problems. We write (owing to the symmetry property)

$$\mathbf{W} = \begin{bmatrix} w_1 & w_2 & w_3 & . & . & w_p \\ . & w_{p+1} & w_{p+2} & . & . & w_{2p-1} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & w_{p(p+1)/2} \end{bmatrix}.$$

This matrix can be written as $\sum_{i=1}^{p(p+1)/2} w_i \mathbf{E}_i$, where depending on the column (in the above representation) at which the variable w_i is located (e.g. j), the matrix $\mathbf{E}_i$ is either $\mathbf{E}_i = \mathbf{e}_i\mathbf{e}_j^T + \mathbf{e}_j\mathbf{e}_i^T, j > i$, or $\mathbf{E}_i = \mathbf{e}_i\mathbf{e}_i^T$, where the vector $\mathbf{e}_i$ is the column-vector with 1 in its i-th position. Then, instead of the matrix $\mathbf{W}$, which in general is the parameter needing to be optimized, we need only find $p(p+1)/2$ positions (due to symmetry), these being denoted by $\mathbf{w} = [w_1, w_2, ..., w_{p(p+1)/2}]^T$.

Hence, denoting the optimization parameter by $\mathbf{z} = [\mathbf{w}, \xi]$, we may write the objective function as follows, where to avoid confusion with the probability density f and feature vector $\mathbf{x}$, we use g and $\mathbf{z}$ to denote the objective function and its argument, respectively:

$$g(\mathbf{z}) = -\frac{1}{2}(1-\lambda_1)\Big[\log|\mathbf{W}| - \kappa \mathrm{tr}(\mathbf{W}\mathbf{V})\Big] + \lambda_1 \xi. \tag{4.22}$$

Denoting the log-barrier parameter by μ, the optimization problem $\mathrm{CP}_2(\kappa)$ may be replaced by

$$\min_{\mathbf{z},\mathbf{u}} \; g(\mathbf{z}) - \mu \sum_{i=1}^{2} \log u(i),$$

subject to the new constraints

$$\begin{aligned} &\xi - \underline{\xi} - u(1) = 0 \\ &\xi + \log\Big|\begin{bmatrix} \mathbf{W}_x & \mathbf{W}_{23} \\ \mathbf{W}_{32} & \mathbf{W}_{33} \end{bmatrix}\Big| - \log|\mathbf{W}_{33}| - u(2) = 0. \end{aligned}$$

The new optimization problem is convex and therefor the solution to the KKT conditions provides the global optimum. In order to solve the KKT conditions, we employ the Newton method [94]. Define the vector $\mathbf{y}$ containing the Lagrangian

multipliers. Following [94], we form the dual normal matrix

$$\mathbf{N}(\mathbf{z}, \mathbf{y}, \mathbf{u}) = \mathbf{Hess}(\mathbf{z}, \mathbf{y}) + \mathbf{A}^T(\mathbf{z})\mathbf{U}^{-1}\mathbf{Y}\mathbf{A}(\mathbf{z}), \tag{4.23}$$

where $\mathbf{Hess}(\mathbf{z}, \mathbf{y})$ is the Hessian matrix and the matrices $\mathbf{U}$ and $\mathbf{Y}$ are diagonal matrices whose elements correspond to vectors $\mathbf{u}$ and $\mathbf{y}$, respectively. In equation (4.23), the matrix $\mathbf{A}(\mathbf{z})$ is the Jacobian matrix of the constraints (please refer to Appendix C.3 for the Hessian and Jacobian calculus). Hence, the direction is found based on the Newton's method solver for the KKT conditions.

Considering the k-th iteration, once the direction, $\mathbf{\Delta z}_{(k)}$, is determined, a line-search is employed to find an appropriate length of each step. As a standard approach for constrained problems for determining the step length, $\alpha_{(k)}$, we use the "merit function" similar to that of [94].

The line-search used in the k-th step of the procedure ($\mathbf{z}_{k+1} = \mathbf{z}_k + \alpha_k \Delta \mathbf{z}_k$) is described in detail in Algorithm 1.

Algorithm 1 Line Search for $\alpha_{(k)}$ for the centering program

Input: $\alpha_{(k),1}$, $\rho = 0.5$ (Default Value)
Output: $\alpha_{(k)}$
Initialize: $\overline{\alpha} = 0.95(\max\{-\frac{\Delta u_{(k)}}{u_{(k)}}, -\frac{\Delta y_{(k)}}{y_{(k)}}; i = 1, 2\})$ ([94]-Section 2.1)
$\alpha_{max} \leftarrow \overline{\alpha}$, $\overline{\mathbf{W}} \leftarrow \mathbf{W} + \alpha_{max}\Delta\mathbf{W}_{(k)}$
while $\overline{\mathbf{W}} \leq 0$ **do**
  $\alpha_{max} \leftarrow \rho\alpha_{max}$ (similar to [95])
  $\overline{\mathbf{W}} \leftarrow \mathbf{W} + \alpha_{max}\Delta\mathbf{W}_{(k)}$
end while
if $\alpha_{max} \leq \alpha_{(k),1}$ **then**
  $\alpha_{(k),1} \leftarrow 0.6\alpha_{max}$
end if
Form the merit function $\phi_{(k)}(\alpha)$ (similar to [94])
Implement back-tracking Algorithm for $\alpha \in [\alpha_{(k),1}, \alpha_{max}]$
return $\alpha_{(k)}$

The augmented parameter vector $\mathbf{u}$ and the Lagrangian multiplier vector $\mathbf{y}$ must be element-wise non-negative. Hence, as the initialization, we find an upper bound denoted by $\overline{\alpha}$ [94]. To assure the positive definiteness of the matrix $\mathbf{W}$, we decrease this maximum until the resulting matrix at the current iteration becomes positive definite. From [95], if at the previous iteration the matrix $\mathbf{W}$ satisfies positive definiteness, then for a symmetric $\Delta\mathbf{W}$, there exists an α_{max} for which decreasing α will preserve positive definiteness. The parameter ρ is set to 0.5 as a default value. We provide the algorithm with the input

$$\alpha_{(k),1} = -2\frac{\phi_{(k)}(0) - \phi_{(k-1)}(0)}{\phi'_{(k)}(0)}.$$

The "back-tracking algorithm" [96] implementing the Wolfe first condition searches for the best reduction in the merit function.

4.3.4 Solving CP_3

The optimization problem CP_3 is a linearly-constrained quadratic programming problem. The quadratic programming without the positive definiteness constraint could be easily solved. However, since we seek to find a proper prior distribution, a positive definite matrix is of interest, making the quadratic programming more challenging. To cope with this constraint, we add the logarithm of the determinant of the matrix to the objective function. The added term can be considered as a log-barrier function used to satisfy the constraint of having a matrix with positive determinant. Because the latter condition is only a necessary condition for positive definiteness, we still check the step size to be sure the search remains in the feasible region of positive definite matrices. Overall, the search space is more restricted, thereby leading to faster convergence. Hence, we simply add the term $\mu \log(|\mathbf{\Psi}|)$ to the objective function, the parameter μ being the barrier parameter.

4.3.5 Regularization Parameter

The parameter κ represents the spread of the prior, larger κ meaning that the prior is more centered about the scale matrix. Thus, κ can be viewed as the total amount of information in the prior. The regularization parameter aims at making a balance between two sources of information; (1) data through expected likelihood, and (2) slackness variables controlling the conditional entropy. λ_1 governs the relative importance of the slackness variables (information in the pathways) to the total information. We can view the total information, as represented by κ, as being a "sum" of the amount of data used to form the prior and a proportion of κ relating to the importance of the slackness variables. Under this heuristic $\kappa = n_p + \lambda_1\kappa$, so that $\lambda_1 = \frac{\kappa - n_p}{\kappa}$.

We can also view κ as a sum of the data used to form the prior and the amount of data, n_{pw}, that is "equivalent" to the pathway knowledge (recognizing that this "equivalence" is purely a heuristic notion). This leads to $\kappa = n_p + n_{pw}$. Inserting this expresion into the expression for λ_1 yields

$$\lambda_1 = \frac{n_{pw}}{n_p + n_{pw}}. \tag{4.24}$$

We are left with defining n_{pw}. In the simulations we let $n_{pw} = mp$ for different values of $m \geq 2$ and see that the performance is not very sensitive to m so that a default value could simply be $n_{pw} = 2p$.

Reflecting on the preceding heuristics we see that we are confronted with a standard problem in pattern recognition, how to regularize two conflicting factors. One thinks of the problem of adding a complexity term when dealing with model selection. We take the usual approach of applying some heuristics and then demonstrating the benefit of the regularization via simulation.

4.3.6 Differences Between RML and REML Methods

Although both the RML classifier method of [77] and the current REML prior construction method involve uncertainty classes, they are very different. The RML classifier is built using two estimates of the class-conditional distributions that are improved using the uncertainty classes. Moreover, there is no prior probability involved and prior knowledge in the form of finite uncertainty classes of distributions is utilized to improve classification accuracy. There are two key differences between the RML and REML methods: (1) the REML method is used to construct prior probabilities to be utilized by a Bayesian framework, e.g. optimal Bayesian classification, and the designed classifier is optimal with respect to the assumed model; (2) the RML classification rule needs a knowledge transformation, i.e. from biological pathways to a set of models, whereas the REML prior construction approach performs this knowledge transformation via the proposed optimization framework while automatically assigning probabilities to the models.

4.4 Simulations on Synthetic Examples

Our aim is to compute the true error associated with the OBC using the REML prior to examine the performance of the proposed prior construction approach. To perform the simulations, we need to fix the ground-truth model from which sample data are taken or pathways built up. We propose a method of generating synthetic pathways for a fixed model to serve as the true model governing the stochastic regulations in the network. Sample points are generated according to this model. The generated pathways and sample points are then used for classifier design as depicted in Figure 4.3.

For the simulations, we may have one or two sets of pathways, $\mathcal{G}$ or $(\mathcal{G}_0, \mathcal{G}_1)$, corresponding to one or two classes. The sample data are split into $S_{n_p}^{prior}$ and $S_{n_t}^{train}$

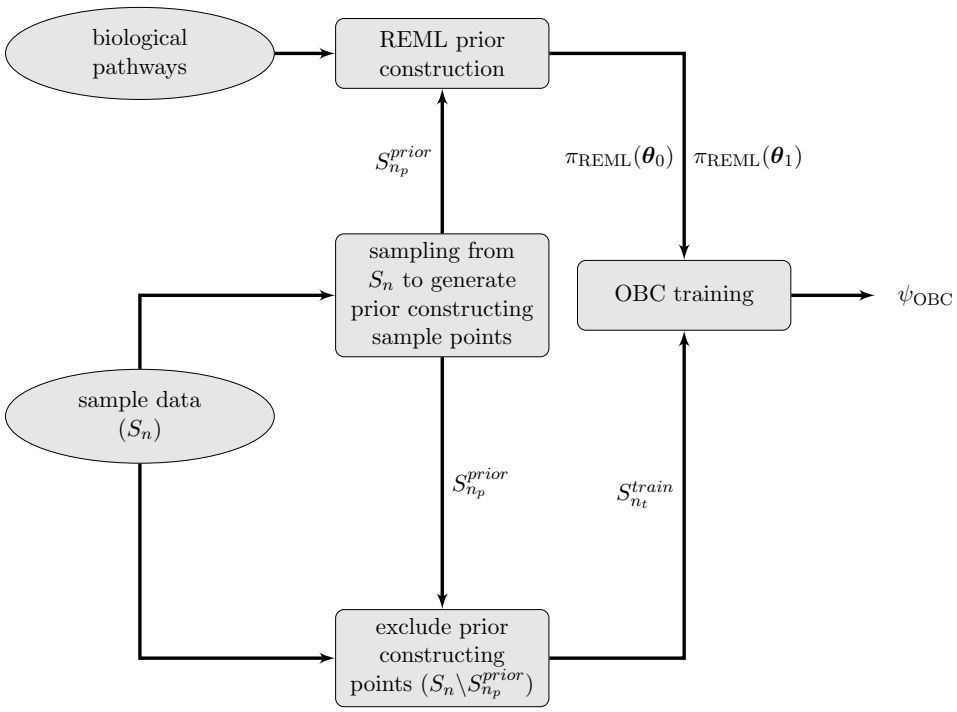


Figure 4.3: An illustrative view of the methodology of splitting sample data into two parts for the purpose of training the optimal Bayesian classifier. The training module is implemented using equation (5.2).

for prior construction and OBC training, respectively. The pathways are combined with $S_{n_p}^{prior}$ to construct the REML prior distribution. The constructed priors will be utilized with the rest of sample data $S_{n_t}^{train}$ to train the OBC. .

4.4.1 Generating Synthetic Pathways Inspired by Real Experiments

We propose a method to generate synthetic pathways with different amounts of incompleteness and uncertainty, controlled by the number of experiments and number of sample points in each experiment. Details for the Gaussian case are described in Algorithm 2, in which we assume an underlying ground-truth stochastic system governed by a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}^{true}, \boldsymbol{\Sigma}^{true})$. An experiment takes observations, $\mathbf{X}$, from this distribution. These observations, denoted by $S_{|\mathcal{O}|}$ in Algorithm 2, are used for pathway construction. Each experiment generates a set, $\mathcal{G}_i$, of signaling pathways.

Pathway construction is based on Coefficient of Determination (CoD) [97]. The CoD for a random variable x, considering the vector $\mathbf{y}$ as its predictor set, is defined by $\text{CoD}_{\mathbf{y}}(x) = (\varepsilon - \varepsilon_{\bullet})/\varepsilon$, where ε is the error of predicting x without observations, that is, based on its own statistics, and $\varepsilon_{\bullet}$ is the error of the optimal predictor of x based on $\mathbf{y}$. The CoD has been used since the early days of microarrays to analyze gene interaction [98]. Based on the observations $\mathbf{X}$, the model covariance matrix is estimated and the CoD is computed using the least minimum-mean-squared error (LMMSE) estimator. For entity $x(i)$, we denote all subsets of size k, excluding $x(i)$, by $G_k^{-x(i)}$. The best CoD-based set of size k is given by*

$$R_{x(i)} = \underset{\mathbf{y} \in G_k^{-x(i)}}{\arg\max}\ \text{CoD}_{\mathbf{y}}(x(i)).$$

To choose APS or RPS, denote the corresponding LMMSE estimate of $x(i)$ as

*In this work, we avoid self-regulation.

Algorithm 2 Synthetic Pathways Generation

Input: $\boldsymbol{\mu}^{true}, \boldsymbol{\Sigma}^{true}$,$r \in (50, 100)$, $k \geq 1$
Output: $\mathcal{G}$
for $i = 1$ to $|\mathcal{E}|$ **do**
  $S_{|\mathcal{O}|} \leftarrow$ take $|\mathcal{O}|$ Sample Points $\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}^{true}, \boldsymbol{\Sigma}^{true}); i = 1, ..., |\mathcal{O}|$
  CoD-based pathways construction using $S_{|\mathcal{O}|}$
  for $d = 1$ to p **do**
    $R_{x(d)} = \underset{\mathbf{y} \in \mathcal{X}_k^{-x(d)}}{\arg\max}\ \mathrm{CoD}_{\mathbf{y}}(x(d))$
    $\hat{x}_{\mathrm{LMMSE}}(d) =$ LMMSE of $x(d)$ based on $R_{x(d)}$ using $S_{|\mathcal{O}|}$
    use $\hat{x}_{\mathrm{LMMSE}}(d)$: positive/negative coefficient: APS/RPS started from $x(d)$
  end for
  build $\mathcal{G}_i$
end for
combine $\mathcal{G}_i$'s to build a consensus $\mathcal{G}$
for $d = 1$ to p **do**
  $A \leftarrow \emptyset$
  for $i = 1$ to $|\mathcal{E}|$ **do**
    $A \leftarrow$ collect all the Entities x in $\mathcal{G}_i$ for which we have an APS, $x(d) \longrightarrow x$, or RPS $x(d) \longrightarrow\!\!\mid x$
  end for
  $\tilde{A} \leftarrow$ count the repetitions in A, take union, and sort the elements based on their repetitions
  $k' \leftarrow$ average of the counts in $\tilde{A}$
  $\tilde{k} = \lfloor \max\{k', r|\mathcal{E}|/100\} \rfloor$
  $R_{x(d)} \leftarrow$ select the first $\tilde{k}$ elements from $\tilde{A}$
end for
build the consensus $\mathcal{G}$ using $R_{x(d)}, d \in \{1, ..., p\}$
return $\mathcal{G}$

$\hat{x}_{\mathrm{LMMSE}}(i)$. If the coefficient associated with a variable in this estimate is positive, then we assume APS; if it is negative, then we assume RPS. For example, if $\hat{x}_{\mathrm{LMMSE}}(1) = 0.3x(2) - 0.7x(3)$, then $x(1) \longrightarrow x(2)$ and $x(1) \longrightarrow\!\!\mid x(3)$.

Referring to Algorithm 2, each set, $\mathcal{G}_i$, of pathways is constructed via CoD maximization after observing sample points. Having $|\mathcal{E}|$ sample sets, we have $|\mathcal{E}|$ sets of pathways $\mathcal{G}_1, \mathcal{G}_2, ..., \mathcal{G}_{|\mathcal{E}|}$. These need to be combined to find a single consensus (similar to Figure 1). To build this consensus, for each entity its regulatory set is

the union of regulatory sets obtained in $\mathcal{G}_1, \mathcal{G}_2, ..., \mathcal{G}_{|\mathcal{E}|}$. This union is denoted by A in Algorithm 2. Then we find the most frequent entities (controlled by $\tilde{k}$) in A, as observed in the regulatory sets. Moreover, for a link to exist in the final consensus, it must be present in a certain percentage, $r\%$, of the $\mathcal{G}_i$'s. Knowing all the regulatory sets, a single consensus is constructed.

In our simulations, we build the ground-truth covariance matrix using a blocked structure proposed in [42] to model the covariance matrix of gene expression microarrays. Here, however, we place a small correlation between blocks. A 3-block covariance matrix with block size 3 has the structure

$$\Sigma = \begin{bmatrix} \mathbf{B}_1 & \mathbf{C} & \mathbf{C} \\ \mathbf{C} & \mathbf{B}_2 & \mathbf{C} \\ \mathbf{C} & \mathbf{C} & \mathbf{B}_3 \end{bmatrix}, \tag{4.25}$$

where

$$\mathbf{B}_i = \begin{bmatrix} \sigma^2 & \rho_i\sigma^2 & \rho_i\sigma^2 \\ \rho_i\sigma^2 & \sigma^2 & \rho_i\sigma^2 \\ \rho_i\sigma^2 & \rho_i\sigma^2 & \sigma^2 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} \rho_c\sigma^2 & \rho_c\sigma^2 & \rho_c\sigma^2 \\ \rho_c\sigma^2 & \rho_c\sigma^2 & \rho_c\sigma^2 \\ \rho_c\sigma^2 & \rho_c\sigma^2 & \rho_c\sigma^2 \end{bmatrix}, \tag{4.26}$$

σ^2 is the variance of each variable, $\rho_i, i = 1, 2, 3$, are the correlation coefficients inside blocks, and ρ_c is the correlation coefficient between elements of different blocks.

4.4.2 Simulation Setup

The more concentrated the prior distribution is around the value of $\boldsymbol{\theta} = [\boldsymbol{\theta}_0, \boldsymbol{\theta}_1]$ corresponding to the true feature-label distribution, the better should be the performance of the optimal Bayesian classifier. Since our aim herein is prior construction, we analyze the simulations in that light. Let the misclassification error of a designed classifier, $\psi : \mathbb{R}^p \to \{0, 1\}$, designed via feature-label distributions parameterized by

$\boldsymbol{\theta}$ be denoted by $\epsilon(\psi, \boldsymbol{\theta}) = \Pr(\psi(\mathbf{x}) \neq y|\boldsymbol{\theta})$.

Assume that we observe sample points $S_n = S_{n_p}^{prior} \cup S_{n_t}^{train}$. Denote the OBC designed according to REML priors constructed using $S_{n_p}^{prior}$ and training points $S_{n_t}^{train}$ by $\psi_{\text{OBC},n_t}^{n_p}$. We are concerned with $\epsilon_n(\psi_{\text{OBC},n_t}^{n_p}, \boldsymbol{\theta}_{true})$. If the solutions to the optimization paradigms stated in CP_2 and CP_3, shown by $\pi_{\text{REML}}^y = \pi_{\text{REML}}(\boldsymbol{\theta}_y), y = 0, 1$, produce good priors, that is, priors that have strong concentration around $\boldsymbol{\theta}_{true}$, then we should have $\epsilon_n(\psi_{\text{OBC},n_t}^{n_p}, \boldsymbol{\theta}_{true}) \leq \epsilon_n(\psi, \boldsymbol{\theta}_{true})$, where ψ is some other classifier, the exact relation depending on the feature-label distribution, classification rule, and sample size. On the other hand, if $\pi_{\text{REML}}^y; y = 0, 1$, are not concentrated around $\boldsymbol{\theta}_{true}$, then it may be that $\epsilon_n(\psi_{\text{OBC},n_t}^{n_p}, \boldsymbol{\theta}_{true}) > \epsilon_n(\psi, \boldsymbol{\theta}_{true})$.

Fixing the true feature-label distribution, we generate n points, composing $S_{n,i}$ in the i-th iteration, where, $i = 1, ..., M$. These points are split (randomly) into two parts, denoted by $S_{n_p,i}^{prior}$ and $S_{n_t,i}^{train}$, where $n_p + n_t = n$. Denote the given pathways by $\mathcal{G}$. Using $\mathcal{G}$ and $S_{n_p,i}^{prior}$, we construct prior distributions $\pi_{\text{REML},i}^y; y \in \{0, 1\}$. These are updated using the remaining points $S_{n_t,i}^{train}$ from which $\psi_{\text{OBC},n_t,i}^{n_p}$ is trained. The expected true error, $\epsilon_n(\psi_{\text{OBC},n_t}^{n_p}, \boldsymbol{\theta}_{true})$, is evaluated via Monte-Carlo simulations:

$$\epsilon_n(\psi_{\text{OBC},n_t}^{n_p}, \boldsymbol{\theta}_{true}) \approx \frac{1}{M} \sum_{i=1}^{M} \epsilon_n(\psi_{\text{OBC},n_t,i}^{n_p}, \boldsymbol{\theta}_{true}), \tag{4.27}$$

where the error term, $\epsilon_n(\psi_{\text{OBC},n_t,i}^{n_p}, \boldsymbol{\theta}_{true})$ is also computed via Monte-Carlo simulations with $10,000$ repetitions. The overall strategy, repeated through Monte-Carlo simulations, is partly shown in Figure 4.3, and implemented step-wise as follows:

1. Fix true parameterization for two classes: $[\boldsymbol{\mu}_y^{true}, \boldsymbol{\Sigma}_y^{true}], y \in \{0, 1\}$.

2. Use Algorithm 2 to generate two sets of pathways, $\mathcal{G}_y, y \in \{0, 1\}$.

3. Take observations from $\mathcal{N}(\boldsymbol{\mu}_y^{true}, \boldsymbol{\Sigma}_y^{true})$ to generate S_n.

4. Randomly choose n_p points from S_n for prior construction, i.e., $S_{n_p}^{prior}$, and the rest $S_{n_t}^{train}$ for training.

5. Use $S_{n_p}^{prior}$ and $\mathcal{G}_y$ to construct the prior $\pi_{\text{REML}}^y, y \in \{0,1\}$, by REML ($\text{CP}_2$ and CP_3).

6. Use (5.2) to optimally combine the priors, $\pi_{\text{REML}}^y, y \in \{0,1\}$, and $S_{n_t}^{train}$ to build the OBC, $\psi_{\text{OBC},n_t}^{n_p}$.

The parameters used in our simulations are summarized in Table 4.1. We considered a setting with $p = 8$ entities. The covariance matrix in the form of (4.25) is used with block sizes $3, 3, 2$ for the first, second, and the third blocks, respectively.

Table 4.1: Table of parameters used for simulations. Two configurations associated with two mean values are considered. Configurations C1, C2, C3 and C4 correspond to the Bayes errors of $\epsilon_{\text{Bayes}} = 0.167$, 0.155, 0.091, and 0.085, respectively.

Class y	$\boldsymbol{\Sigma}_y^{true}$	$\boldsymbol{\mu}_y^{true}$	c	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{O}\rvert$	ν_y	M
0	$\rho_1 = \rho_3 = 0.3$ $\rho_2 = -0.3, \rho_c = 0.1$	C1&C2: $0.3\mathbf{1}_p$ C3&C4: $0.5\mathbf{1}_p$	C1&C3: 0.5 C2&C4: 0.6	50	100	n_0^p	15000
1	$2\boldsymbol{\Sigma}_{1-y}^{true}$	C1&C2: $-0.3\mathbf{1}_p$ C3&C4: $-0.5\mathbf{1}_p$	C1&C3: 0.5 C2&C4: 0.6	50	100	n_1^p	15000

We compute the Monte-Carlo approximation of the expected true error of the designed OBC using the priors from $\text{CP}_2 + \text{CP}_3$ (shown by REML) and Jeffreys' prior. We also train LDA and QDA classifiers for the purpose of comparison. In the simulations, we fixed the true underlying model for two classes according to Table 4.1.

4.4.3 Results

We set λ_1 according to (5.30), $\lambda_2 = 0.5$, and consider three sample sizes, $n \in \{30, 50, 70\}$, and two class prior probabilities $c \in \{0.5, 0.6\}$. The sample sizes n_0 and n_1 are determined according to the class prior probability as $n_0 = cn$, and $n_1 = n - n_0$. We consider $\kappa_y = mp + n_y^p, m = 2, 3, 4$. We change the ratio of the number of sample points used for prior construction to the total sample size, $r_p = \frac{n_0^p + n_1^p}{n}$, from 0.1 to 0.9. We consider at most 90% to keep points for prior update and finding the posterior. The sample sizes allocated for prior construction are determined as $n_y^p = \lceil r_p n_y \rceil, y = 0, 1$. For example, for $c = 0.6$ and $n = 30$, when 50% of the points are used for prior construction, $n_0^p = 9$ and $n_1^p = 6$.

The results for the settings in Table 4.1 for $m = 2$ are shown in Figure 4.4. Since we split the data, n_p for REML prior construction and $n - n_p$ to design the OBC from the REML prior, we need to examine the effect of n_p. Therefore, we plot the expected true error as a function of the percentage of the data points used for prior construction, $100 \times \frac{n_0^p + n_1^p}{n}\%$, for the OBC using the REML prior. The work-flow is depicted in Figure 4.5, in which there are two general possibilities: (1) use all data points for prior construction, shown in the hypotenuse of the figure, or (2) use part of the data for prior construction and the rest for constructing the posterior.

We compare these results to both quadratic discriminant analysis (QDA) and linear discriminant analysis (LDA). QDA is the plug-in classifier for the Gaussian model with different covariance matrices, meaning that it is obtained from the Bayes (optimal) classifier for the true model by estimating the means and covariance matrices by the sample mean and sample covariance matrices, respectively. LDA is the plug-in classifier for the Gaussian model with common covariance matrix. With small samples, LDA often performs better than QDA in the different-covariance model on

account of better estimation using the pooled sample covariance matrix for LDA. We also consider the OBC with Jeffreys' non-informative prior. Since there is no data splitting for QDA, LDA, and the OBC with a non-informative prior, all sample points are used for classifier construction so that the plots in Figure 4.4 are constant.

In Figures 4.4(a), 4.4(d), 4.4(g), and 4.4(j), $n = 30$, by increasing number of sample points used for the prior construction, the true error decreases. Thus, one should use at least 90% of the sample points for prior construction. However, when the total number of sample points increases, from 30 to 70, there is an optimal number of points which should be utilized for the prior construction. For instance, as illustrated in Figures 4.4(c), 4.4(f), 4.4(i), 4.4(l), after about $n_p = 30$, the true error of the designed OBC increases. Note that in Figure 4.4(k), the LDA classifier outperforms the OBC. Here we must remind ourselves that the OBC is optimal on average relative to the prior distribution, but may be outperformed for individual distributions. Even in this case, however, the REML-based OBC still significantly outperforms the OBC with Jeffreys' prior. Rsults for $m = 3, 4$ are provided on the companion website.

The simulations demonstrate that splitting the data provides better performance – that is, using n_p points ($n_p < n$) to design the prior and the remaining $n - n_p$ points to train the OBC provides the minimum expected error. To be precise, for a given n, we are interested in the number of sample points for which the minimum expected true error is achieved using the OBC designed via the REML prior, namely,

$$n_p^*(n) := \underset{n_p \in \{2,\ldots,n\}}{\arg\min}\, \epsilon_n(\psi_{\mathrm{OBC},n-n_p}^{n_p}, \boldsymbol{\theta}_{true}).$$

$n_p^*(n)$ represents the REML-optimal investment of data size in the prior construction process. After this point, the remaining points should be employed to update the

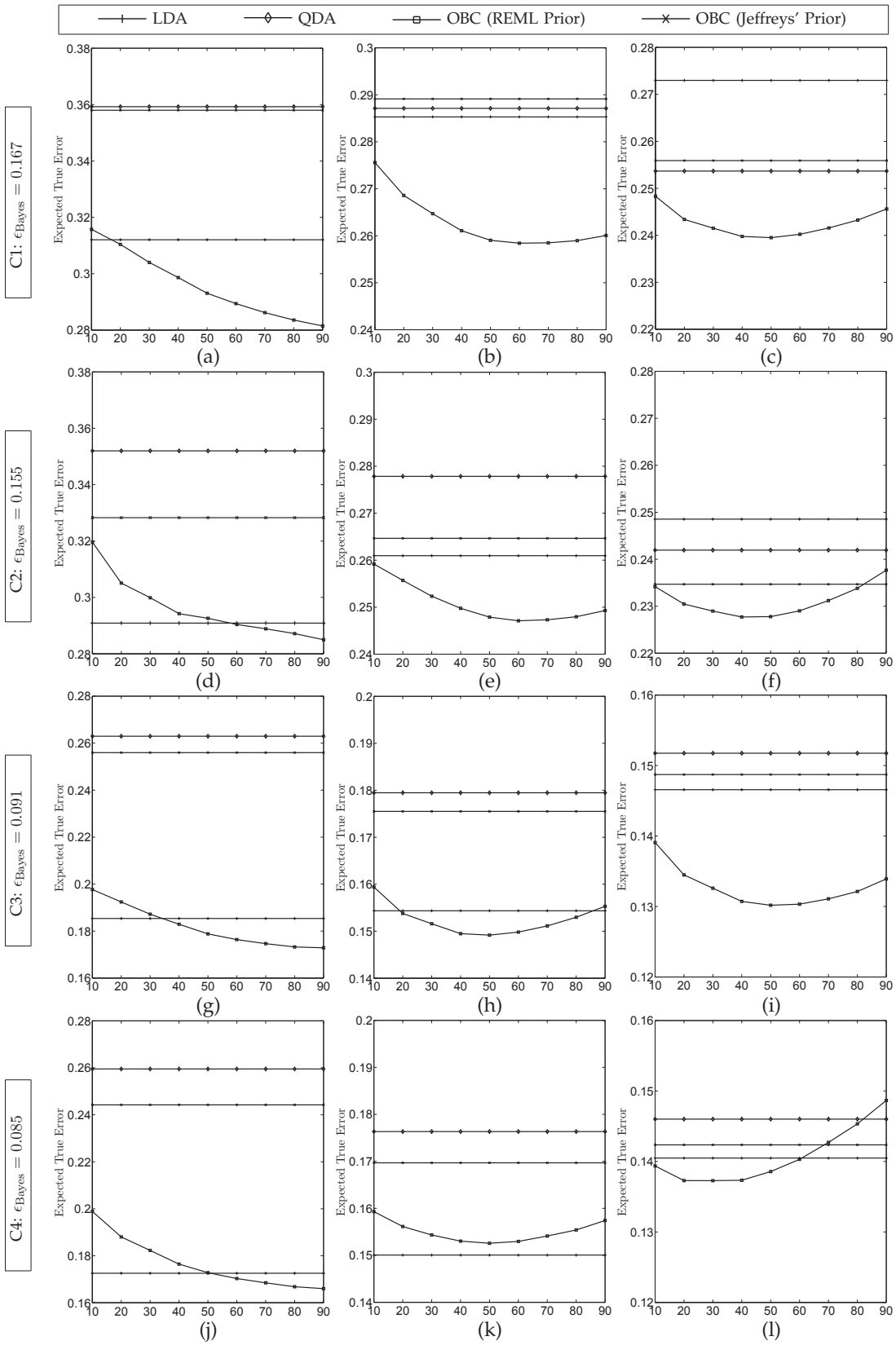


Figure 4.4: The expected true error as a function of the percentage of the sample points used for prior construction, $\frac{n_0^p + n_1^p}{n}$ (%), shown in the $x-$axis. Sample points for each class are stratified according to $c = \Pr(y = 0)$.

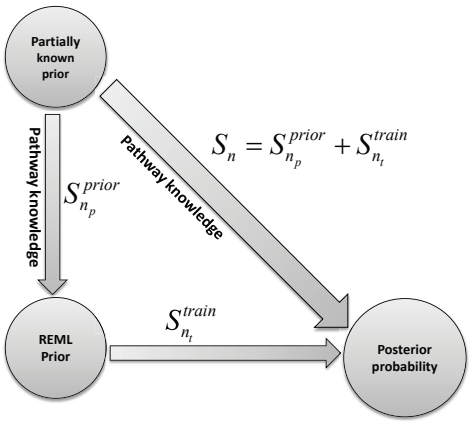


Figure 4.5: A schematic view of two possibilities starting from a partially known prior probability, i.e., in the Normal-Wishart prior in this dissertation, we assume known ν and κ. First, using some part of the data for prior construction, and then using the rest for finding the posterior probability. Second, utilizing all the data points with the pathways to find a prior knowledge, or precisely the posterior probability.

constructed prior. Since there is no closed form for the true error of the OBC designed using the REML prior, the exact value of n_p^* cannot be determined. Thus, we search for n_p^* via Monte-Carlo simulations: For fixed n, we exhaustively search for $n_p^*(n)$ by increasing n_p from 2 to n. We only consider configurations C1 and C3 for which $c = 0.5$. Tables 4.2 and 4.3 demonstrate n_p^* as a function of total sample size n for four scenarios $m = 2, 3, 4$. Sample size n is changed from 30 to 70.

The key point is that increasing n does not necessarily lead to a larger n_p^*; on the contrary, there is a saturation point after which increasing n does not significantly influence the optimal sample size for prior construction. For both tables (reflecting $\epsilon_{\text{Bayes}} = 0.167$ and $\epsilon_{\text{Bayes}} = 0.091$), the optimal value is approximately $n_p^* \approx 30$, with small variations around 30 having negligible effect on classifier performance. This means that 30 points for prior construction provides close to optimal performance when $n \geq 30$, so that we can view the REML prior as taking up to 30 sample points

Table 4.2: The optimal prior constructing sample size, n_p^* as a function of total sample size for the configuration C1.

κ_y \ n	30	34	38	42	46	50	54	58	62	66	70
$\kappa_y = 2p + n_y^p$	26	28	34	32	26	28	32	28	28	28	32
$\kappa_y = 3p + n_y^p$	28	30	30	34	28	28	30	24	26	34	28
$\kappa_y = 4p + n_y^p$	30	28	32	38	24	34	32	34	30	28	32

Table 4.3: The optimal prior constructing sample size, n_p^* as a function of total sample size for the configuration C3.

κ_y \ n	30	34	38	42	46	50	54	58	62	66	70
$\kappa_y = 2p + n_y^p$	30	26	28	28	32	28	28	26	28	28	30
$\kappa_y = 3p + n_y^p$	26	28	26	30	30	26	28	28	28	28	28
$\kappa_y = 4p + n_y^p$	30	28	28	30	24	28	28	32	28	28	30

for its construction, after which further sample points, however many there be, are used for posterior construction. Should we have $n < 30$, using prior information is still superior to a completely data-driven classifier; however, classifier design is strictly from the prior without using a posterior to design the OBC. We would like a closed form for the true error of the OBC designed using the REML prior, but this problem appears difficult given the nature of the prior information and the optimization problems involved in prior construction.

4.5 An Example Inspired by the Colon Cancer Pathway

4.5.1 Pathway Description

In this section, we evaluate the performance of the proposed method on real pathways. These pathways, associated with colon cancer, are depicted in Figure 4.6. This is a diagram that includes three basic pathways: the Ras/Raf/Mek pathway at the left and middle in red, the PI3K pathway in the middle in blue, and the JAK/STAT pathway on the right in green. On the top are the ligands/stimulation

factors, EGF, HGF, NRG1, IL6, etc. They carry the external signals generated by neighboring cells (sometimes themselves). Immediately under the factors are the ligand receptors, which are anchored at the membrane. Once the ligand binds to its receptor, it will initiate the downstream process, usually to form a dimer or similar complex so that the kinase in one unit can activate the other unit. If two nodes are drawn as attached together, they normally closely bind together to form a dimer. For example, EGFR-ERBB2 is a heterodimer. MET-MET is an homodimer. Or it means the interaction is conducted in a physically specific location. For example, we believe JAK-SHP2-SOS must be three proteins interacting in a place very close to the membrane where the receptor IS6ST-IS6R is located. These reactions happen very fast once the ligand binds to the receptor. A long arrow means that the protein will move into the cytoplasm and activate/inhibit/modify the target protein. As indicated in the legend, "+P" means phosphorylation and "T" means transcription. "GAP" means GTPase-activating protein, because RHEB is a GTPase. So this is an activating process. Overall, phosphorylation and GTPas-activating are both protein modification procedures that happen very fast since they involve no transcription/translation. In particular, such a process cannot be observed in a transcription assay, such as a microarray or RNAseq.

From the wiring diagram in Figure 4.6, we concentrate on 11 entities: EGF, Ras, MEK1/2, PIK3CA, STAT3, mTORC1, HGF, IL6, PKC, SPYR4, and TSC1/TSC2. Thus, the feature vector is

$$\mathbf{x} = [\text{EGF},\ \text{HGF},\ \text{IL6},\ \text{Ras},\ \text{PIK3CA},\ \text{STAT3},\ \text{TSC1/TSC2},\ \text{mTORC1},\ \text{SPYR4},\ \text{PKC},\ \text{MEK1/2}],$$

where the TSC1/TSC2 tumor suppressor complex is considered as a single entity. Since we do not exactly know what type of functioning exists for each of these

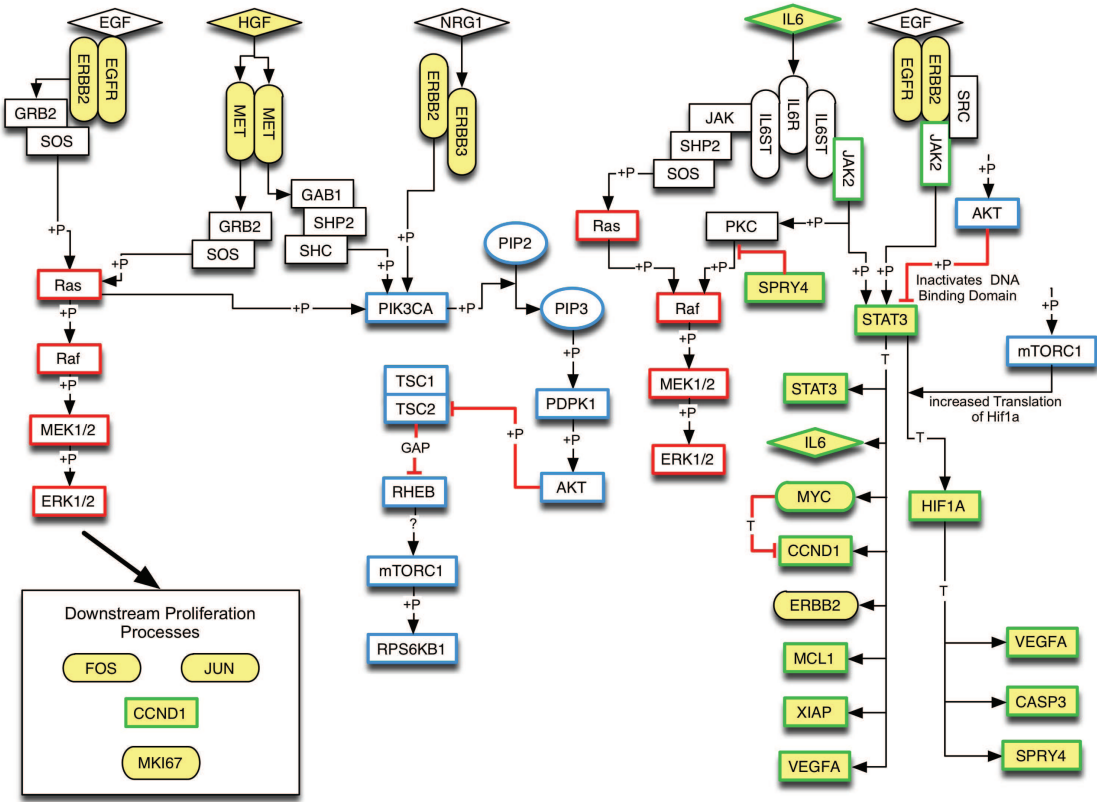


Figure 4.6: A wiring diagram showing proliferation and survival pathway elements whose transcriptional states could be altered in a cell exposed to the drug lapatinib [3, 4]. Nodes marked in yellow are ones for which a reporter would be used to assess transcription for that gene. The places where the drug of interest and other drugs that act at other points on these pathways are indicated by red labels.

genes, we simply assign some dependency between these genes according to the pathways in Figure 4.6. We do a similar procedure for both classes, assuming some mutations or changes for the malfunctioning label. From these assumed dependencies, we construct covariance matrices for two classes.

4.5.2 Pathway-Consistent True Model Construction

As discussed in Section 4.4.1, the basic requirements for the numerical experiments are pathways for two classes and sample data generated using a fixed true model. In the Gaussian case, we need to fix the underlying true mean vector and covariance matrix, $\boldsymbol{\mu}_y^{true}, \boldsymbol{\Sigma}_y^{true}$, for $y \in \{0, 1\}$. Owing to the structure of the colon cancer pathways in Figure 4.6, we first set the covaraince matrix restricted to the top genes [EGF, HGF, IL6], i.e. $[x(1)\ x(2)\ x(3)]$. We denote the mean vector and covariance matrix restricted to these three genes by $[\boldsymbol{\mu}_0^{true}]_{[x(1)\ x(2)\ x(3)]}$ and $[\boldsymbol{\Sigma}_0^{true}]_{[x(1)\ x(2)\ x(3)]}$, respectively. Thus, for class $y = 0$,

$$[\text{EGF},\ \text{HGF},\ \text{IL6}] \sim \mathcal{N}([\boldsymbol{\mu}_0^{true}]_{[x(1)\ x(2)\ x(3)]}, [\boldsymbol{\Sigma}_0^{true}]_{[x(1)\ x(2)\ x(3)]}).$$

We assume

$$[\boldsymbol{\mu}_1^{true}]_{[x(1)\ x(2)\ x(3)]} = -[\boldsymbol{\mu}_0^{true}]_{[x(1)\ x(2)\ x(3)]}$$

and

$$[\boldsymbol{\Sigma}_1^{true}]_{[x(1)\ x(2)\ x(3)]} = 2[\boldsymbol{\Sigma}_0^{true}]_{[x(1)\ x(2)\ x(3)]}.$$

Then, to keep the Gaussianity, for both classes we assume linear dependencies in the form of $x(i) = \mathbf{a}_i^T \mathbf{x}_{i-1} + z_i; i = 4, 5, ..., 11$, where the vectors $\mathbf{a}_i, i = 4, ..., 11$, are coefficients determining the influence of each gene in $\mathbf{x}_{i-1} = [x(1), ..., x(i-1)]^T$ on the target gene $x(i)$. The z_is are additive zero-mean Gaussian noise, $z_i \sim N(0, \sigma_i^2)$, considered to model the effects of latent variables outside the model [69, 99].

Having Figure 4.6 as the foundation for the pathways, if two genes are not connected we simply assume that the corresponding coefficient in the vector $\mathbf{a}_i$ is zero. If there is an APS/RPS, we assume a positive/negative coefficient, respectively. Hence, considering the normal functioning of the cell, we consider the following linear relationships among the variables conditioned on being in class $y = 0$:

$$\begin{aligned}
\text{Ras} &= a_4(1)\text{EGF} + a_4(2)\text{HGF} + a_4(3)\text{IL6} + z_4 && \text{(4.28a)}\\
\text{PIK3CA} &= a_5(2)\text{HGF} + a_5(4)\text{Ras} + z_5 && \text{(4.28b)}\\
\text{STAT3} &= a_6(1)\text{EGF} + a_6(3)\text{IL6} + a_6(5)\text{PIK3CA} + z_6 && \text{(4.28c)}\\
\text{TSC1/TSC2} &= a_7(5)\text{PIK3CA} + z_7 && \text{(4.28d)}\\
\text{mTORC1} &= a_8(7)\text{TSC1/TSC2} + z_8 && \text{(4.28e)}\\
\text{SPRY4} &= a_9(6)\text{STAT3} + a_9(8)\text{mTORC1} + z_9 && \text{(4.28f)}\\
\text{PKC} &= a_{10}(3)\text{IL6} + a_{10}(9)\text{SPRY4} + z_{10} && \text{(4.28g)}\\
\text{MEK1/2} &= a_{11}(4)\text{Ras} + a_{11}(10)\text{PKC} + z_{11} && \text{(4.28h)}
\end{aligned}$$

in which, those $a_i(j)$'s, not contributing in equations (4.28a)-(4.28h), are set to zero. Moreover, for nonzero coefficients, except $a_6(5)$, $a_8(7)$, and $a_{10}(9)$, all other coefficients are positive. The other difference we assume for distinguishing two classes is a mutation for the TSC1/TSC2 tumor suppressor complex [100–102]. Precisely, for $y = 1$, we change equation (4.28d) to $x(7) = \text{TSC1/TSC2} = z_7$, meaning that this gene is stuck at 0 with a small probability of being changed. Considering the conditional entropy constraints, we extract the regulatory set connections used for the REML prior construction in Table 4.4 for the two classes. We set $|a_i(j)| = \frac{1}{N_i}$, where N_i is the number of nonzero elements of $\mathbf{a}_i$. The sign is determined based on whether the influence is through an APS or an RPS. For example, for STAT3: $a_6(1) = a_6(2) = \frac{1}{3}$ and $a_6(3) = -\frac{1}{3}$.

Table 4.4: Regulatory sets of the genes considered in our classification scenario using pathways in Figure 4.6. The second and the third columns correspond to two classes $y = 0$ and $y = 1$, respectively. The only mutation considered to distinguish two classes is in TSC1/TSC2 complex which is stuck at zero.

Gene	Regulatory set ($y = 0$)	Regulatory set ($y = 1$)
EGF	{Ras, STAT3}	{Ras, STAT3}
HGF	{Ras, PIK3CA}	{Ras, PIK3CA}
IL6	{Ras, STAT3, PKC}	{Ras, STAT3, PKC}
Ras	{MEK1/2, PIK3CA}	{MEK1/2, PIK3CA}
PIK3CA	{STAT3, TSC1/TSC2}	{STAT3}
STAT3	{STAT3, IL6, SPRY4}	{STAT3, IL6, SPRY4}
TSC1/TSC2	{mTORC1}	{mTORC1}
mTORC1	{SPRY4}	{SPRY4}
SPRY4	{PKC}	{PKC}
PKC	{MEK1/2}	{MEK1/2}
MEK1/2	$\emptyset$	$\emptyset$

The coefficients for the true model used for simulations are given in Table 4.5, $\mathbf{1}_m$ denotes an all-one column-vector with dimension m. Then, according to these coefficients, using equations (4.28a)-(4.28h), we build the underlying true mean vectors and covariance matrices for both classes. These moments will be used to generate data points during our simulations, i.e., $\mathbf{x} \sim c\mathcal{N}(\boldsymbol{\mu}_0^{true}, \boldsymbol{\Sigma}_0^{true}) + (1-c)\mathcal{N}(\boldsymbol{\mu}_1^{true}, \boldsymbol{\Sigma}_1^{true})$, where c is fixed in our simulations to 0.5. We also have $p = 11$.

4.5.3 Results

Similar to Section 4.4.3, we show the results for different sample sizes, but for a single Bayes error $\epsilon_{\text{Bayes}} = 0.108$ in Figure 4.7, which shows the comparisons for sample sizes $n = 30$, 50, and 70, with $n_0 = n_1 = n/2$ and $m = 2$ ($m = 3, 4$ on

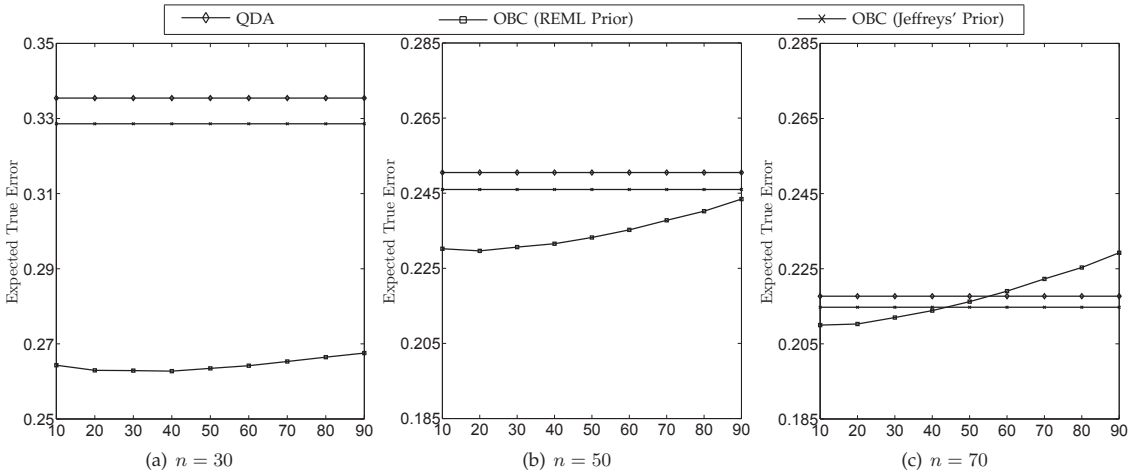
Table 4.5: Table of parameters used for simulations. The Bayes error is $\epsilon_{\text{Bayes}} = 0.132$.

Class y	$[\boldsymbol{\Sigma}_y^{true}]_{[x(1)\ x(2)\ x(3)]}$	$\boldsymbol{\mu}_y^{true}$	Noise variance
0	$\begin{bmatrix} 1 & 0.2 & 0.2 \\ 0.2 & 1 & 0.2 \\ 0.2 & 0.2 & 1 \end{bmatrix}$	$0.3\mathbf{1}_p$	$\sigma_i^2 = 0.2,\ i = 1, \ldots, 8$
1	$\begin{bmatrix} 2 & 0.4 & 0.4 \\ 0.4 & 2 & 0.4 \\ 0.4 & 0.4 & 2 \end{bmatrix}$	$-0.3\mathbf{1}_p$	$\sigma_i^2 = 0.05,\ i \neq 7$

the companion website). We have removed the line for LDA, since the LDA error is so large that the differences between the other methods could not be easily seen. One can see that the superiority of the OBC designed using the constructed prior diminishes as number of sample points increases, with only small improvement over QDA. Nonetheless, we see a significant improvement in the small sample settings ($n \leq 50$), which is our ultimate goal.

4.6 Discussion

Purely data-driven approaches to classifier design with small samples tend to produce poor classifiers whose errors cannot be reliably estimated. The importance of small-sample classification is highlighted by its prevalence in genomic/proteomic applications. In general, prior (probability) selection is one of the main challenges when one is dealing with any Bayesian framework. Conjugate priors are of great interest because of their convenient properties for deriving the posterior probabilities; however, there is no general rigorous mathematical machinery from which to estimate the hyperparameters. The proposed optimization framework is different from its predecessors in the sense that the REML prior relies on sample data and incorporates these data with "pure prior knowledge" to obtain a prior probability. The objective function is based on the notion of a model selection criterion, where the criterion is



(a) $n = 30$ (b) $n = 50$ (c) $n = 70$

Figure 4.7: The expected true error as a function of the percentage of the sample points used for prior construction in the biological pathways shown in Figure 4.6. The $x-$axis shows $\frac{n_0^p+n_1^p}{n}$ (%). Three sample sizes are considered: $n = 30$, $n = 50$, and $n = 70$. The expected true errors for the LDA classification rule from left to right are 0.434, 0.414, and 0.404, respectively. The parameter $\kappa_y = 2p + n_y^p$ is fixed for these results.

marginalized using the prior probability. The performance of the designed prior is examined by evaluating the true error of the optimal Bayesian classifier designed via the posterior.

As a final comment, let us note that the overarching goal is to use prior knowledge, in the form of biological pathways, to assist in the design of genomic classifiers. Since we use some initial data in prior construction and thereafter use new data to construct a posterior distribution in the Bayesian framework, one might consider this a "hybrid" approach. But from the perspective of our goal, integration of pathway knowledge and data, this characterization is semantic. The fundamental conclusion is that pathway knowledge and data are used in a Bayesian framework to produce classifiers that are superior to those based on data alone, and this is done via an optimization procedure that transforms the pathway knowledge into constraints on

the feature-label distribution.

5. DIRICHLET PRIOR CONSTRUCTION ON MULTINOMIAL DISTRIBUTION

In this section, we consider the problem of prior probability construction for the purpose of learning an optimal Bayesian classifier when the underlying model is discrete. First, we introduce the notion of objective-based priors, when the prior information is in the form of signaling pathways. Unlike the previously used methods where the prior information is either in the form of known inequalities or equalities, we consider the notion of "slackness." In order to bring the slackness variables, the interactions in the pathways are quantified from a Bayesian perspective, "mapping the signaling pathways to a set of constraints on the hyperparameter space." Then, we extend maximum entropy and maximal data information prior to the proposed framework. Moreover, a recently introduced method of prior construction, regularized expected mean log-likelihood, is also revisited. Our problem of interest in this part is discrete classification, and hence we consider the optimal Bayesian classification when the likelihood function results from a multinomial distribution. All the methods are studied for Dirichlet prior families. We examine the proposed framework on a simplified set of pathways involving the TP53 gene. We show that the Bayesian framework utilizing the informative constructed priors via objective-based priors framework significantly outperforms those rules which do not incorporate prior knowledge.

5.1 Background

5.1.1 Optimal Bayesian Classifier

Given a binary classification problem with classes $y \in \{0, 1\}$, we observe a collection of n sample points, S_n, in a sample space $\mathcal{X}$, with n_y i.i.d. points from each

class. Call c the *a priori* probability that an individual sample point, $\mathbf{x}$ ($\in \{0,1\}^p$ or $\in \mathbb{R}^p$ in a $p-$dimensional binary or continuous case, respectively), is from class 0, and let the class-conditional distribution for class y, denoted $p_{\boldsymbol{\theta}_y}(\boldsymbol{x}|y)$ (sometimes f is used for continuous cases), be parameterized by $\boldsymbol{\theta}_y$. The feature-label distribution is completely specified by the modeling parameters $\boldsymbol{\theta} = [c, \boldsymbol{\theta}_0, \boldsymbol{\theta}_1]$. In [78], it is assumed that c, $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_1$ are all independent prior to observing the data. Denoting the prior for $\boldsymbol{\theta}_y$ by $\pi(\boldsymbol{\theta}_y)$, we have $\pi(\boldsymbol{\theta}) = \pi(c)\pi(\boldsymbol{\theta}_0)\pi(\boldsymbol{\theta}_1)$. Moreover, the posterior preserves this independence and denoting it by $\pi^*(\boldsymbol{\theta}_y)$, the following is obtained [39]

$$\pi^*(\boldsymbol{\theta}_y) \propto \pi(\boldsymbol{\theta}_y) \prod_{i=1}^{n_y} p_{\theta_y}(\mathbf{x}_i^y|y) \tag{5.1}$$

where $\mathbf{x}_i^y$ is the $i-$th sample point in class y.

Assume a discrete sample space with b bins, i.e. $\mathcal{X} = \{1, ..., b\}$. Let p_k^0 and p_k^1 be the class-conditional probabilities in bin $k \in \{1, ..., b\}$ for class 0 and 1, respectively, and define u_k^y to be the number of sample points observed in bin $k \in \{1, ..., b\}$ from class $y \in \{0, 1\}$. The class sizes are thus $n_y = \sum_{k=1}^{b} u_k^y$. A general discrete classifier assigns each bin to a class.

The discrete Bayesian model defines $\boldsymbol{\theta}_y = [p_1^y, ..., p_b^y]$ where only $b-1$ of these are independent. The parameter space of $\boldsymbol{\theta}_y$ is defined to be the set of a valid bin probabilities, $[p_1^y, ..., p_b^y] \in \boldsymbol{\Theta}_y$ if and only if $0 \leq p_k^y \leq 1$ with $\sum_{k=1}^{b} p_k^y = 1$. Then, considering Dirichlet priors with parameters $\boldsymbol{\alpha}^y, y \in \{0, 1\}$ (refer to Appendix D.1 for the definition) for two classes, the OBC is given by (refer to [39] for more details)

$$\psi_{\text{OBC}}(k) = \begin{cases} 0 \; if E_{\pi^*}[c] \frac{u_k^0 + \alpha_k^0}{n_0 + \alpha_0^0} \geq (1 - E_{\pi^*}[c]) \frac{u_k^1 + \alpha_k^1}{n_1 + \alpha_0^1} \\ 1 \; o.w. \end{cases} . \tag{5.2}$$

Tracing back the formula for the optimal Bayesian classifier or that of any Bayesian framework, it is observed that the prior probabilities parameters (hyperparameters) play the central role, thereby, one could see that the fundamental component prior to use of this Bayesian approach is to have an accurate prior probability. In what follows we drop the sub (sup)-script denoting the dependency to the label, y, but one should notice that the prior knowledge is assumed to be available for both classes, separately. Now, here is the question: How can one construct prior probabilities $\pi(\boldsymbol{\theta}_y),\ y = 0, 1$? In the next part, we give a formal definition for the problem of prior construction.

5.2 Objective-based Informative Priors

The Jeffreys' prior was the first attempt for constructing priors after about 200 years of using uniform priors of Bayes and Laplace [44, 45]. The Jeffreys' prior is the one with the following property [45]

$$\pi(\boldsymbol{\theta}) \propto \sqrt{\mathcal{I}[\boldsymbol{\theta}]} = \sqrt{\det\Big[E[\frac{\partial^2}{\partial\theta_i\theta_j}\log f(x|\boldsymbol{\theta})\Big]_{i,j}} \tag{5.3}$$

where $\mathcal{I}(\boldsymbol{\theta})$ is the *Fisher Information* of the parameter $\boldsymbol{\theta}$. The main drawback of the Jeffreys' prior is that in many cases it yields an improper prior which does not generate a proper posterior probability. It also shows some other unexpected behaviors which limits its applicability [103]. Nonetheless, in the discrete scenario of multinomial model for the likelihood, one would obtain a proper prior given by

$$\pi(\theta_i) \propto {\theta_i}^{-1/2}, \tag{5.4}$$

which is equivalent to a Dirichlet distribution with $\alpha_i = \frac{1}{2}, \forall i = 1, \ldots, b$.

Unlike the Jeffreys' prior, all the attempts for constructing prior probabilities in

the last seven decades have been concentrated on an optimization view for the prior construction where the objective function is chosen so that it represents some type of measure of the information. In this work, we are mainly interested in regularization framework, where we introduce it in the general framework. The notion of regularization goes back to solving ill-posed integral equation by Tikhanov [31]. After Tikhanov, there has been an extensive number of works either in statistics, signal processing, or machine learning dealing with regularization [28, 30, 104].

In the general framework, we define a *regularized objective-based informative prior probability* as the solution to the following optimization problem

$$\begin{aligned} &\min_{\pi(\boldsymbol{\theta})} \quad -(1-\boldsymbol{\lambda}^T\mathbf{1})E_{\boldsymbol{\theta}}[g_0(\boldsymbol{\theta})] + \boldsymbol{\lambda}^T\boldsymbol{L}(\boldsymbol{\xi}), \\ &\text{Subject to: } \begin{cases} \mathbf{0}_m \preceq \boldsymbol{\xi} \\ E_{\boldsymbol{\theta}}[g_i(\boldsymbol{\theta})] \leq \xi_i; i = 1, ..., m \\ \int_{\boldsymbol{\Theta}} \pi(d\boldsymbol{\theta}) = 1 \end{cases} \end{aligned} \tag{5.5}$$

in which the function $E_{\boldsymbol{\theta}}[g_0(\boldsymbol{\theta})]$ is an information measuring term, e.g. the negative entropy where the function $g_0(\boldsymbol{\theta}) = \ln \pi(\boldsymbol{\theta})$, and $\boldsymbol{L}(\boldsymbol{\xi})$ is a linear function on the slackness variables $\boldsymbol{\xi}$. The vector $\boldsymbol{\xi}$, encompasses the slackness variables which are also optimization parameters, i.e. it is an ordered representation integrating all the variables in the forms of $\xi_i^a,\ \xi_i^r,\ \xi_{ij}^{ca},\ \xi_{ij}^{cr},\ \xi_i^{reg}$. Define,

$$\boldsymbol{L}(\boldsymbol{\xi}) = [\sum_{i=1}^{|\mathcal{C}|} \xi_i^{reg},\ \sum_{(ij)\in\mathcal{G}_a} \xi_{ij}^a + \sum_{(ij)\in\mathcal{G}_r} \xi_{ij}^r + \sum_{(ijk)\in\mathcal{G}_{ca}} \xi_{ijk}^{ca} + \sum_{(ijk)\in\mathcal{G}_{cr}} \xi_{ijk}^{cr}]$$

In (5.5), the vector $\boldsymbol{\lambda} = [\lambda^{reg}, \lambda^{fun}]$, for which we have $\boldsymbol{\lambda}^T\mathbf{1} \leq 1$ and $\boldsymbol{\lambda} \succeq \mathbf{0}$, is the regularization vector (or the design parameter) depending on the relative importance

of different sources of information. The constraints $E_{\boldsymbol{\theta}}[g_i(\boldsymbol{\theta})]; i = 1, \ldots, m$, are extracted from the prior knowledge. In the case of restricted prior probability family, Π, parametrized by a vector $\boldsymbol{\alpha}$ (hyperparameters), we define

$$f_0(\boldsymbol{\alpha}, \boldsymbol{\xi}) = -(1 - \boldsymbol{\lambda}^T \mathbf{1}) E_{\boldsymbol{\theta}}[g_0(\boldsymbol{\theta})|\boldsymbol{\alpha}] + \boldsymbol{\lambda}^T \boldsymbol{L}(\boldsymbol{\xi})$$
$$f_i(\boldsymbol{\alpha}, \boldsymbol{\xi}) = E_{\boldsymbol{\theta}}[g_i(\boldsymbol{\theta})|\boldsymbol{\alpha}] - \xi_i \leq 0; i = 1, ..., m.$$

Then, the optimization problem can be rewritten as follows

$$\begin{aligned} &\min_{\pi(\boldsymbol{\theta}|\boldsymbol{\alpha}) \in \Pi} \; f_0(\boldsymbol{\alpha}, \boldsymbol{\xi}) \\ &\text{Subject to: } \begin{cases} f_i(\boldsymbol{\alpha}, \boldsymbol{\xi}) \leq 0; i = 1, ..., m \\ \mathbf{0}_m \preceq \boldsymbol{\xi} \end{cases} \end{aligned} \tag{5.6}$$

where Π is the feasible region to which the prior distribution belongs. From equation (5.6), one can see that in the parametric prior family, e.g. Dirichlet distributions, the objective function and constraints are reduced to functions of only the parameter vector $\boldsymbol{\alpha}$ and the slackness variables. Since the regularization parameters are used to make a balance between different sources of information, we assume that for each "type" of prior knowledge, the corresponding element in the vector $\boldsymbol{\lambda}$ are equal. In other words, for all ξ^a_{ij}, ξ^r_{ij}, ξ^{ca}_{ijk}, and ξ^{cr}_{ijk} we assume one regularization parameter denoted by λ^{fun} to emphasize on the "functional" essence of this type of information. Similarly for the regulatory set information, we use λ^{reg}. Hence, the term $\boldsymbol{\lambda}^T \boldsymbol{L}(\boldsymbol{\xi})$, in equation (5.6) can be expanded as follows

$$\lambda^{reg} \sum_{i \in C} \xi^{reg}_i + \lambda^{fun} \Big[\sum_{(i,j) \in \mathcal{G}_a \cup \mathcal{G}_r} \xi^a_{ij} + \xi^r_{ij} + \sum_{(i,j,k) \in \mathcal{G}_{ca} \cup \mathcal{G}_{cr}} \xi^{ca}_{ijk} + \xi^{cr}_{ijk} \Big] \tag{5.7}$$

The constraints in the general objective-based informative prior framework are as

follows

$$E_{\boldsymbol{\theta}}\left[\Pr(x_j=1|x_i=1,\boldsymbol{x}_{j,\text{rep}}=\mathbf{0})\right]\geq 1-\xi_{ij}^{a};\ \forall(i,j)\in\mathcal{G}_a \tag{5.8a}$$

$$E_{\boldsymbol{\theta}}\left[\Pr(x_j=0|x_i=1)\right]\geq 1-\xi_{ij}^{r},;\ \forall(i,j)\in\mathcal{G}_r \tag{5.8b}$$

$$E_{\boldsymbol{\theta}}\left[\Pr(x_j=1|x_i=1,x_k=0,\boldsymbol{x}_{j,\text{rep}}=\mathbf{0})\right]\geq 1-\xi_{ijk}^{ca};\ \forall(i,j,k)\in\mathcal{G}_{ca} \tag{5.8c}$$

$$E_{\boldsymbol{\theta}}\left[\Pr(x_j=0|x_i=1,x_k=0)\right]\geq 1-\xi_{ijk}^{cr};\ \forall(i,j,k)\in\mathcal{G}_{cr} \tag{5.8d}$$

$$E_{\boldsymbol{\theta}}\left[H_{\boldsymbol{\theta}}[x_i|R_{x_i}]\right]\leq \xi_i^{reg};\ x_i\in C \tag{5.8e}$$

In the following subsections, we consider 3 constructive methods to select prior probabilities compatible with the available prior information. The first two methods are traditionally introduce for construct least-informative priors. We adopt these methods, and modify them.

5.2.1 Regularized Maximum-Entropy Priors

The principle of maximum-entropy was first stated in statistical mechanics almost 55 years ago by Jaynes in [55] as an inference method [105]. This is used for the probability construction of the different (random) states (in the state space) that can be taken, i.e., microscopic states of the system. In statistical mechanics the state functions are random, due to the randomness in the states, and only some mean values of these state functions can be measured [106]. In this way, the maximum entropy probability is the one whose (information) entropy is maximized subject to these mean values: It leaves us with the greatest uncertainty given the constraint in order to prevent adding spurious information. Mathematically speaking, inserting $g_0(\boldsymbol{\theta}) = -\ln\pi(\boldsymbol{\theta})$ in (5.6) with $\boldsymbol{\lambda}=\mathbf{0}$ (i.e., no slack variables) and some predetermined vector $\boldsymbol{\xi}$ leads to the primitive maximum entropy setting (refer

to Appendix D.2 for details). Incorporating the prior knowledge, we extend the notion of maximum entropy probability into the Bayesian setting with slack variables as in (5.6), where the objective function is given by

$$f_0(\boldsymbol{\alpha}, \boldsymbol{\xi}) = -(1 - \boldsymbol{\lambda}^T \mathbf{1}) H[\boldsymbol{\theta}] + \boldsymbol{\lambda}^T \boldsymbol{L}(\boldsymbol{\xi}). \tag{5.9}$$

Inferring from equation (5.9), the RMEP objective function makes a balance between the negative entropy and the knowledge obtained from signaling pathways.

5.2.2 Regularized Maximal Data Information Priors

The maximal data information prior (MDIP) is introduced by Zellner, et. al. [107]. Zellner's choice of objective function is a criterion for prior probability construction to remain "maximally committed to the data" [103]. Adopting the original method into the new framework, the MDIP is the one with

$$g_0(\boldsymbol{\theta}) = -[\ln \pi(\boldsymbol{\theta}) + H[p(x|\boldsymbol{\theta})]],$$

in which $p(x|\boldsymbol{\theta})$ is the likelihood of x when it is parameterized by $\boldsymbol{\theta}$. Taking the expectation with respect to $\boldsymbol{\theta}$, we obtain

$$E_{\boldsymbol{\theta}}[g_0(\boldsymbol{\theta})] = H[\boldsymbol{\theta}] - E_{\boldsymbol{\theta}}[H[p(x|\boldsymbol{\theta})]].$$

In the MDIP, "data" does not mean any actual observation, rather it is used and then marginalized by finding the entropy (refer to Appendix D.2 for details). Similar to the RMEP, the regularized extension of MDIP (RMDIP) is the solution to the

optimization problem in (5.6) in which

$$f_0(\boldsymbol{\alpha}, \boldsymbol{\xi}) = -(1 - \boldsymbol{\lambda}^T \mathbf{1}) \Big[H[\boldsymbol{\theta}] - E_{\boldsymbol{\theta}}[H[f(x|\boldsymbol{\theta})]] \Big] + \boldsymbol{\lambda}^T \boldsymbol{L}(\boldsymbol{\xi}). \tag{5.10}$$

Going from (5.9) to (5.10), one can see that the entropy is subtracted by the prior-average of the entropy of the likelihood of data.

5.2.3 Regularized Expected Mean Log-Likelihood Priors

The general framework for the regularized expected mean log-likelihood priors (REMLP) is detailed in Section 4. The main difference between regularized expected mean log-likelihood prior (REMLP) with its preceding methods, is the way it takes the observations into account [41]. Prior to introducing the REML, all the prior constructing methods were maximally ignorant to the observations (measurements). However, the REML optimization problem searches for the priors which are designed to "remain committed to some part of the sample data" through the expected mean-log-likelihood function while satisfying the constraints imposed by the pathways. The expectation of the log-likelihood is taken with respect to the prior, to marginalize the dependency of the mean-log-likelihood to the actual feature-label distribution parameters and map it to the hyperparameter space. Henceforth, for notational ease, we drop the index y.

To this end, we first split the given sample, $\boldsymbol{u}$, into two parts: $\boldsymbol{u}^{prior}$ and $\boldsymbol{u}^{train}$, with $|\boldsymbol{u}| = |\boldsymbol{u}^{prior}| + |\boldsymbol{u}^{train}|$, where the former is used for prior construction. Here, we restate the REMLP in the general regularized framework given in equation (5.6). The REML prior (REMPL) is found by solving the optimization problem in (5.6) when the objective function is given by

$$f_0(\boldsymbol{\alpha}, \boldsymbol{\xi}) = -(1 - \boldsymbol{\lambda}^T \mathbf{1}) E_{\boldsymbol{\theta}}[\ell_{n_p}(\boldsymbol{\theta}; \boldsymbol{u}^{prior})] + \boldsymbol{\lambda}^T \boldsymbol{L}(\boldsymbol{\xi}). \tag{5.11}$$

where $\ell_{n_p}(\boldsymbol{\theta}; \boldsymbol{u}^{prior})$ is the log-likelihood function of the samples $\boldsymbol{u}^{prior}$. In [41] it has been shown that the variable $\ell_{n_p}(\boldsymbol{\theta}; \boldsymbol{u}^{prior})$ can be interpreted as a measure of "similarity" between the true model and the one governed by the parameter $\boldsymbol{\theta}$. This is similar to the Akaike's information criterion for model selection [89].

5.3 Derivation of Optimization Frameworks for Dirichlet Priors

In this section, we revisit all the aforementioned adopted prior constructing methods, when the feasible space is limited to the family of Dirichlet distributions. The *Dirichlet process* as a prior probability has been extensively studied in the non-parametric Bayesian inference problems [43, 108–111]. In [112], a constructive definition of the Dirichlet measure is discussed. Since in this section we concentrate on the multinomial model, the Dirichlet process prior can be reduced to Dirichlet distribution. In order to be consistent with the standard notation, we denote the Dirichlet parameter by $\boldsymbol{\alpha}$. Then, the Dirichlet distribution, denoted by $\boldsymbol{p} \sim \mathcal{D}ir(\boldsymbol{\alpha})$, is defined as

$$\pi(\boldsymbol{p}|\boldsymbol{\alpha}) = \frac{\Gamma(\sum_{k=1}^{b} \alpha_k)}{\prod_{k=1}^{b} \Gamma(\alpha_k)} \prod_{k=1}^{b} p_k^{\alpha_k - 1}. \tag{5.12}$$

Define $\alpha_0 = \sum_{k=1}^{b} \alpha_k$, which is interpreted as a measure of the strength of the prior knowledge [108]. This parameter can be chosen by the practitioner to represent the strength of his conviction, independent of his opinion about the "shape" of the distribution, $\boldsymbol{\alpha}$, [43]. Finally, we define the feasible region, Π, for a given $\alpha_0 = \sum_{k=1}^{b} \alpha_k$, as follows

$$\Pi = \{\mathcal{D}ir(\boldsymbol{\alpha}) : \boldsymbol{\alpha} \in \mathcal{S}_{b-1}^{\alpha_0}\}$$

where $\mathcal{S}_{b-1}^{\alpha_0}$ is the $\alpha_0 - (b-1)$-dimension simplex, i.e. $0 \leq \sum_{k=1}^{b-1} \alpha_k \leq \alpha_0$.

Now, we state two main lemmas which are the basics for the transformation of the pathways knowledge to the hyperparameter space (the proofs are left in the

appendices). The function $\psi(x) : \mathbb{R}_+ \to \mathbb{R}$, is the digamma function defined as $\psi(x) = \frac{d}{dx} \ln \Gamma(x)$.

Lemma 4. *Suppose that* $\boldsymbol{p} = (p_1, p_2, \ldots, p_b) \sim \mathcal{D}ir(\alpha_1, \alpha_2, \ldots, \alpha_b)$, *with* $\sum_{k=1}^{b} \alpha_k = \alpha_0$. *Then, for any (nonempty) disjoint subsets A and B* ($A \cap B = \emptyset$) *of the set* $\mathcal{X} = \{1, 2, ..., b\}$, *the first moment of the random fraction* $\frac{\sum_{i \in A} p_i}{\sum_{i \in A} p_i + \sum_{j \in B} p_j}$ *is scale invariance, and is given by*

$$E_{\boldsymbol{p}}[\frac{\sum_{i \in A} p_i}{\sum_{i \in A} p_i + \sum_{j \in B} p_j}] = \frac{\sum_{i \in A} \alpha_i}{\sum_{i \in A} \alpha_i + \sum_{j \in B} \alpha_j}. \tag{5.13}$$

Furthermore, the variance of the fraction above is given by

$$\mathrm{Var}_{\boldsymbol{p}}[\frac{\sum_{i \in A} p_i}{\sum_{i \in A} p_i + \sum_{j \in B} p_j}] = \frac{\sum_{i \in A} \alpha_i \sum_{i \in B} \alpha_i}{(\sum_{i \in A} \alpha_i + \sum_{j \in B} \alpha_j)^2 (\sum_{i \in A} \alpha_i + \sum_{j \in B} \alpha_j + 1)}. \tag{5.14}$$

Proof. Proof is given in Appendix D. Q.E.D.

Corollary 3. *Suppose that* $\boldsymbol{p} = (p_1, p_2, \ldots, p_b) \sim \mathcal{D}ir(\alpha_1, \alpha_2, \ldots, \alpha_b)$, *with* $\sum_{k=1}^{b} \alpha_k = \alpha_0$. *Then, for any (nonempty) disjoint subsets A and B* ($A \cap B = \emptyset$) *of the set* $\mathcal{X} = \{1, 2, ..., b\}$, *as* $\alpha_i \to \infty;\ \forall i = 1, \ldots, b$, *with* $\lim_{\alpha_i \to \infty \alpha_0 \to \infty} \frac{\alpha_i}{\alpha_0} = \overline{\alpha}_i$, *we have*

$$\frac{\sum_{i \in A} p_i}{\sum_{i \in A} p_i + \sum_{j \in B} p_j} \xrightarrow{p} \frac{\sum_{i \in A} \overline{\alpha}_i}{\sum_{i \in A} \overline{\alpha}_i + \sum_{j \in B} \overline{\alpha}_j} \tag{5.15}$$

Proof. An immediate consequence of Lemma 4 is that as $\alpha_i \to \infty;\ \forall i = 1, \ldots, b$, with $\lim_{\alpha_i \to \infty \alpha_0 \to \infty} \frac{\alpha_i}{\alpha_0} = \overline{\alpha}_i$ the variance of the random variable $\frac{\sum_{i \in A} p_i}{\sum_{i \in A} p_i + \sum_{j \in B} p_j}$ goes to zero. Therefore, using the Chebyshev's inequality, the convergence in probability is readily resulted. Q.E.D.

Corollary 3 could be seen as a justification of the effectiveness of the constraint

on the Dirichlet parameter for large values of α_0. Now, we state the second lemma dealing with the conditional entropy.

Lemma 5. *Suppose that the binary-valued random vector* $(u_1, u_2, \ldots, u_b)$ *is distributed by a multinomial distribution* $\mathcal{M}ult(p_1, p_2, ..., p_b; 1)$, *whereas the vector* $\boldsymbol{p} = (p_1, p_2, \ldots, p_b)$ *is itself distributed by a Dirichlet distribution as* $\boldsymbol{p} \sim \mathcal{D}ir(\alpha_1, \alpha_2, \ldots, \alpha_b)$, *with* $\sum_{k=1}^{b} \alpha_k = \alpha_0$. *Moreover, for any arbitrary subsets* $A_0, A_1, \ldots, A_M$ *of the set* $\mathcal{X} = \{1, 2, \ldots, b\}$, *define* $Z_{A_i} = \sum_{j \in A_i} u_j$. *Then, we have*

$$E_{\boldsymbol{p}}[H[Z_{A_0}|Z_{A_1}, \ldots, Z_{A_M}]] = \frac{1}{\alpha_0} \sum_{k=1}^{2^M} \sum_{y=0}^{1} \Big[\psi(\sum_{i \in B_k^{0,y}} \alpha_i + \sum_{i \in B_k^{0,1-y}} \alpha_i + 1) - \psi(\sum_{i \in B_k^{0,y}} \alpha_i + 1)\Big] \times \sum_{i \in B_k^{0,y}} \alpha_i, \tag{5.16}$$

in which we have

$$B_k^{0,y} = \bigcap_{i=1}^{M} (A_i \mathbb{I}_{y_i^k} \cup A_i^c \mathbb{I}_{1-y_i^k}) \cap (A_0 \mathbb{I}_{1-y} \cup A_0^c \mathbb{I}_y), \ (y_1^k, y_2^k, \ldots, y_M^k) \in \{0,1\}^M,$$

with the convention of $A\mathbb{I}_0 = \emptyset$ *and* $A\mathbb{I}_1 = A$.

Proof. Proof is given in Appendix D. Q.E.D.

Using two known facts about the digamma function, being $\psi(x+1) = \psi(x) + \frac{1}{x}$; $\forall x \in \mathbb{R}_+$ and $\psi(x) \approx \ln x$, for large values of α_0, the expression in equation (5.16) can be approximated as follows

$$E_{\boldsymbol{p}}[H[Z_{A_0}|Z_{A_1}, \ldots, Z_{A_M}]] \approx \frac{1}{\alpha_0} \sum_{k=1}^{2^M} \sum_{y=0}^{1} \Big[\ln \frac{\sum_{i \in B_k^{0,y}} \alpha_i + \sum_{i \in B_k^{0,1-y}} \alpha_i}{\sum_{i \in B_k^{0,y}} \alpha_i}\Big] \times \sum_{i \in B_k^{0,y}} \alpha_i, \tag{5.17}$$

which is the corresponding expression for the conditional entropy of the binary-valued random variable $(u_1, u_2, \ldots, u_b) \sim \mathcal{M}ult(\frac{\alpha_1}{\alpha_0}, \frac{\alpha_2}{\alpha_0}, \ldots, \frac{\alpha_b}{\alpha_0}; 1)$.

5.3.1 From Pathways to Constraints on Dirichlet Parameter

In this subsection, using the main properties in Lemmas 4-5, we map the connections in the biological pathways, detailed in Table 3.2, to the constraints on the Dirichlet parameter vector. We make connections between a gene and a subset of states in the steady-state behavior. In order to make analogy, one can see that having a single gene being on or off, i.e. $x_i = 0$ or $x_i = 1$ respectively, corresponds to a partition of the states, $\mathcal{X} = \{1, \ldots, b\}$. For example, considering our case of interest (binary valued variables), half of states correspond to $x_i = 0$ and the other half corresponds to $x_i = 1$.

To ease the notation, the portion of the state space $\mathcal{X}$ for which $(x_i = k_1, x_j = k_2)$ and $(x_i \neq k_1, x_j = k_2)$, for any $k_1, k_2 \in \{0, 1\}$, are denoted by $\mathcal{X}^{i,j}_{k_1,k_2}$ and $\mathcal{X}^{i,j}_{k_1^c,k_2}$, respectively. Similarly, the portion of the state space $\mathcal{X}$ for which we have $R_x = \boldsymbol{b}$ and $R_x \neq \boldsymbol{b}$, for any $\boldsymbol{b} \in \{0, 1\}^{|R_x|}$, is denoted by $\mathcal{X}^{R_x}_{\boldsymbol{b}}$ and $\mathcal{X}^{R_x}_{\boldsymbol{b}^c}$, respectively. Furthermore, for a vector $\boldsymbol{\alpha}$ indexed by $\mathcal{X}$, we denote the variable, indicating the summation of its entities in $\mathcal{X}^{i,j}_{k_1^c,k_2}$ by

$$\overline{\alpha}^{i,j}_{b_1,b_2} = \sum_{k \in \mathcal{X}^{i,j}_{b_1,b_2}} \alpha_k. \tag{5.18}$$

The notation above, is extended below for the cases where there are more than two fixed genes. Here, we only derive the equivalent constraints on the APS, and the regulatory sets. Other constraints can be derived similarly and are summarized in Table 5.1. Considering an APS, we expand the conditional probability as follows

$$
\begin{aligned}
&E_{\boldsymbol{p}}[\Pr(x_j = 1 | x_i = 1, \mathbf{x}_{\text{rep},j} = \mathbf{0})] \\
&= E_{\boldsymbol{p}}\Big[\frac{\Pr(x_i = 1, x_j = 1, \mathbf{x}_{\text{rep},j} = \mathbf{0})}{\Pr(x_i = 1, x_j = 1, \mathbf{x}_{\text{rep},j} = \mathbf{0}) + \Pr(x_i = 1, x_j = 0, \mathbf{x}_{\text{rep},j} = \mathbf{0})}\Big] \\
&= E_{\boldsymbol{p}}\Big[\frac{\sum_{k \in \mathcal{X}_{1,1,\mathbf{0}}^{i,j,\text{rep}_j}} p_k}{\sum_{k \in \mathcal{X}_{1,1,\mathbf{0}}^{i,j,\text{rep}_j}} p_k + \sum_{k \in \mathcal{X}_{1,0,\mathbf{0}}^{i,j,\text{rep}_j}} p_k}\Big] = \frac{\overline{\alpha}_{1,1,\mathbf{0}}^{i,j,\text{rep}_j}}{\overline{\alpha}_{1,1,\mathbf{0}}^{i,j,\text{rep}_j} + \overline{\alpha}_{1,0,\mathbf{0}}^{i,j,\text{rep}_j}}
\end{aligned}
\tag{5.19}
$$

where we use from the notation introduced above. The last equality in (5.19) results from applying Lemma 4 (equation (5.13)) with the fact that the two sets $\mathcal{X}_{1,1,\mathbf{0}}^{i,j,\text{rep}_j}$ and $\mathcal{X}_{1,0,\mathbf{0}}^{i,j,\text{rep}_j}$ are disjoint subsets of $\mathcal{X}$.

In order to find a closed form expression, as a function of the Dirichlet parameter, for the conditional entropy, here we make an analogy with the our problem with the assumptions in Lemma 5. Considering the conditional entropy for gene x_i given its regulatory set R_{x_i}, we want to simplify $E_{\boldsymbol{p}}[H[x_i|R_{x_i}]]$. For sake of simplicity in presentation, we show the regulatory set element-wise as $R_{x_i} = (x_{i_i}, x_{i_2}, \ldots, x_{i_M})$ where $M = |R_{x_i}|$. Switching to the bin-wise representations of the variables $x_i; i = 1, \ldots, N$, by using $u_1, \ldots, u_b$, we have the following following equivalency in distribution:

$$
x_i \overset{p}{\equiv} \sum_{k \in \mathcal{X}_1^i} u_k,
$$

where $(u_1, u_2, \ldots, u_b) \sim \mathcal{M}ult(p_1, p_2, \ldots, p_b; 1)$. And, hence, the conditional entropy above can be rewritten as follows

$$
H[x_i | x_{i_i}, x_{i_2}, \ldots, x_{i_M}] = H[\sum_{k \in \mathcal{X}_1^i} u_k | \sum_{k \in \mathcal{X}_1^{i1}} u_k, \sum_{k \in \mathcal{X}_1^{i2}} u_k, \ldots, \sum_{k \in \mathcal{X}_1^{iM}} u_k].
$$

Using the result of Lemma 5, we obtain

$$E_{\boldsymbol{p}}[H[x_i|R_{x_i}]] = \sum_{b=0}^{1} \sum_{\boldsymbol{b}\in\{0,1\}^{|R_{x_i}|}} \frac{\overline{\alpha}_{b,\boldsymbol{b}}^{x_i,R_{x_i}}}{\alpha_0} \Big[\psi(\overline{\alpha}_{\boldsymbol{b}}^{R_{x_i}}+1) - \psi(\overline{\alpha}_{b,\boldsymbol{b}}^{x_i,R_{x_i}}+1)\Big].$$

General constraint	Constraint on the Dirichlet parameter
(5.8a)	$\frac{\overline{\alpha}_{1,1,\mathbf{0}}^{i,j,\mathrm{rep}_j}}{\overline{\alpha}_{1,1,\mathbf{0}}^{i,j,\mathrm{rep}_j}+\overline{\alpha}_{1,0,\mathbf{0}}^{i,j,\mathrm{rep}_j}} \geq 1-\xi_{ij}^{a}$
(5.8b)	$\frac{\overline{\alpha}_{1,0}^{i,j}}{\overline{\alpha}_{1,1}^{i,j}+\overline{\alpha}_{1,0}^{i,j}} \geq 1-\xi_{ij}^{r}$
(5.8c)	$\frac{\overline{\alpha}_{1,1,0,\mathbf{0}}^{i,j,k,\mathrm{rep}_j}}{\overline{\alpha}_{1,1,0,\mathbf{0}}^{i,j,k,\mathrm{rep}_j}+\overline{\alpha}_{1,0,0,\mathbf{0}}^{i,j,k,\mathrm{rep}_j}} \geq 1-\xi_{ijk}^{ca}$
(5.8d)	$\frac{\overline{\alpha}_{1,0,0}^{i,j,k}}{\overline{\alpha}_{1,1,0}^{i,j,k}+\overline{\alpha}_{1,0,0}^{i,j,k}} \geq 1-\xi_{ijk}^{cr}$
(5.8e)	$\sum_b \sum_{\boldsymbol{b}} \frac{\overline{\alpha}_{b,\mathbf{b}}^{x_i,R_{x_i}}}{\alpha_0} \Big[\psi(\overline{\alpha}_{\mathbf{b}}^{R_{x_i}}+1) - \psi(\overline{\alpha}_{b,\mathbf{b}}^{x_i,R_{x_i}}+1)\Big] \leq \xi_i^{reg}$

Table 5.1: Table of constraints on the Dirichlet parameter corresponding to different interactions or prior information existing in the signaling pathways.

5.3.2 Objective Functions

In this subsection, we derive the closed-form expression of the objective functions for three prior construction methods, REMP, RMDIP, and REMLP as a function of Dirichlet parameter.

5.3.2.1 RMEP Dirichlet

It is known that if $\boldsymbol{p} \sim \mathcal{D}ir(\boldsymbol{\alpha})$, then the entropy $H[\boldsymbol{p}]$ is given by [90]

$$H[\boldsymbol{p}] = \sum_{k=1}^{b} \Big[\ln \Gamma(\alpha_i) - (\alpha_i - 1)\psi(\alpha_i) \Big] + (\alpha_0 - b)\psi(\alpha_0) - \ln \Gamma(\alpha_0), \tag{5.20}$$

where the last two terms do not depend on the individual vector components, instead are determined ahead of hyperparameter estimation. Hence, removing the constant parts (those which are not found through optimization problem), the RMEP for the Dirichlet family of priors solves the following constrained optimization problem

$$\text{RMEP-D}: \quad \begin{aligned} &\min_{\boldsymbol{\alpha}} -(1 - \boldsymbol{\lambda}^T \mathbf{1}) \sum_{k=1}^{b} \Big[\ln \Gamma(\alpha_i) - (\alpha_i - 1)\psi(\alpha_i) \Big] + \boldsymbol{\lambda}^T \boldsymbol{L}(\boldsymbol{\xi}) \\ &\text{Subject to:} \begin{cases} \boldsymbol{\alpha} \succ \mathbf{0};\ \boldsymbol{\xi} \succeq \mathbf{0} \\ \sum_{i=1}^{b} \alpha_i = \alpha_0 \\ \text{eqs (5.8a)} - \text{(5.8e)} \end{cases} . \end{aligned} \tag{5.21}$$

Consider the case that instead of prior probability, the posterior probability would directly be obtained.

5.3.2.2 RMDIP Dirichlet

According to equation (5.10) and employing equation (5.20), after removing the constant terms, for the RMDIP we have

$$\begin{aligned} \sum_{k=1}^{b} \Big[\ln \Gamma(\alpha_i) - (\alpha_i - 1)\psi(\alpha_i) \Big] + E_{\boldsymbol{p}}[\sum_{k=1}^{b} p_i \ln p_i] = \sum_{k=1}^{b} \Big[\ln \Gamma(\alpha_i) - (\alpha_i - 1)\psi(\alpha_i) \Big] \\ + \sum_{k=1}^{b} \frac{\alpha_i}{\alpha_0} \Big[\psi(\alpha_i + 1) - \psi(\alpha_0 + 1) \Big], \end{aligned} \tag{5.22}$$

Hence, the RMDIP optimization problem for the Dirichlet prior (RMDIP-D) after removing the constant parts (assuming that α_0 is known and fixed), is given by

$$\text{RMDIP-D}: \begin{cases} \min\limits_{\boldsymbol{\alpha}} -(1-\boldsymbol{\lambda}^T\mathbf{1})\sum\limits_{k=1}^{b}\Big[\ln\Gamma(\alpha_i)-(\alpha_i-1)\psi(\alpha_i)+\alpha_i\dfrac{\psi(\alpha_i+1)}{\alpha_0}\Big]+\boldsymbol{\lambda}^T\boldsymbol{L}(\boldsymbol{\xi}) \\ \text{Subject to:} \begin{cases} \boldsymbol{\alpha}\succ\mathbf{0};\ \boldsymbol{\xi}\succeq\mathbf{0} \\ \sum_{i=1}^{b}\alpha_i=\alpha_0 \\ \text{eqs (5.8a)} - \text{(5.8e)} \end{cases} \end{cases} . \tag{5.23}$$

5.3.2.3 REMLP Dirichlet

Similar to above, we split the given data points into two parts: one for prior construction and the rest for Bayesian classifier learning. We show the number of sample point used for prior construction simply by n_p.

The mean-log-likelihood function, considering the multinomial distribution, in this case can be written as

$$\ell_{n_p}([p_1,...,p_b]) = \frac{1}{n_p}\sum_{k=1}^{b} u_k^{prior}\log p_k + \log\frac{n_p!}{u_1!...u_b!}. \tag{5.24}$$

Removing the constant parts, using property **P4** in Appendix D, the expected mean-log-likelihood function is given by

$$E_{\boldsymbol{\theta}}[\ell_{n_p}([p_1,...,p_b])] = \frac{1}{n_p}\sum_{k=1}^{b} u_k^{prior}[\psi(\alpha_k)-\psi(\alpha_0)]. \tag{5.25}$$

Denote the prior constructing sample set size by n_p. Then, defining $\hat{p}_k := \frac{1}{n_p}u_k^{prior}$, the REMLP Dirichlet (REMLP-D) prior for a known α_0, is found by solving feature-distribution, whereby the final optimization problem for the REMLP-D is given by

$$\text{REMLP-D}: \quad \begin{cases} \min\limits_{\boldsymbol{\alpha}} -(1-\boldsymbol{\lambda}^T\mathbf{1})\sum\limits_{k=1}^{b}\hat{p}_k\psi(\alpha_k)+\boldsymbol{\lambda}^T\boldsymbol{L}(\boldsymbol{\xi}) \\ \text{Subject to:} \begin{cases} \boldsymbol{\alpha}\succ\mathbf{0};\ \boldsymbol{\xi}\succeq\mathbf{0} \\ \sum_{i=1}^{b}\alpha_i=\alpha_0 \\ \text{eqs (5.8a)} - \text{(5.8e)} \end{cases} \end{cases} . \tag{5.26}$$

5.4 Practical Implications of the Objective-Based Priors

In this section, we study some of the basic practical implications regarding the proposed methodologies. Convexity of optimization problems and regularization parameter selections will be covered.

5.4.1 On the Convexity of the Prior Constructing Optimization Problems

The optimization problems proposed in this section for prior construction are not convex programmings. This is mainly due to the constraints coming from the transformation of the knowledge in the pathways.

5.4.1.1 Convexity of the Objective Functions

So far we have introduced three optimization problems in which the objective functions are different while the constraints are the same which are in accordance with our view of the prior knowledge. We summarize our results regarding the objective functions in the following lemma:

Lemma 6. *Suppose that* Π *contains the Dirichlet family of prior distributions. Then, the corresponding objective functions for RMEP and REML (equations* (5.21) *and* (5.26)*) are all convex provided that* $\boldsymbol{\alpha}$ *and* $\boldsymbol{\xi}$ *belong to a convex set. The objective function for RMDIP, equation* (5.23) *is also convex provided that* $\alpha_0 > 1$ *and that* $\boldsymbol{\alpha}$ *and* $\boldsymbol{\xi}$ *belong to a convex set.*

Proof. Throughout the proof we often use from the fact the digamma function $\psi(x)$ is concave in x. Therefore, the proof for the REML objective function is easy, since the only non-linear part of is is the negative of digamma function, making it convex provided that the feasible region is convex.

Now, we move on to the RMEP function. Ignoring the linear function, we need to show that the term $-\sum_{k=1}^{b}\Big[\ln\Gamma(\alpha_i)-(\alpha_i-1)\psi(\alpha_i)\Big]$ is convex in $\boldsymbol{\alpha}$. Since this term is the summation of a single function on elements of the vector, α_i, it is sufficient to show that each individual summand is convex in its element, i.e. showing that $h(x) := (x-1)\psi(x) - \ln\Gamma(x)$ is convex in x. It is only enough to show that the second derivative of $h(x)$ is positive in its domain. In order to show that we show that the derivative itself is increasing. For the derivative, one may write

$$f' = (x-1)\psi'(x) = x\psi'(x) - \psi'(x).$$

We split the problem into two parts: $x \in (0,1]$ and $x \in (1,\infty)$. In the first case, $x \in (0,1]$, take the second derivative, where we have $f''(x) = \psi'(x) + (x-1)\psi''(x)$. Since $\psi(x)$ is an increasing and concave function, and also $x-1$ is negative, the second derivative is positive for $\forall x \in (0,1]$. Now we consider the second interval: $x \in (1,\infty)$. Expanding the function $\psi'(x)$, we have $\psi'(x) = \frac{1}{x} + \frac{1}{2x^2} + o(\frac{1}{x^3})$, and hence $x\psi'(x) = 1 + \frac{1}{2x} + \frac{1}{6x^2} + o(\frac{1}{x^3})$. Then, the whole term can be rewritten as follows

$$f'(x) = 1 - \frac{1}{2x} - \frac{1}{3x^2} - \frac{1}{6x^3} - \frac{1}{30x^4} + \frac{1}{30x^5} + o(\frac{1}{x^6})$$

In this case, because $x > 1$, the dominant term is $-\frac{1}{2x}$ which is increasing in $x \in (1,\infty)$. Finally, since each summand is convex, the linear combination is convex too.

Now consider the RMDIP function: $-\sum_{k=1}^{b}\Big[\ln\Gamma(\alpha_i)-(\alpha_i-1)\psi(\alpha_i)+\alpha_i\frac{\psi(\alpha_i+1)}{\alpha_0}\Big]$,

where similar to above, we only consider one term: $(x-1)\psi(x) - x\frac{\psi(x+1)}{\alpha_0} - \ln\Gamma(x)$. From the properties of digamma function, we know that $\psi(x+1) = \psi(x) + \frac{1}{x}$, from which, by defining $\beta = \frac{\alpha_0 - 1}{\alpha_0}$, we rewrite the function as follows

$$(x-1)\psi(x) - \frac{1}{\alpha_0}(1 + x\psi(x)) - \ln\Gamma(x) = (\beta x - 1)\psi(x) - \ln\Gamma(x),$$

for which the derivative is given by

$$\psi(x)(\beta x - 1) + \psi(x)(\beta - 1).$$

Since for $\alpha_0 > 1$, $\beta \in (0, 1)$, the second term above, $\psi(x)(\beta - 1)$ is convex. Similar to RMEP function, we can show that the first term is also convex, again due to having $\beta < 1$. The summation of two convex functions leads to another convex function finishing the proof. Q.E.D.

5.4.1.2 Convexity of the Constraints

Since all the three objective functions are convex (with the condition of having $\alpha_0 \geq 1$ for RMDIP), we only need to have convex constraints to have the whole optimization problems convex. Unfortunately, we are unable of showing that the constraints are convex in their current view, requiring us to use existing algorithms for nonconvex optimization problems. A drawback of these methods is that there is no guarantee for them to converge to the global optimum, which there might converge to the local optimums. An effective method to solve nonconvex optimization problems is sequential convex programming approach [93].

5.4.2 Sequential Convex Programming

Sequential convex programming (SCP) solves a non-convex optimization problem by iteratively constructing a convex subproblem, being an approximation of the original problem around the current iterate of the optimization parameter [113,114]. The constructed convex subproblem is solved via existing efficient solvers. Knowing from Lemma 6 that all the objective functions are convex, we would only need to approximate the constraints by their first-order Taylor series expansion. In the following, we consider the problem when the scale parameter in the Dirichlet prior, α_0, is known.

Denoting the aggregated optimization parameter by $\boldsymbol{z} = [\boldsymbol{\alpha},\ \boldsymbol{\xi}]$, the regularized objective-based prior constructing framework in its general form may be written as follows

$$\begin{aligned} &\min_{\boldsymbol{z}} f_0(\boldsymbol{z}), \\ &\text{Subject to: } \begin{cases} z_i > 0;\ i = b+1, \ldots, m+b \\ \sum_{i=b+1}^{b+m} z_i = \alpha_0 \\ f_i(\boldsymbol{z}) \leq 0; i = 1, \ldots, m. \end{cases} \end{aligned} \tag{5.27}$$

Incorporating the constraints, a new aggregated function $f_\mu(\boldsymbol{z}) = f_0(\boldsymbol{z}) + \mu \sum_{i=1}^{m} |f_i(\boldsymbol{z})|^+$ is built, in which the parameter μ is the penalty on the constraints violation. The operator $|x|^+$ incurs a cost only if x is positive. Here, we deal with convex objective functions, i.e. $f_0(\boldsymbol{z})$ is convex. However, the constraint $f_i(\boldsymbol{z});\ i = 1, \ldots, m$ are non-convex, which, in the $k-$th iteration of SCP, are replaced by their first-order approximations around the current point $\boldsymbol{z}^{(k)}$ leading to the approximated function

$$\hat{f}_\mu(\boldsymbol{z}; \boldsymbol{z}^{(k)}) = f_0(\boldsymbol{z}) + \mu \sum_{i=1}^{b+m} |f_i(\boldsymbol{z}^{(k)}) + (\boldsymbol{z} - \boldsymbol{z}^{(k)})^T \nabla f_i(\boldsymbol{z}^{(k)})|^+. \tag{5.28}$$

Then, any existing convex programming solver (e.g. interior-point method) can be employed to solve the following constrained convex program:

$$\text{CP}^{(k)}: \quad \begin{aligned} &\min_{\boldsymbol{z}} \hat{f}_\mu(\boldsymbol{z}; \boldsymbol{z}^{(k)}), \\ &\text{Subject to:} \begin{cases} z_i > 0; \ i = b+1, \ldots, m+b \\ \sum_{i=b+1}^{b+m} z_i = \alpha_0 \\ \boldsymbol{z} \in \tau^{(k)}, \end{cases} \end{aligned} \tag{5.29}$$

where $\tau^{(k)}$ is the trust region in which the approximation is valid to some extent. In our implementation, we use $\tau^{(k)} = \{\boldsymbol{z} \in \mathbb{R}^{b+m} || \boldsymbol{z} - \boldsymbol{z}^{(k)}| < \rho^{(k)}\}$ whose size is controlled by $\rho^{(k)}$.

The algorithm is illustrated in details in Algorithm 3, where three loops are implemented: the outermost loop aims at satisfying the constraints by tuning the parameter μ. The two inner loops iteratively update the approximations and the trust region size, until some stopping criteria are met. The trust region size is controlled by the parameter γ in Algorithm 3, where depending on the accuracy of the approximated model, the trust region is either expended or shrunken for the next iteration. The accuracy is evaluated via function decrease in the model to that of the true function.

Knowing the computational limitations, two stopping criteria, in the form of number of iterations, are also added for inner loops. Moreover, the outermost loop stops the algorithm whenever the sum of constraints violations falls below some predetermined threshold, denoted by δ in the algorithm. Any convex programming solver can be used for the approximated subproblems. In this work, we use "interior-point" method. Specially, we use `MATLAB fmincon` function for non-linear constrained optimization, with the upper bound of 3000 function evaluations.

Algorithm 3 Sequential convex programming

Input Parameters:
$\boldsymbol{z}_0$: initial solution
μ_0: initial penalty coefficient (default: 2)
$\mu_{\max}$: maximum penalty coefficient (default: 200)
t: penalty increment factor (default: 1.3)
ρ_0: initial trust region size (default: α_0– Dirichlet scale parameter)
ρ_{thresh}: trust region size threshold (default: 0.01)
γ ($\in (0,1)$): model to true improvement ratio (default: 0.7)
β_- ($\in (0,1)$): trust region shrinkage coefficient (default: 0.5)
β_+ ($\in (1,\infty)$): trust region expansion coefficient (default: 1.1)
δ : threshold for constraints satisfaction (default: 0.001)
Output: $\boldsymbol{z}^* = [\boldsymbol{\alpha}^*,\ \boldsymbol{\xi}^*]$
Initialize: $k = 0,\ \boldsymbol{z}^{(k)} = \boldsymbol{z}_0,\ \rho^{(k)} = \rho_0,\ \mu = \mu_0$
while $\mu < \mu_{\max}$ && $\sum_{i=1}^{m} |f_i(\boldsymbol{z}^{(k)})|^+ < \delta$ **do**
  construct the function: $f_\mu(\boldsymbol{z}) = f_0(\boldsymbol{z}) + \mu \sum_{i=1}^{|\boldsymbol{\xi}|} |f_i(\boldsymbol{z})|^+$
  while $\text{count}_{\text{out}} < \text{count}_{\text{out}}^{\max}$ **do**
    construct the convex function as in equation (5.28) around $\boldsymbol{z}^{(k)}$
    while $\text{count}_{\text{trust}} < \text{count}_{\text{trust}}^{\max}$ **do**
      $(\tilde{\boldsymbol{z}}, \tilde{f}) \leftarrow$ solve $\text{CP}^{(k)}$ using $\boldsymbol{z}^{(k)}$ and $\rho^{(k)}$
      if $f(\tilde{\boldsymbol{z}}) - f(\boldsymbol{z}^{(k)}) \geq \gamma(\tilde{f} - \hat{f}(\boldsymbol{z}^{(k)}))$ **then**
        $\rho^{(k+1)} = \beta_+ \rho^{(k)}$
        $\boldsymbol{z}^{(k+1)} = \tilde{\boldsymbol{z}}$
        break;
      else
        $\rho^{(k+1)} = \beta_- \rho^{(k)}$
      end if
      $\text{count}_{\text{trust}} \leftarrow \text{count}_{\text{trust}} + 1$
    end while
    $\text{count}_{\text{trust}} \leftarrow 0$
    $\text{count}_{\text{out}} \leftarrow \text{count}_{\text{out}} + 1$
    if $\rho^{(k)} < \rho_{\text{thresh}}$ **then**
      break;
    end if
  end while
  $\mu \leftarrow t\mu$, $k \leftarrow k+1$, $\text{count}_{\text{out}} \leftarrow 0$, $\rho^{(k)} \leftarrow \rho_0$
end while
return $\boldsymbol{z}^{(k)}$

5.4.3 Regularization Parameter Selection

In order to choose a regularization parameter we take a heuristic approach similar to that of [41]. The parameter α_0 represents the spread of the prior, larger α_0 meaning

that the prior is more centered about the scale vector. Thus, α_0 can be viewed as the total amount of information in the prior.

We start this section with REMLP method, whereby the regularization parameter aims at making a balance between two sources of information; (1) data through expected likelihood, and (2) slackness variables controlling the conditional entropy. λ^{reg} and λ^{fun}, respectively govern the relative importance of the regulatory information and functional information in the pathways to the total information. We can view the total information, as represented by α_0, as being a "sum" of the amount of data used to form the prior and a proportion of α_0 relating to the importance of the slackness variables. Under this heuristic $\alpha_0 = n_p + (\lambda^{reg} + \lambda^{fun})\alpha_0$, so that $\lambda^{reg} + \lambda^{fun} = \frac{\alpha_0 - n_p}{\alpha_0}$. In this dissertation, we heuristically use $\lambda^{reg} = 2\lambda^{fun}$.

We can also view α_0 as a sum of the data used to form the prior and the amount of data, n_{pw}, that is "equivalent" to the pathway knowledge (recognizing that this "equivalence" is purely a heuristic notion). This leads to $\alpha_0 = n_p + n_{pw}$. Inserting this expression into the expression for λ^{reg} and λ^{fun} yields

$$\begin{aligned} \lambda^{fun} &= 2/3\frac{n_{pw}}{n_p + n_{pw}} \\ \lambda^{reg} &= 1/3\frac{n_{pw}}{n_p + n_{pw}} \end{aligned} \tag{5.30}$$

We are left with defining n_{pw}. In the simulations we let $n_{pw} = mb$, where b as so far is the number of bins. In contrast to the situation in [41], here we have a little sensitivity to the choice of m. We have shown the results for $m = 1$ and 2 for fair comparison.

Since there is no data utilized in the RMEP and RMDIP process, the corresponding regularization parameters cannot be found using the heuristics above. Instead, we need to use more sophisticated approaches. One possible approach to take is

cross-validation, where the the data are split into training the classifier and the rest is used for performance evaluation, being misclassification rate. Nonetheless, in this thesis, we simply use $\lambda^{reg} = 0.6$ and $\lambda^{fun} = 0.3$.

Reflecting on the preceding heuristics we see that we are confronted with a standard problem in pattern recognition, how to regularize two conflicting factors. One thinks of the problem of adding a complexity term when dealing with model selection. We take the usual approach of applying some heuristics and then demonstrating the benefit of the regularization via simulation. Moreover, although the heuristic approach in this section is used in our simulations, we also develop a method based on cross-validation described in details in the Supplementary Materials. Due to computational limitations, the cross-validation based method is not implemented here. However, in practice, where choosing a regularization parameter is a one-time task, one could use the proposed cross-validation method.

5.5 Numerical Experiments

In this section, we examine the performance of objective-based prior construction methods on a set of pathways, playing important role in different contexts: a simplified pathways involved p53 gene consisting of an extra-cellular dna − dsb signal, and a feedback interaction between ATM and p53 − Mdm2 − Wip1. Here, we focus on Boolean modeling of gene/protein values introduced by Stuart kauffman [115]. The Boolean network (BN) model of the problem yields a deterministic representation of the system's dynamics. Despite of its strength, lying in its simplicity, the BN was then further extended to embrace finer factors into the model where BN with perturbation (BNp) and probabilistic BN (PBN) were introduced in [69]. In a BNp, a probabilistic factor is also considered to capture "latent variables," influencing the dynamics, outside of the network under study. In this model, a perturbation

probability, denoted by p_{per}, influences the transitioning between states.

5.5.1 Pathways Involved TP53

5.5.1.1 Pathways Description

To characterize the dynamical behavior of normal pathways involving the p53 gene illustrated in Figure 5.1.

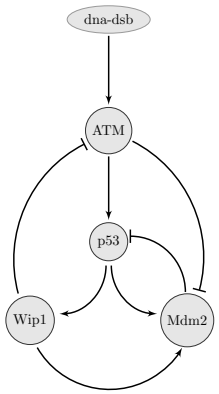


Figure 5.1: A set of simplified pathways involved the TP53 gene, redrawn from [5].

We evaluated the performance of the proposed algorithm based on a family of BNs constructed from pathways that involve the p53 gene. Tumor suppressor gene p53 has been extensively studied and it is known to be involved in various well-known biological pathways. It has been observed that p53 is mutated in 30-50% of common human cancers [116]. In fact, in the presence of DNA damage, a mutant p53 may lead to the emergence of abnormal cells. Figure 5.1 shows the pathways that involve the tumor suppressor gene p53 [117]. These pathways operate in different ways depending on the context, determined by the presence (or absence) of a DNA damage event that results in DNA double-strand breaks. The uncertainty in these

pathways, in the context of Boolean network modeling, has been previously studied in [1, 5].

The logic relationships between the genes, shown in the pathways in Figure 5.1, are given in Table 5.2.

Table 5.2: Boolean functions of the pathways shown in Figure 5.1 [1].

Gene	Node name	Regulating function	Regulatory set
dna − dsb	v_1	Extracellular signal	
ATM	v_2	$\overline{v_4} \wedge (v_2 \vee v_1)$	$\{v_3, v_5\}$
P53	v_3	$\overline{v_5} \wedge (v_2 \vee v_4)$	$\{v_4, v_5\}$
Wip1	v_4	v_3	$\{v_2, v_5\}$
Mdm2	v_5	$\overline{v_2} \wedge (v_3 \vee v_4)$	$\{v_3\}$

Sequencing data of 138 patients with glioblastoma, provided by TCGA, showed that 32% and 12% of the patients suffered from the alteration in the $p53$ and $Mdm2$ genes, respectively. Also among 316 patients with serous ovarian cancer, 96% suffered from the mutation of $p53$ [5]. A similar study has revealed that about 26% of 216 patients with sarcoma have amplified $Mdm2$.

5.5.1.2 Classification Problem

The initial feature vector making the pathways in Figure 5.1 is composed of the following elements

$$[\text{dna} - \text{dsb}, \text{ATM}, \text{P53}, \text{Wip1}, \text{Mdm2}]. \tag{5.31}$$

In order to construct the true model for evaluating the performance of the proposed method, we use the logical dependencies given in [1]. Mathematically speaking, the value of each gene in the pathways is determined either according to logical in Table 5.2 or by flipping its current value chosen by a Bernoulli variable whose probability of success is p_{per}. As the perturbation probability increases, the final stochastic system becomes more different from the Boolean network constructed directly from Table 5.2.

Considering the perturbation probability leads to having a transition probability matrix (TPM) instead of an adjacency matrix. If p_{per} takes nonzero value, then the resulting Markov chain becomes irreducible, and hence, possesses a steady state vector. Using the Boolean functions in Table 5.2, we build the true steady-state distributions $\boldsymbol{p}^{y,true}$; $y \in \{0, 1\}$ for two classes. The full functional regulations are summarized in Table 5.2 yielding one single deterministic Boolean network, representing the relationships governing the cell in its normal functioning. Then, the feature distribution can be computed using a small perturbation probability, e.g. $p_{pert} = 0.05$. It leads to an ergodic and irreducible Markov chain which possesses a steady-state distribution, corresponding to the cell normal functioning. Now, based on the studies reported above in existing cancer data, we consider a classification problem as follows: No mutation vs permanently deactivation of p53.

Owing to the nature of dna − dsb which is not in fact a measurable component, we marginalize the probability mass functions above to concentrate only on the following components

$$\boldsymbol{x} = [\text{ATM}, \text{P53}, \text{Wip1}, \text{Mdm2}]. \tag{5.32}$$

Consequently, two class-conditional probability mass functions $\boldsymbol{p}^{0,true}$ and $\boldsymbol{p}^{1,true}$ are fixed, which will be further utilized for taking samples.

5.5.2 Results: Expected True Errors

We find the steady-state distributions on the pathways described above when the perturbation probability is set to $p_{\text{per}} = 0.05$. Increasing the perturbation probability would lead the network to act more randomly rather following the regulatory relationships. The sampling from the true distributions are assumed to be random where the class sizes are randomly chosen based on a predetermined true class prior probability, $c = 0.5$. In the simulations, we compare 4 cases: (1) Histogram classification rule, and 3 OBC obtained with different priors: (2) REMP for both classes assuming that the connections led to $p53$ do not exist, and hence the prior knowledge is less, (3) RMDIP for both classes $y \in \{0, 1\}$, and (4) REMLP for both classes $y \in \{0, 1\}$.

Out of 5 priors addressed in this section, 4 of them are data-independent, whereas the REMLP method utilizes some part of the data for prior construction which will be considered later. These data-independent prior constructing methods can be used before any observation. Moreover, while the uniform and Jeffreys' priors are prior knowledge independent, the proposed RMEP and RMDIP methods are capable of incorporating pathway knowledge. In our simulations study, we only examine the performance in the cases where the prior knowledge, in the form of signaling pathways, is available only for one of the class, being $y = 0$ throughout this section. Therefore, in what follows we denote the given pathways by $\mathcal{G}$ without any class-dependent indexing. Once the priors are constructed, they will be further utilized in a Bayesian framework to construct optimal Bayesian classifiers whenever labeled observations are acquired.

Fixing the true class-conditional bin probabilities, denoted by $\boldsymbol{p}^{0,true}$ and $\boldsymbol{p}^{1,true}$ respectively for class zero and one, we take n samples from the bins, composing

$\boldsymbol{u}^{y}_{n_y,i}$; $y \in \{0,1\}$ in the i-th iteration, whereas, $i = 1, ..., M$ and $n_0 + n_1 = n$, where n_0 and n_1 are random. The samples $\boldsymbol{u}^{0}_{n_0,i}$ and $\boldsymbol{u}^{1}_{n_1,i}$ are then used for updating the prior (constructing the posterior), and classifier design via equation (5.2).

Now consider the REML method. Since we are assuming the pathways are only available for the class $y = 0$, the sample vector $\boldsymbol{u}^{1}_{n_1,i}$ will be unchanged and used for updating the priors. In order to implement the REMLP method, the points in $\boldsymbol{u}^{0}_{n_0,i}$ must be randomly split into two parts, denoted by $\boldsymbol{u}^{0,prior}_{n^p_0,i}$ and $\boldsymbol{u}^{0,train}_{n^t_0,i}$, respectively for prior construction and updating the prior. We also keep the equality $n^0_p + n^0_t = n^0$. Using $\mathcal{G}$ and $\boldsymbol{u}^{0,prior}_{n^p_0,i}$, prior probability $\pi^0_{\text{REML},i}$ is constructed using the optimization problem in (5.26). Similar to the other cases, the rest of the points with $y = 0$, $\boldsymbol{u}^{0,train}_{n^t_0,i}$ will be used for training the OBC classifier using (5.2). Denote the constructed classifier by $\psi_{\boldsymbol{u}_{n,i}}$, where we denote the overall sample vector by $\boldsymbol{u}_{n,i} = \boldsymbol{u}^{0}_{n_0,i} + \boldsymbol{u}^{1}_{n_1,i}$. The true error of this designed classifier is obtained by

$$\epsilon(\psi_{\boldsymbol{u}_{n,i}}|\boldsymbol{u}_{n,i}) = \sum_{k=1}^{b} c p_k^{0,true} I_{\psi_{\boldsymbol{u}_{n,i}}(k)=1} + (1-c) p_k^{1,true} I_{\psi_{\boldsymbol{u}_{n,i}}(k)=0}. \tag{5.33}$$

We then compute the expected true error (with respect to the sample points $\boldsymbol{u}$, via Monte-Carlo (MC) simulations. Assuming an $M-$iteration MC procedure, the expected true error will be given by

$$E[\epsilon(\psi_{\boldsymbol{u}_{n,i}})] = \frac{1}{M} \sum_{i=1}^{M} \epsilon(\psi_{\boldsymbol{u}_{n,i}}|\boldsymbol{u}_{n,i}).$$

The overall strategy, for a fixed sample size, class prior probability, and set of pathways for class zero (n, c, and $\mathcal{G}$, respectively), repeated through MC simulations, is implemented step-wise as follows:

1. Fix true bin probabilities for two classes: $\boldsymbol{p}^{0,true}$, $\boldsymbol{p}^{1,true}$.

2. Determine n_0 and n_1 randomly with the given n and c, i.e. $n_0 = \lceil cn \rceil$, $n_1 = n - n_0$. In this work, we consider two values for the $c = 0.4$ and $c = 0.6$.

3. Take observations from the multinomial distributions: $\boldsymbol{u}_{n_0}^0$ and $\boldsymbol{u}_{n_1}^1$.

4. Using n_0 and n_1, the class prior probability is estimated using maximum-likelihood: $\hat{c} = \frac{n_0}{n}$.

5. (Only for REMLP) Randomly choose n_p points from $\boldsymbol{u}_{n_0}^0$ for prior construction, i.e., $\boldsymbol{u}_{n_0^p,i}^{0,prior}$ and the rest $\boldsymbol{u}_{n_0^t,i}^{0,train}$, for training. Then, use $\boldsymbol{u}_{n_0^p,i}^{0,prior}$ and $\mathcal{G}$ to construct the REMLP.

6. Use (5.2) to optimally combine the priors and the training data, ($\boldsymbol{u}_{n_0}$, $\boldsymbol{u}_{n_1}$ and $\boldsymbol{u}_{n_0^t,i}^{0,train}$ $\boldsymbol{u}_{n_1}$, respectively for non-REMLP and REMLP methods) and design the classifier.

7. Compute the true error associated with the designed classifier using equation (5.33).

The results are shown in Figure 5.2. Figures 5.2a and 5.2b show the expected true errors for the case with $m = 2$, $c = 0.6$ and $n = 20$ and $n = 30$, respectively. Similarly, the results for the case with $c = 0.4$ are shown in Figures 5.2c-5.2d.

From Figure 5.2, one can see that the OBC classifier designed using the prior constructing methods significantly outperforms those of designed using histogram rule. Nonetheless, one should realize that the OBC classifiers are not designed to be optimal for any specific model, rather they tend to perform optimally over an assumed model for the uncertainty one would have about the true model. The curves associated with the REMLP Dirichlet prior shows a convex property, meaning that at some point, increasing prior constructing points does not lead to a better performing

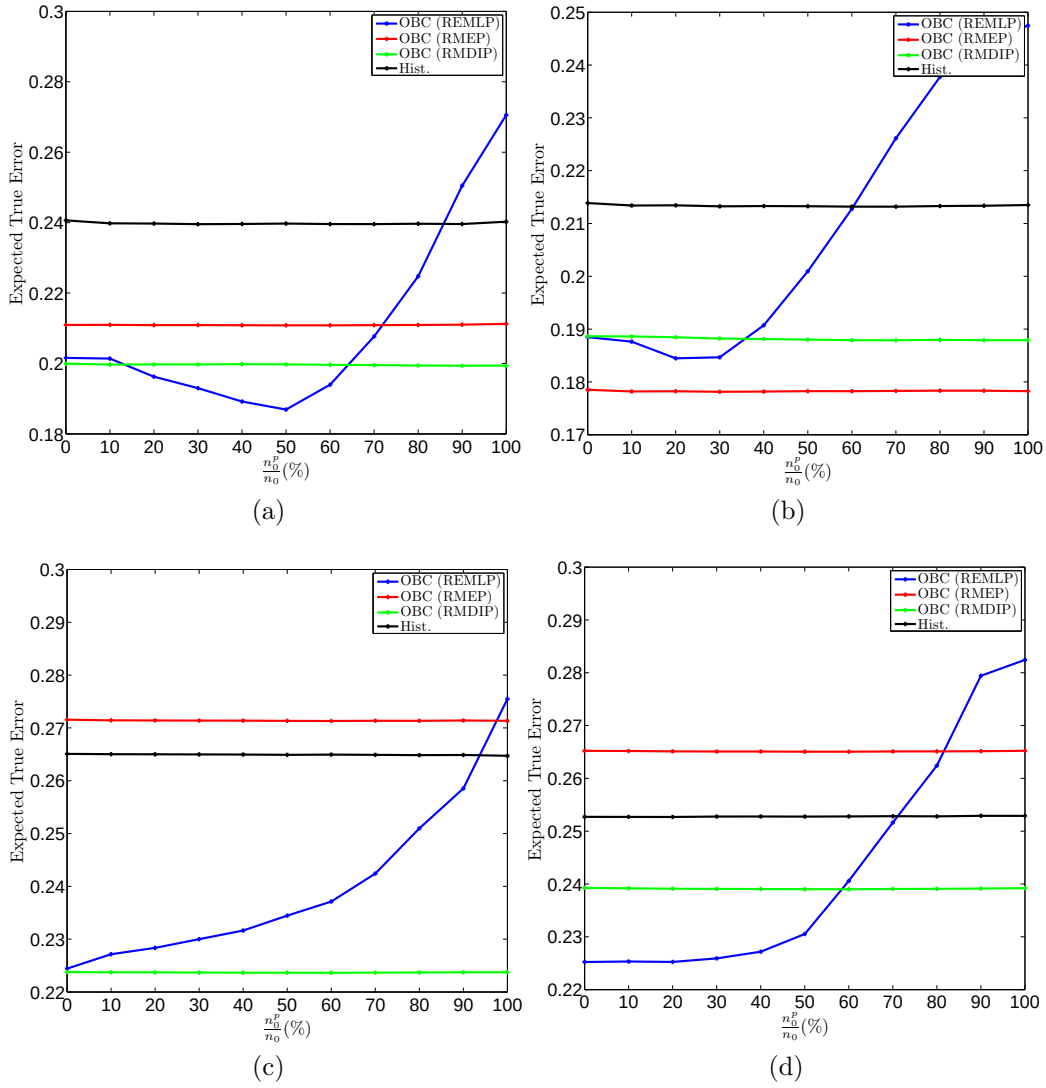


Figure 5.2: The expected true error of different prior constructing methods used for classifying between the normal cell functioning and permanently down-regulated the tumor suppressor gene $p53$.

classifier. Hence, it suggests that there should be some optimal number of points to be invested for prior construction.

5.6 Discussion

In one hand, the pattern recognition community has confronted with small-sample problems in bioinformatics. On the other hand, it is widely known that there is a huge amount of information in different contexts which can significantly improve the task of phenotype classification. One of the important sources of information in biology is signaling pathways where interactions between substantial components are illustrated in graphs. Edges in in these graphs convey the type of influence, being either activating or suppressing. In this section, we propose a unified regularized framework to translate this prior knowledge via mathematical and engineering tools to prior probabilities. These prior probabilities can be utilized in different Bayesian settings to design operators performing optimally with respect to our uncertainty about the problem. We have proposed three methods, regularized maximum entropy, regularized maximal data information, and regularized expected mean log-likelihood to construct the priors. Among these, the first two are extensions of two of the widely used methods for prior construction. The difference between these and the last method is in their view towards the problem: while the REMP and RMDIP are ignorant to the observed data, the REML method takes advantage of some portion of the data to shape and guide the prior. Through simulations on both synthetic and real examples, we showed that quantifying existing prior information from a Bayesian perspective, and then utilizing it in designing an OBC classifier, significantly improves the classification accuracy. An important factor in any regularized framework is the role played by the regularization parameters. We used a heuristic method to select the regularization parameters. Nonetheless, we have developed the

foundations for selecting regularization parameter in a cross-validation procedure in which the classification accuracy is measured for choosing an appropriate parameter.

6. CONCLUSION

In this dissertation, we have presented a comprehensive study of the use of prior knowledge in the design of classifiers to be applied on genomic data. Considering the phenotype classification, and more specifically cancer biology, the most common type of trusted prior knowledge is in the form of signaling pathways. In that regard, we develop mathematical and statistical tools for designing enhanced classifiers, which significantly outperform traditional data-driven classification methods. The main contribution of this dissertation lies in the fact that, for the first time in protemic/genomic classification, the right source of prior knowledge is mathematically transferred to testable information, and further utilized to designing classifiers.

First, we define a problem which arises directly for biological classification using pathway information. We develop a novel classifier design paradigm that allows us to design enhanced classifiers by incorporating available prior knowledge of the process generating the observation data. As shown in our simulations, such knowledge can significantly improve the performance of the designed classifier, especially, when the sample size is small. Having laid the theoretical groundwork for enhancing steady-state classifier design via the use of prior process knowledge, our plan is to apply the methodology to developing better biomedical classifiers in the presence of partial knowledge of the underlying genetic regulatory network. More generally, given the ubiquity of large feature sets and relatively small sample sizes now common in many disciplines, including medicine, material science, environmental science, and transportation, there will no doubt be an increasing number of methods proposed for using prior knowledge in classifier design. We believe it is important to provide analytic performance characterization of the classifiers on standard models, as we

have done in this work, so that their behavior can be understood.

Secondly, we propose a Bayesian framework to transform the knowledge in the signaling pathways to the mathematically sound expressions, which can be directly interpreted by the corresponding paramterization in the Bayesian setting. All the common sub-structures in the signaling pathways, i.e. activation, inhibition, and regulatory set connections, are quantified from a Bayesian perspective, where the "prior-expected structural-specific behaviors" of the models inside the uncertainty class satisfy the pathway knowledge.

Thirdly, in this dissertation, we show that purely data-driven approaches to classifier design with small samples tend to produce poor classifiers whose errors cannot be reliably estimated. The importance of small-sample classification is highlighted by its prevalence in genomic/proteomic applications. In general, prior (probability) selection is one of the main challenges when one is dealing with any Bayesian framework. Conjugate priors are of great interest because of their convenient properties for deriving the posterior probabilities; however, there is no general rigorous mathematical machinery from which to estimate the hyperparameters. The proposed optimization framework is different from its predecessors in the sense that the REML prior relies on sample data and incorporates these data with "pure prior knowledge" to obtain a prior probability. The objective function is based on the notion of a model selection criterion, where the criterion is marginalized using the prior probability. The performance of the designed prior is examined by evaluating the true error of the optimal Bayesian classifier designed via the posterior. Moreover, since we use some initial data in prior construction and thereafter use new data to construct a posterior distribution in the Bayesian framework, one might consider this a "hybrid" approach. But from the perspective of our goal, integration of pathway knowledge and data, this characterization is semantic.

Finally, we generalize the problem of prior construction using biological pathways in a general objective-based framework, where the objective function reflects some information measuring functional regularized with the knowledge extracted from pathways. The REML method is shown to be a special case of this general methodology, and two other classical approaches are also adapted in our proposed framework. Having that developed, one can now readily transform any set of biological signaling pathways to a set of constraints in the hyper-parameter space. As a final comment on this part, to the best of our knowledge for the first time, in this dissertation, rigorous mathematical methodology is developed by which biological pathways can be readily mapped to a Dirichlet prior distribution, which can be later used for designing optimal Bayesian classifier, or optimal Bayesian control policy.

In conclusion, let us note that the overarching goal is to use prior knowledge, in the form of biological pathways, to assist in the design of genomic classifiers. In that regard, model-based classification rules are strongly capable of embracing the available prior information which comes from prior knowledge. The fundamental conclusion is that pathway knowledge and data are integrated to produce classifiers that are superior to those based on data alone, and this is done via optimization procedures which are mostly based on incorporating the information obtained by transforming the pathway knowledge into constraints on the feature-label distribution.

REFERENCES

[1] R. K. Layek, A. Datta, and E. R. Dougherty. From biological pathways to regulatory networks. *Mol. BioSyst.*, vol. 7, no. 3, pp. 843–851, 2011.

[2] J. M. Knight, A. Datta, and E. R. Dougherty. Generating stochastic gene regulatory networks consistent with pathway information and steady-state behavior. *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 6, pp. 1701–1710, 2012.

[3] J. Hua, C. Sima, M. Cypert, G. C. Gooden, S. Shack, et al. Tracking transcriptional activities with high-content epifluorescent imaging. *Journal of Biomedical Optics*, vol. 17, no. 4, pp. 0460081–04600815, 2012.

[4] R. Layek, A. Datta, M. Bittner, and E. R. Dougherty. Cancer therapy design based on pathway logic. *Bioinformatics*, vol. 27, no. 4, pp. 548–555, 2011.

[5] M. Shahrokh Esfahani, B. J. Yoon, and E. R. Dougherty. Probabilistic reconstruction of the tumor progression process in gene regulatory networks in the presence of uncertainty. *BMC Bioinformatics*, vol. 12, no. Suppl 10:S9, 2011.

[6] T. Breslin, M. Krogh, C. Peterson, and C. Troein. Signal transduction pathway profiling of individual tumor samples. *BMC Bioinformatics*, vol. 6, no. 1, pp. 163, 2005.

[7] J P. Svensson, L. JA Stalpers, R. EE Esveldt-van Lange, N. AP Franken, J. Haveman, at al. Analysis of gene expression using gene sets discriminates cancer patients with and without late radiation toxicity. *PLoS Medicine*, vol. 3, no. 10, pp. e422, 2006.

[8] E. Lee, H-Y. Chuang, J-W. Kim, T. Ideker, and D. Lee. Inferring pathway activity toward precise disease classification. *PLoS Computational Biology*, vol. 4, no. 11, pp. e1000217, 2008.

[9] J. Su, B-J. Yoon, and E. R Dougherty. Accurate and reliable cancer classification based on probabilistic inference of pathway activity. *PLoS One*, vol. 4, no. 12, pp. e8161, 2009.

[10] H-S. Eo, J. Y. Heo, Y. Choi, Y. Hwang, and H-S. Choi. A pathway-based classification of breast cancer integrating data on differentially expressed genes, copy number variations and microrna target genes. *Molecules and Cells*, vol. 34, no. 4, pp. 393–398, 2012.

[11] Z. Wen, Z-P. Liu, Y. Yan, G. Piao, Z. Liu, et al. Identifying responsive modules by mathematical programming: An application to budding yeast cell cycle. *PloS One*, vol. 7, no. 7, pp. e41854, 2012.

[12] S. Kim, M. Kon, and C. DeLisi. Pathway-based classification of cancer subtypes. *Biology Direct*, vol. 7, no. 1, pp. 1–22, 2012.

[13] N. Khunlertgit and B.-J. Yoon. Identification of robust pathway markers for cancer through rank-based pathway activity inference. *Advances in Bioinformatics*, vol. 2013, Article ID 618461, 2013.

[14] M. L Gatza, J. E Lucas, W. T Barry, J. W. Kim, Q. Wang, et al. A pathway-based classification of human breast cancer. *Proceedings of the National Academy of Sciences*, vol. 107, no. 15, pp. 6994–6999, 2010.

[15] J. R Nevins. Pathway-based classification of lung cancer: a strategy to guide therapeutic selection. in *Proceedings of the American Thoracic Society*, vol. 8, no. 2, pp. 180, 2011.

[16] Z. Guo, T. Zhang, X. Li, Q Wang, J. Xu, et al. Towards precise classification of cancers based on robust gene functional expression profiles. *BMC Bioinformatics*, vol. 6, no. 1, pp. 58, 2005.

[17] J. Tomfohr, J. Lu, and T.B. Kepler. Pathway level analysis of gene expression using singular value decomposition. *BMC Bioinformatics*, vol. 6, no. 1, pp. 225, 2005.

[18] H. Y. Chuang, E. Lee, Y. T. Liu, D. Lee, and T. Ideker. Network-based classification of breast cancer metastasis. *Molecular Systems Biology*, vol. 3, no. 1, 2007.

[19] J. Su, B-J. Yoon, and E. R. Dougherty. Identification of diagnostic subnetwork markers for cancer in human protein-protein interaction network. *BMC Bioinformatics*, vol. 11, no. Suppl 6, S8, 2010.

[20] Z. Wen, Z-P. Liu, Z. Liu, Y. Zhang, and Luonan Chen. An integrated approach to identify causal network modules of complex diseases with application to colorectal cancer. *Journal of the American Medical Informatics Association*, vol. 20, no. 4, pp. 659–667, 2013.

[21] A. N. Tikhanov. On the stability of inverse problems. *C. R. (Doklady) Aca. Sci. URSS (N. S.)*, vol. 39, no. 5, pp. 195-198, 1943.

[22] D. L Phillips. A technique for the numerical solution of certain integral equations of the first kind. *Journal of the ACM (JACM)*, vol. 9, no. 1, pp. 84–97, 1962.

[23] A. E. Hoerl and R. W. Kennard. Ridge regression: applications to nonorthogonal problems. *Technometrics*, vol. 12, no. 1, pp. 69–82, 1970.

[24] L. Breiman. Heuristics of instability and stabilization in model selection. *The Annals of Statistics*, vol. 24, no. 6, pp. 2350–2383, 1996.

[25] D. L. Donoho and I. M. Johnstone. Minimax estimation via wavelet shrinkage. *The Annals of Statistics*, vol. 26, no. 3, pp. 879–921, 1998.

[26] J. Fan and R. Li. Statistical challenges with high dimensionality: Feature selection in knowledge discovery. *ArXiv Preprint, Math/0602133*, 2006.

[27] Y. C Eldar. Generalized sure for exponential families: Applications to regularization. *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 471–481, 2009.

[28] A. Wiesel. Unified framework to regularized covariance estimation in scaled gaussian models. *IEEE Transactions on Signal Processing*, vol. 60, no. 1, pp. 29–38, 2012.

[29] A. Aubry, A. De Maio, L. Pallotta, and A. Farina. Maximum likelihood estimation of a structured covariance matrix with a condition number constraint. *IEEE Transactions on Signal Processing*, vol. 60, no. 6, pp. 3004–3021, 2012.

[30] J-H. Won, J. Lim, S-J. Kim, and B. Rajaratnam. Condition-number-regularized covariance estimation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 73, no. 3, pp. 427–450, 2013.

[31] P. J. Bickel, B. Li, A. B. Tsybakov, S. A van de Geer, B. Yu, et al. Regularization in Statistics. *Test*, vol, 15, no. 2, pp. 271–344, 2006.

[32] A. Zollanvari, U. M. Braga-Neto, and E. R. Dougherty. Analytic study of performance of error estimators for linear discriminant analysis. *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4238–4255, 2011.

[33] A. Zollanvari, U. M. Braga-Neto, and E. R. Dougherty. Joint sampling distribution between actual and estimated classification errors for linear discriminant analysis. *IEEE Transactions on Information Theory*, vol. 56, no. 2, pp. 784–804, 2010.

[34] V. Berikov and A. Litvinenko. The influence of prior knowledge on the expected performance of a classifier. *Pattern Recognition Letters*, vol. 24, no. 15, pp. 2537–2548, 2003.

[35] U. Braga-Neto and E. R. Dougherty. Exact performance of error estimators for discrete classifiers. *Pattern Recognition*, vol. 38, no. 11, pp. 1799–1814, 2005.

[36] A. Zollanvari, U. Braga-Neto, and E. R. Dougherty. Exact representation of the second-order moments for resubstitution and leave-one-out error estimation for linear discriminant analysis in the univariate heteroskedastic Gaussian model. *Pattern Recognition*, vol. 45, no. 2, pp. 908–917, 2012.

[37] L. A. Dalton and E. R. Dougherty. Bayesian minimum mean-square error estimation for classification error part i: Definition and the Bayesian MMSE error estimator for discrete classification. *IEEE Transactions on Signal Processing*, vol. 59, no. 1, pp. 115–129, 2011.

[38] L. A. Dalton and E. R. Dougherty. Bayesian minimum mean-square error estimation for classification error part ii: Linear classification of Gaussian models. *IEEE Transactions on Signal Processing*, vol. 59, no. 1, pp. 130–144, 2011.

[39] L. A. Dalton and E. R. Dougherty. Optimal classifiers with minimum expected error within a Bayesian framework–part I: Discrete and Gaussian models. *Pattern Recognition*, vol. 46, no. 5, pp. 1301–1314, 2013.

[40] L. A. Dalton and E. R. Dougherty. Optimal classifiers with minimum expected error within a Bayesian framework–part II: Properties and performance analysis. *Pattern Recognition*, vol. 46, no. 5, pp. 1288–1300, 2013.

[41] M. Shahrokh Esfahani and E. R. Dougherty. Incorporation of biological pathway knowledge in the construction of priors for optimal Bayesian classification. *IEEE/ACM Transactions on Computational Biology and Bioinformatics,* doi:10.1109/TCBB.2013.143, in press, 2014.

[42] J. Hua, Z. Xiong, J. Lowey, E. Suh, and E. R. Dougherty. Optimal number of features as a function of sample size for various classification rules. *Bioinformatics*, vol. 21, no. 8, pp. 1509–1515, 2005.

[43] C. E. Antoniak. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics*, vol. 2, no. 6, pp. 1152–1174, 1974.

[44] J. O. Berger and J. M. Bernardo. On the development of reference priors. *Bayesian Statistics*, vol. 4, no. 4, pp. 35–60, 1992.

[45] H. Jeffreys. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 186, no. 1007, pp. 453–461, 1946.

[46] R Kashyap. Prior probability and uncertainty. *IEEE Transactions on Information Theory*, vol. 17, no. 6, pp. 641–650, 1971.

[47] J. M. Bernardo. Reference posterior distributions for Bayesian inference. *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 41, no. 2, pp. 113–147, 1979.

[48] J. Rissanen. A universal prior for integers and estimation by minimum description length. *The Annals of Statistics*, vol. 11, no. 2, pp. 416–431, 1983.

[49] J. C. Spall and S. D. Hill. Least-informative Bayesian prior distributions for finite samples based on information theory. *IEEE Transactions on Automatic Control*, vol. 35, no. 5, pp. 580–583, 1990.

[50] C. C. Rodríguez, E. Priors, and C. C. Rodr'iguez. Entropic priors. PA, Pennsylvania State University: CiteSeerX, 1991.

[51] A. Zellner. *Past and recent results on maximal data information priors*. Chicago, IL: University of Chicago, Graduate School of Business, Department of Economics, 1995.

[52] R. E. Kass and L. Wasserman. The selection of prior distributions by formal rules. *Journal of the American Statistical Association*, vol. 91, no. 435, pp. 1343–1370, 1996.

[53] J. O. Berger, J. M. Bernardo, and D. Sun. Objective priors for discrete parameter spaces. *Journal of the American Statistical Association*, vol. 107, no. 498, pp. 636–648, 2012.

[54] A. Caticha and R. Preuss. Maximum entropy and Bayesian data analysis: Entropic prior distributions. *Physical Review E*, vol. 70, no. 4, pp. 046127, 2004.

[55] E. T. Jaynes. Information theory and statistical mechanics. *Physical Review*, vol. 106, no. 4, pp. 620, 1957.

[56] E. T. Jaynes. Prior probabilities. *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 3, pp. 227–241, 1968.

[57] A. Zellner. Models, prior information, and Bayesian analysis. *Journal of Econometrics*, vol. 75, no. 1, pp. 51–68, 1996.

[58] P. J. Huber. A robust version of the probability ratio test. *The Annals of Mathematical Statistics*, vol. 36, no. 6, pp. 1753–1758, 1965.

[59] K. S. Vastola and H. V. Poor. An analysis of the effects of spectral uncertainty on wiener filtering. *Automatica*, vol. 19, no. 3, pp. 289–293, 1983.

[60] S. A. Kassam and H. V. Poor. Robust techniques for signal processing: A survey. *Proceedings of the IEEE*, vol. 73, no. 3, pp. 433–481, 1985.

[61] A. M. Grigoryan and E. R. Dougherty. Bayesian robust optimal linear filters. *Signal Processing*, vol. 81, no. 12, pp. 2503–2521, 2001.

[62] J. Martín, C. J Pérez, and P. Müller. Bayesian robustness for decision making problems: Applications in medical contexts. *International Journal of Approximate Reasoning*, vol. 50, no. 2, pp. 315–323, 2009.

[63] J. Unnikrishnan, V. V. Veeravalli, and S. P. Meyn. Minimax robust quickest change detection. *IEEE Transactions on Information Theory*, vol. 57, no. 3, pp. 1604–1614, 2011.

[64] K. Vastola and H. Poor. On the p-point uncertainty class (corresp.). *IEEE Transactions on Information Theory*, vol. 30, no. 2, pp. 374–376, 1984.

[65] G. Matz and F. Hlawatsch. Minimax robust nonstationary signal estimation based on a p-point uncertainty model. *Journal of the Franklin Institute*, vol. 337, no. 4, pp. 403–419, 2000.

[66] P. J. Brown and P. W. K. Rundell. Kernel estimates for categorical data. *Technometrics*, vol. 27, no. 3, pp. 293–299, 1985.

[67] C.M. Stein. Estimation of the mean of a multivariate normal distribution. *The Annals of Statistics*, vol. 9, no. 6, pp. 1135–1151, 1981.

[68] G. Zipf. The psycho-biology of language: An introduction to dynamic philology. Oxford, England: Houghton, Miffin, 1935.

[69] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang. Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, vol. 18, no. 2, pp. 261–274, 2002.

[70] N. A. Smith and R. W. Tromble. Sampling uniformly from the unit simplex. Technical Report. Baltimore, MD: Johns Hopkins University, 2004.

[71] M. Karin. Nf-κb as a critical link between inflammation and cancer. *Cold Spring Harbor Perspectives in Biology*, vol. 1, no. 5, pp. a000141, 2009.

[72] P. J. Delves and I. M. Roitt. *Roitt's essential immunology.* Hoboken, NJ: Wiley-Blackwell, 2006.

[73] J. M. Knight and E. R. Dougherty. Attractor estimation and model refinement for stochastic regulatory network models. in *Proceedings of IEEE International Workshop on Genomic Signal Processing and Statistics (GENSIPS'2011), San Antonio, TX, December 2011.* IEEE, 2011, pp. 54–55.

[74] J.M. Knight, A. Datta, and E.R. Dougherty. Generating stochastic gene regulatory networks consistent with pathway information and steady-state behavior *IEEE Transaction on Biomedical Engineering*, vol. 59, no. 6, pp. 1701–1710, 1984.

[75] E. R Dougherty, M. Brun, J. M Trent, and M. L Bittner. Conditioning-based modeling of contextual genomic regulation. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 6, no. 2, pp. 310–320, 2009.

[76] A. Abdi, M. B. Tahoori, and E. S. Emamian. Fault diagnosis engineering of digital circuits can identify vulnerable molecules in complex cellular pathways. *Science Signaling*, vol. 1, no. 42, pp. ra10, 2008.

[77] M. Shahrokh Esfahani, J. Knight, A. Zollanvari, B-J. Yoon, and E. R. Dougherty. Classifier design given an uncertainty class of feature distributions via regularized maximum likelihood and the incorporation of biological pathway knowledge in steady-state phenotype classification. *Pattern Recognition*, vol. 46, no. 10, pp. 2783–2797, 2013.

[78] L. A Dalton and E. R Dougherty. Bayesian minimum mean-square error estimation for classification error–part I: Definition and the Bayesian MMSE error estimator for discrete classification. *IEEE Transactions on Signal Processing*, vol. 59, no. 1, pp. 115–129, 2011.

[79] L. A. Dalton and E. R. Dougherty. Bayesian minimum mean-square error estimation for classification error–part ii: the Bayesian MMSE error estimator for linear classification of Gaussian distributions. *IEEE Transactions on Signal Processing*, vol. 59, no. 1, pp. 130–144, 2011.

[80] D. V. Lindley. On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, vol. 27, no. 4, pp. 986-1005, 1956.

[81] B-J. Yoon, X. Qian, and E. R. Dougherty. Quantifying the objective cost of uncertainty in complex dynamical systems. *IEEE Transactions on Signal Processing*, vol. 61, no. 9, pp. 32256–2266, 2013.

[82] N. Friedman, D. Geiger and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, vol. 29, no. 2, pp. 131–163, 1997.

[83] J. Cheng and R. Greiner. Comparing Bayesian network classifiers. In *Proceedings of the Fifteenth conference on Uncertainty in Artificial Intelligence*, Montreal, Quebec: Morgan Kaufmann Publishers Inc., 1999, pp. 101–108.

[84] J. Cheng and R. Greiner. Learning Bayesian belief network classifiers: Algorithms and system. *Advances in Artificial Intelligence*, vol. 2056, pp. 141–151, 2001.

[85] D. Grossman and P. Domingos. Learning Bayesian network classifiers by maximizing conditional likelihood. In *Proceedings of the Twenty-First International Conference on Machine Learning*, New York, NY: ACM Press, 2004, pp. 46–53.

[86] H. Akaike. Information theory and an extension of the maximum likelihood principle. In *International Symposium on Information Theory, 2 nd, Tsahkadsor, Armenian SSR*, pp. 267–281, 1973.

[87] H. Akaike. A Bayesian analysis of the minimum AIC procedure. *Annals of the Institute of Statistical mathematics*, vol. 30, no. 1, pp. 9–14, 1978.

[88] H. Bozdogan. Model selection and Akaike's information criterion (AIC): The general theory and its analytical extensions. *Psychometrika*, vol. 52, no. 3, pp. 345–370, 1987.

[89] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, vol. 19, no. 6, pp. 716–723, 1974.

[90] C. M. Bishop. *Pattern recognition and machine learning*, New York, NY: Springer, 2006, vol. 31.

[91] L. R Haff. An identity for the Wishart distribution with applications. *Journal of Multivariate Analysis*, vol. 9, no. 4, pp. 531–544, 1979.

[92] A. Dembo, T. M. Cover, and J. A. Thomas. Information theoretic inequalities. *IEEE Transactions on Information Theory*, vol. 37, no. 6, pp. 1501–1518, 1991.

[93] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge, UK: Cambridge University Press, 2004.

[94] R. J. Vanderbei and D. F. Shanno. An interior-point algorithm for nonconvex nonlinear programming. *Computational Optimization and Applications*, vol. 13, no. 1, pp. 231–252, 1999.

[95] C. J. Hsieh, M. A. Sustik, I. S. Dhillon, and P. Ravikumar. Sparse inverse covariance matrix estimation using quadratic approximation. *Advances in Neural Information Processing Systems (NIPS)*, 24, pp. 2330–2338 2011.

[96] J. Nocedal and S. J Wright. *Numerical optimization*. New York, NY: Springer Verlag, 1999.

[97] E. R. Dougherty, S. Kim, and Y. Chen. Coefficient of determination in nonlinear signal processing. *Signal Processing*, vol. 80, no. 10, pp. 2219–2235, 2000.

[98] S. Kim, E. R Dougherty, M. L Bittner, Y. Chen, K. Sivakumar, et al. General nonlinear framework for the analysis of gene interaction via multivariate expression arrays. *Journal of Biomedical Optics*, vol. 5, no. 4, pp. 411–424, 2000.

[99] E. R. Dougherty and M. L. Bittner. *Epistemology of the cell: a systems perspective on biological knowledge*. New York, NY: Wiley-IEEE Press, 2011, vol. 34.

[100] X. Gao, Y. Zhang, P. Arrazola, O. Hino, T. Kobayashi, at al. Tsc tumour suppressor proteins antagonize amino-acid–TOR signalling. *Nature Cell Biology*, vol. 4, no. 9, pp. 699–704, 2002.

[101] L. S. Harrington, G. M. Findlay, A. Gray, T. Tolkacheva, S. Wigfield, et al. The TSC1-2 tumor suppressor controls insulin–pi3k signaling via regulation of irs proteins. *The Journal of Cell Biology*, vol. 166, no. 2, pp. 213–223, 2004.

[102] J. Brugarolas, K. Lei, R. L. Hurley, B. D. Manning, J. H. Reiling, et al. Regulation of mTOR function in response to hypoxia by REDD1 and the TSC1/TSC2 tumor suppressor complex. *Genes & Development*, vol. 18, no. 23, pp. 2893–2904, 2004.

[103] N. Ebrahimi, E. Maasoumi, and E. S. Soofi. Measuring informativeness of data by entropy and variance. In *Advances in Econometrics, Income Distribution and Scientific Methodology*, pp. 61–77. New York, NY: Springer, 1999.

[104] A. Antoniadis and J. Fan. Regularization of wavelet approximations. *Journal of the American Statistical Association*, vol. 96, no. 455, pp. 939–967, 2001.

[105] J. Shore and R. W. Johnson. Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy. *IEEE Transactions on Information Theory*, vol. 26, no. 1, pp. 26–37, 1980.

[106] S. Guiasu and A. Shenitzer. The principle of maximum entropy. *The Mathematical Intelligencer*, vol. 7, no. 1, pp. 42–48, 1985.

[107] A. Zellner. *Maximal data information prior distributions*. Amsterdam: North-Holland, 1977.

[108] T. S. Ferguson. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, vol. 1, no. 2, pp. 209–230, 1973.

[109] Thomas S Ferguson. Prior distributions on spaces of probability measures. *The Annals of Statistics*, vol. 2, no. 4, pp. 615–629, 1974.

[110] M. Zhou, H. Chen, J. Paisley, L. Ren, L. Li, et al. Nonparametric Bayesian dictionary learning for analysis of noisy and incomplete images. *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 130–144, 2012.

[111] W. Buntine, L. Du, and P. Nurmi. Bayesian networks on Dirichlet distributed vectors. In *Proceedings of the 5th European Workshop on Probabilistic Graphical Models (PGM-10), 2010*, 2010, pp. 33-40.

[112] J. Sethuraman. A constructive definition of Dirichlet priors. Technical report, Tallahassee, FL: Florida State University, DTIC Document, 1991.

[113] S. Boyd. Sequential convex programming. *Lecture Notes, Stanford University at http://www.stanford.edu/class/ee364b/lectures.html*, 2008.

[114] J. Schulman, A. Lee, I. Awwal, H. Bradlow, and P. Abbeel. Finding locally optimal, collision-free trajectories with sequential convex optimization. *Submitted. Draft at https://sites. google. com/site/rss2013trajopt*, 2013.

[115] S. A Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, vol. 22, no. 3, pp. 437–467, 1969.

[116] R. A. Weinberg. *The biology of cancer*. New York, NY: Garland Science, 2007.

[117] E. Batchelor, A. Loewer, and G. Lahav. The ups and downs of p53: understanding protein dynamics in single cells. *Nature Reviews Cancer*, vol. 9, no. 5, pp. 371–377, 2009.

[118] P. Hall. *The bootstrap and Edgeworth expansion*. New York, NY: Springer Verlag, 1997.

[119] S. Geisser. The predictive sample reuse method with applications. *Journal of the American Statistical Association*, vol. 70, no. 350, no. 320–328, 1975.

[120] G. H. Golub, M. Heath, and G. Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, vol. 21, no. 2, pp. 215–223, 1979.

APPENDIX A

PRELIMINARIES AND PROOFS IN SECTION 2

A.1 Proof of Theorems 1 and 2

In this appendix, we prove Theorems 1 and 2 for $y = 0$. The case $y = 1$ can be handled similarly. Let the inner expectation in (2.14), $\mathrm{E}_{S_n}\Big[\Pr(\psi_{n,\Pi^0,\Pi^1}(X) \neq Y|S_n)\Big]$, be denoted by EXP_1. Then

$$\begin{aligned}\mathrm{EXP}_1 &= \mathrm{E}_{S_n}\left[\sum_k \Pr(X=k,Y=0)\mathrm{I}_{\{\psi_{n,\Pi^0.\Pi^1}=1\}} + \Pr(X=k,Y=1)I_{\{\psi_{n,\Pi^0.\Pi^1}=0\}}\right] \\ &= c_0 \sum_k \left[\pi^0(k)\Pr(\hat{c_1}\tfrac{(1-\lambda_1)u_k^1+\lambda_1\overline{\pi}^1(k)}{(1-\lambda_1)n_1+\lambda_1} \geq \hat{c_0}\tfrac{(1-\lambda_0)u_k^0+\lambda_0\overline{\pi}^0(k)}{(1-\lambda_0)n_0+\lambda_0})\right] \\ &+ c_1 \sum_k \left[\pi^1(k)\Pr(\hat{c_0}\tfrac{(1-\lambda_0)u_k^0+\lambda_0\overline{\pi}^0(k)}{(1-\lambda_0)n_0+\lambda_0} > \hat{c_1}\tfrac{(1-\lambda_1)u_k^1+\lambda_1\overline{\pi}^1(k)}{(1-\lambda_1)n_1+\lambda_1})\right],\end{aligned} \tag{A.1}$$

in which we apply $\hat{c_y} = \frac{n_y}{n}; y = 0, 1$. We denote the average distribution by $\overline{\boldsymbol{\pi}}_y; y = 0, 1$ which can be computed by $\overline{\boldsymbol{\pi}}_y = (1-\varepsilon_y)\boldsymbol{\pi}_{ac}^y + \varepsilon_y\overline{\boldsymbol{\pi}}$, where $\overline{\boldsymbol{\pi}}$ is the average of contaminating distributions. Now, for $y = 0, 1$, define

$$\begin{aligned} g_y &:= (1-\lambda_y)n_y(n_{1-y}(1-\lambda_{1-y}) + \lambda_{1-y}) \\ \alpha_y &:= \tfrac{g_y\lambda_y}{1-\lambda_y} \\ p_y(k) &:= \alpha_y\overline{\pi}_y(k). \end{aligned} \tag{A.2}$$

Equation (A.1) can be written as

$$\begin{aligned}\mathrm{EXP}_1 &= \sum_{k=1}^{b}\Big[\Pr(X=k|Y=0)c_0\Pr(g_1u_k^1 + p_1(k) \geq g_0u_k^0 + p_0(k))\} \\ &+ \Pr(X=k|Y=1)c_1\Pr(g_0u_k^0 + p_0(k) > g_1u_k^1 + p_1(k))\Big],\end{aligned} \tag{A.3}$$

$$
\begin{aligned}
\mathrm{EXP}_1 \;&= \sum_{k=1}^{b} \pi_{ac}^0(k) c_0 \Big[\textstyle\sum_{l_0=0}^{n_0} \{ \sum_{m=\zeta_{k,l_0}^0}^{n_1} (\pi_{ac}^1(k))^m (1-\pi_{ac}^1(k))^{n_1-m} \binom{n_1}{m} \} \\
&\quad (\pi_{ac}^0(k)^{l_0} (1-\pi_{ac}^0(k))^{n_0-l_0} \binom{n_0}{l_0}) \Big] \\
&\quad + \textstyle\sum_{k=1}^{b} \pi_{ac}^1(k) c_1 \Big[\sum_{l_1=0}^{n_1} \{ \sum_{m=\zeta_{k,l_1}^1}^{n_0} (\pi_{ac}^0(k))^m (1-\pi_{ac}^0(k))^{n_0-m} \binom{n_0}{m} \} \\
&\quad \times (\pi_{ac}^1(k)^{l_1} (1-\pi_{ac}^1(k))^{n_1-l_1} \binom{n_1}{l_1}) \Big],
\end{aligned}
\tag{A.4}
$$

where

$$
\begin{aligned}
\underline{\zeta}_{k,l_0}^0 &= \max\{0, \left\lfloor \tfrac{g_0 l_0 + p_0(k) - p_1(k)}{g_1} \right\rfloor + 1\}, \\
\underline{\zeta}_{k,l_1}^1 &= \max\{0, \left\lfloor \tfrac{g_1 l_1 + p_1(k) - p_0(k)}{g_0} \right\rfloor + 1\}.
\end{aligned}
\tag{A.5}
$$

In (A.4), we have two random variables $\underline{\zeta}_{k,l_0}^0$ and $\underline{\zeta}_{k,l_1}^1$ depending on the uncertainty classes Π^0 and Π^1, respectively. We present the distributions of these random variables for the uncertainty class models described in Section 2.1 in the following subsections:

A.1.0.1 $\varepsilon-$Contamination Class

We first show that the contaminating part $\pi(k)$ in (2.10) has a Beta distribution $B(1, b-1)$, where b is the number of states. Suppose that the contaminating distributions come from a uniform distribution on a $(b-1)$- simplex. Thus, as $\Delta x \to 0$,

$$
\begin{aligned}
\Pr(x - \Delta x/2 < \pi(k) < x + \Delta x/2) \;&= \Delta x \frac{\mathrm{Vol}(\mathcal{S}_{b-2}^{1-x})}{\mathrm{Vol}(\mathcal{S}_{b-1})} = \Delta x \frac{\frac{(1-x)^{b-2}}{(b-2)!}}{\frac{1}{(b-1)!}} \\
&= \Delta x (b-1)(1-x)^{b-2}
\end{aligned}
\tag{A.6}
$$

where Vol(.) denotes volume under the specified argument and $\mathcal{S}_{b-1}$ and $\mathcal{S}_{b-2}^{1-x}$ are the unit $(b-1)-$simplex and $(b-2)-$simplex with corners on $1-x$, respectively. (A.6) can be written as a density function according to

$$
f_{\pi(k)}(x) = (b-1)(1-x)^{b-2}, x \in (0,1) \tag{A.7}
$$

which is a Beta distribution with parameters 1 and $b-1$ whose mean and variance are $\frac{1}{b}$ and $\frac{b-1}{b^2(b+1)}$, respectively. Using the *Edgeworth expansion* to approximate the cumulative density function of $\overline{\pi}(k)$, [118], we obtain

$$\Pr(\overline{\pi}(k) < x) = \Phi(z) + R_{|\Pi^0|} \tag{A.8}$$

where $z = \sqrt{|\Pi^0|}\frac{x-\frac{1}{b}}{\sqrt{\frac{b-1}{b^2(b+1)}}}$, and we have

$$R_{|\Pi^0|} := \lim_{w\to\infty} \frac{\sum_{v=1}^{w} r_v(\Pi^0)}{\exp(c|\Pi^0|)}, c > 0. \tag{A.9}$$

In (A.9), according to the Edgeworth expansion, we have

$$r_v(|\Pi^0|) = O(|\Pi^0|^{\frac{v}{2}-1}). \tag{A.10}$$

Considering (A.9) and (A.10), one can conclude that $R_{|\Pi^0|} \to 0$ for large enough uncertainty classes. Therefore, for large uncertainty classes, we will approximately have $\frac{\overline{\pi}(k)-\frac{1}{b}}{\sqrt{\frac{b-1}{b^2(b+1)}}} \sim N(0, \frac{1}{|\Pi^y|})$. Hence, considering the last line of equation (A.2), we get the following result:

$$\begin{aligned} p_0(k) &\sim N(\alpha_0 \left[(1-\varepsilon_0)\pi^0_{ac}(k) + \tfrac{\varepsilon_0}{b}\right], \alpha_0^2\varepsilon_0^2\tfrac{(b-1)}{b^2(b+1)|\Pi^0|}) \\ p_1(k) &\sim N(\alpha_1 \left[(1-\varepsilon_1)\pi^1_{ac}(k) + \tfrac{\varepsilon_1}{b}\right], \alpha_1^2\varepsilon_1^2\tfrac{(b-1)}{b^2(b+1)|\Pi^1|}). \end{aligned} \tag{A.11}$$

Thus, since $p_0(k)$ and $p_1(k)$ are independent random variables, we get

$$\frac{g_0 l_0 + p_0(k) - p_1(k)}{g_1} \sim N(\mu^0_{k,l_0}, \sigma_0^2),$$

where μ^0_{k,l_0}, σ_0^2 are defined in (2.18). It is now straightforward to find the distribution of $\underline{\zeta}^0_{k,l_0}$ (and similarly $\underline{\zeta}^1_{k,l_1}$) using equation (A.5).

A.1.0.2 p−Point Class

From the mapping defined in 2.13, we know that state k belongs to $s^0_{P^0(k)}$ and $s^1_{P^1(k)}$ under labels zero and one, respectively. Considering class Π^0, similar to (A.6), one can show that

$$p_{\pi(k)}(x) = \frac{|s^0_{P^0(k)}| - 1}{\omega^0_{P^0(k)}}(1 - \frac{x}{\omega^0_{P^0(k)}})^{|s^0_{P^0(k)}|-2}, x \in (0, \omega^0_{P^0(k)}). \tag{A.12}$$

which is equivalent to the random variable $\omega^0_{P^0(k)}Y$ with $Y \sim Beta(1, |s^0_{P^0(k)}| - 1)$. Therefore, similar to (A.11), we obtain

$$\begin{aligned} p_0(k) &\sim N(\alpha_0 \tfrac{\omega^0_1}{|s^0_1|}, \alpha_0^2(\omega^0_1)^2 \tfrac{(|s^0_1|-1)}{|s^0_1|^2(|s^0_1|+1)|\Pi^0|}) \\ p_1(k) &\sim N(\alpha_1 \tfrac{\omega^1_1}{|s^1_1|}, \alpha_1^2(\omega^1_1)^2 \tfrac{(|s^1_1|-1)}{|s^1_1|^2(|s^1_1|+1)|\Pi^1|}), \end{aligned}$$

from which we obtain $\frac{g_0 l_0 + p_0(k) - p_1(k)}{g_1} \sim N(\mu^0_{k,l_0}, \sigma_0^2)$, whereas

$$\begin{aligned} \mu^0_{k,l_0} &= \frac{g_0 l_0 + \alpha_0 \frac{\omega^0_{P^0(k)}}{|s^0_{P^0(k)}|} - \alpha_1 \frac{\omega^1_{P^1(k)}}{|s^1_{P^1(k)}|}}{g_1} \\ {\sigma_0}^2 &= \Big[\alpha_0^2(\omega^0_{P^0(k)})^2 \tfrac{(|s^0_{P^0(k)}|-1)}{|s^0_{P^0(k)}|^2(|s^0_{P^0(k)}|+1)|\Pi^0|} + \alpha_1^2(\omega^1_{P^1(k)})^2 \tfrac{(|s^1_{P^1(k)}|-1)}{|s^1_{P^1(k)}|^2(|s^1_{P^1(k)}|+1)|\Pi^1|}\Big]/g_1^2. \end{aligned} \tag{A.13}$$

Now, one can find the distribution of $\underline{\zeta}^0_{k,l_0}$ according to (A.5). The distribution of $\underline{\zeta}^1_{k,l_1}$ can be found similarly. Afterwards, we obtain equation (2.16).

A.2 Proof of Theorem 3

The second-order moment of the true error of the RML classifier can be written as

$$\mathrm{E}(\epsilon^2_{\mathbf{RML}}) = \mathrm{E}_{\Pi^0,\Pi^1}\left[\mathrm{E}_{S_n}\left[\Pr(\psi_{n,\Pi^0,\Pi^1}(X) \neq Y|S_n)\right]^2 |\Pi^0,\Pi^1\right]. \tag{A.14}$$

For simplicity, we drop the subscript of ψ_{n,Π^0,Π^1}, noting that the classifier depends S_n and Π^0, Π^1. The proof has two parts shown in two appendices. First, we take the expectation with respect to the training data, S_n. Later, we will see that the dependency of the second-order moment on the uncertainty classes manifests itself in the indices of the double-summations (found from combinatorial parts). In the next section, then we find the distribution of those indices, knowing that the randomness comes from the uncertainty classes. Let us start the proof by expanding equation (A.14):

$$\begin{aligned}
\mathrm{E}(\epsilon^2_{\mathbf{RML}}) &= \mathrm{E}_{\Pi^0,\Pi^1}\Big[c_0^2\sum_k(\pi^0_{ac}(k))^2\underbrace{\mathrm{E}_{S_n}[\mathrm{I}_{\{\psi(X=k)=1\}}]}_{A^1} + c_1^2\sum_k(\pi^1_{ac}(k))^2\underbrace{\mathrm{E}_{S_n}[\mathrm{I}_{\{\psi(X=k)=0\}}]}_{A^0} \\
&+c_0^2\sum_{k_1\neq k_2}\pi^0_{ac}(k_1)\pi^0_{ac}(k_2)\underbrace{\mathrm{E}_{S_n}[\mathrm{I}_{\{\psi(X=k_1)=1\}}\mathrm{I}_{\{\psi(X=k_2)=1\}}]}_{B^1} \\
&+c_1^2\sum_{k_1\neq k_2}\pi^1_{ac}(k_1)\pi^1_{ac}(k_2)\underbrace{\mathrm{E}_{S_n}[\mathrm{I}_{\{\psi(X=k_1)=0\}}\mathrm{I}_{\{\psi(X=k_2)=0\}}]}_{B^0} \\
&+c_0c_1\sum_{k_1\neq k_2}\pi^0_{ac}(k_1)\pi^1_{ac}(k_2)\underbrace{\mathrm{E}_{S_n}[\mathrm{I}_{\{\psi(X=k_1)=1\}}\mathrm{I}_{\{\psi(X=k_2)=0\}}]}_{C^1} \\
&+c_0c_1\sum_{k_1\neq k_2}\pi^1_{ac}(k_1)\pi^0_{ac}(k_2)\underbrace{\mathrm{E}_{S_n}[\mathrm{I}_{\{\psi(X=k_1)=0\}}\mathrm{I}_{\{\psi(X=k_2)=1\}}]}_{C^0}\Big].
\end{aligned} \tag{A.15}$$

In (A.15), parts A_0 and A_1 can be found similarly as in Appendix 1. In the following, whenever we sum over $t_1^y, t_2^y; y \in \{0,1\}$ we implicitly consider $t_1^y, t_2^y \geq 0$ and $t_1^y + t_2^y \leq$

n_y. Furthermore, for any pair of $(t_1^y, t_2^y) \succeq \mathbf{0}$ with $t_1^y + t_2^y \leq n_y$, we have

$$\Pr(u_{k_1}^y = t_1^y, u_{k_2}^y = t_2^y) = \Pr(trin(n_y, \pi_{ac}^y(k_1), \pi_{ac}^y(k_2)) = (t_1^y, t_2^y)).$$

Hence, for the B^1, we may write

$$\begin{aligned}
B^1 &= \mathrm{E}_{S_n}[\mathrm{I}_{\{\psi(X=k_1)=1\}}\mathrm{I}_{\{\psi(X=k_2)=1\}}] = \Pr(\psi(X=k_1)=1, \psi(X=k_2)=1) \\
&= \Pr(g_1 u_{k_1}^1 + p_1(k_1) \geq g_0 u_i^0 + p_0(i), g_1 u_{k_2}^1 + p_1(k_2) \geq g_0 u_{k_2}^0 + p_0(k_2)) \\
&= \sum_{t_1^0, t_2^0} \Pr(g_1 u_{k_1}^1 + p_1(k_1) \geq g_0 t_1^0 + p_0(k_1), g_1 u_{k_2}^1 + p_1(k_2) \geq g_0 t_2^0 + p_0(k_2)) \\
&\times \Pr(u_{k_1}^0 = t_1^0, u_{k_2}^0 = t_2^0) \\
&= \sum_{t_1^0, t_2^0} \Pr(u_{k_1}^1 \geq \underline{\zeta}_{k_1, t_1^0}^0, u_{k_2}^1 \geq \underline{\zeta}_{k_2, t_2^0}^0) \Pr(u_{k_1}^0 = t_1^0, u_{k_2}^0 = t_2^0) \\
&= \sum_{t_1^0, t_2^0} \left[\sum_{(t_1^1, t_2^1) \succeq (\underline{\zeta}_{k_1, t_1^0}^0, \underline{\zeta}_{k_2, t_2^0}^0)} \Pr(u_{k_1}^1 = t_1^1, u_{k_2}^1 = t_2^1) \Pr(u_{k_1}^0 = t_1^0, u_{k_2}^0 = t_2^0) \right]
\end{aligned} \tag{A.16}$$

Similarly, we can get

$$B^0 = \sum_{t_1^1, t_2^1} \left[\sum_{(t_1^0, t_2^0) \succeq (\underline{\zeta}_{k_1, t_1^1}^1, \underline{\zeta}_{k_2, t_2^1}^1)} \Pr(u_{k_1}^0 = t_1^0, u_{k_2}^0 = t_2^0) \Pr(u_{k_1}^1 = t_1^1, u_{k_2}^1 = t_2^1) \right]. \tag{A.17}$$

Next, we can obtain C^1

$$\begin{aligned}
C^1 &= \mathrm{E}_{S_n}[\mathrm{I}_{\{\psi(X=i)=1\}}\mathrm{I}_{\{\psi(X=j)=0\}}] = \Pr(\psi(X=k_1)=1, \psi(X=k_2)=0) \\
&= \Pr(g_1 u^1_{k_1} + p_1(k_1) \geq g_0 u^0_{k_1} + p_0(k_1), g_1 u^1_{k_2} + p_1(k_2) < g_0 u^0_{k_2} + p_0(j)) \\
&= \sum_{t^0_1, t^0_2} \Pr(g_1 u^1_{k_1} + p_1(k_1) \geq g_0 t^0_1 + p_0(k_1), g_1 u^1_{k_2} + p_1(k_2) < g_0 t^0_2 + p_0(k_2)) \\
&\times \Pr(u^0_{k_1} = t^0_1, u^0_{k_2} = t^0_2) \\
&= \sum_{t^0_1, t^0_2} \Pr(\underline{\zeta}^0_{k_1, t^0_1} \leq u^1_{k_1}, u^1_{k_2} \leq \overline{\zeta}^0_{k_2, t^0_2}) \Pr(u^0_{k_1} = t^0_1, u^0_{k_2} = t^0_2) \\
&= \sum_{t^0_1, t^0_2} \left[\sum_{t^1_1 \geq \underline{\zeta}^0_{k_1, t^0_1}, t^1_2 \leq \overline{\zeta}^0_{k_2, t^0_2}} \Pr(u^1_{k_1} = t^1_1, u^1_{k_2} = t^1_2) \Pr(u^0_{k_1} = t^0_1, u^0_{k_2} = t^1_1) \right],
\end{aligned} \tag{A.18}$$

Similarly, we obtain

$$C^0 = \sum_{t^1_1, t^1_2} \left[\sum_{t^0_1 \geq \underline{\zeta}^1_{k_1, t^1_1}, t^0_2 \leq, \overline{\zeta}^1_{k_2, t^1_2}} \Pr(u^0_{k_1} = t^0_1, u^0_{k_2} = t^0_2) \Pr(u^1_{k_1} = t^1_1, u^1_{k_2} = t^1_0) \right]. \tag{A.19}$$

In (A.18)-(A.19), we have

$$\begin{aligned}
\overline{\zeta}^0_{k, l_0} &= \min\{ \left\lceil \tfrac{g_0 l_0 + p_0(k) - p_1(k)}{g_1} \right\rceil - 1, n_1 \}, \\
\overline{\zeta}^1_{k, l_1} &= \min\{ \left\lceil \tfrac{g_1 l_1 + p_1(k) - p_0(k)}{g_0} \right\rceil - 1, n_0 \}.
\end{aligned} \tag{A.20}$$

In order to take the last expectation in (A.15) with respect to the uncertainty classes, we need to find the joint distribution of $\underline{\zeta}^0_{k_1, t^0_1}$ and $\underline{\zeta}^0_{k_2, t^0_2}$ (similarly for $\underline{\zeta}^1_{k_1, t^1_1}$ and $\underline{\zeta}^1_{k_2, t^1_2}$), and the joint distribution between $\underline{\zeta}^0_{k_1, t^0_1}$ and $\overline{\zeta}^0_{k_2, t^0_2}$ (similarly for $\underline{\zeta}^1_{k_1, t^1_1}$ and $\overline{\zeta}^1_{k_2, t^1_2}$). These distributions are found in A.3.

A.3 Joint Distributions

To find the joint distribution of $(\underline{\zeta}^0_{k_1,t_1^0}, \underline{\zeta}^0_{k_2,t_2^0})$, we need to approximate the joint distribution of $(p_0(k_1), p_0(k_2))$ defined in equation (A.2). We do this by a (zero-order) Edgeworth expansion. Thus, similar to the single variate case in (A.11), for the multivariate case we have $(p_0(k_1), p_0(k_2)) \sim \mathcal{N}(\boldsymbol{\mu}^0_{k_1,k_2}, \boldsymbol{\Sigma}^0_{k_1,k_2})$, whereas we find the parameters for different uncertainty classes in the following subsections.

A.3.1 $\varepsilon-$Contamination Class

From the definition of the joint probability distribution, for $x_1, x_2 > 0, x_1+x_2 \leq 1$, we have

$$\begin{aligned}
\Pr(\pi(k_1) = x_1, \pi(k_2) = x_2) &= \lim_{\Delta x_1 \to 0 \Delta x_2 \to 0} \frac{\Pr(|\pi(k_1)-x_1| < \frac{\Delta x_1}{2}, |\pi(k_2)-x_2| < \frac{\Delta x_2}{2})}{\Delta x_1 \Delta x_2} \\
&= \lim_{\Delta x_1 \to 0 \Delta x_2 \to 0} \frac{\frac{\Delta x_1 \Delta x_2 \mathrm{Vol}(\mathcal{S}_{b-3}^{1-x_1-x_2})}{\mathrm{Vol}(\mathcal{S}_{b-1})}}{\Delta x_1 \Delta x_2} \\
&= (b-1)(b-2)(1-x_1-x_2)^{b-3}.
\end{aligned} \tag{A.21}$$

Since we are going to use the zero-order Edgeworth expansion, we only need to find the mean vector and the covariance matrix of these random variables. The variances are already found in the previous section of the Appendix. Therefore, we only find the covariance between these variables. Specifically,

$$\begin{aligned}
\mathrm{Cov}[\pi(k_1), \pi(k_2)] &= E[\pi(k_1)\pi(k_2)] - E[\pi(k_1)]E[\pi(k_2)] \\
&= \int_0^1 \int_0^{1-x_1} x_1 x_2 (b-1)(b-2)(1-x_1-x_2)^{b-3} dx_2 dx_1 - \tfrac{1}{b^2} \\
&= \tfrac{-1}{b^2(b+1)},
\end{aligned} \tag{A.22}$$

where in (A.22) we used integration by parts. Hence, considering our definitions in (A.2) for $p_0(k_1)$ and $p_0(k_2)$, we obtain the following for the normal distribution

statistics:

$$\boldsymbol{\mu}^0_{k_1,k_2} = \begin{bmatrix} \alpha_0(\frac{\varepsilon_0}{b} + (1-\varepsilon_0)\pi^0_{ac}(k_1)) \\ \alpha_0(\frac{\varepsilon_0}{b} + (1-\varepsilon_0)\pi^0_{ac}(k_2)) \end{bmatrix}, \tag{A.23}$$

$$\boldsymbol{\Sigma}^0_{k_1,k_2} = \begin{bmatrix} \alpha_0^2\varepsilon_0^2\frac{b-1}{b^2(b+1)|\Pi^0|} & -\alpha_0^2\varepsilon_0^2\frac{1}{b^2(b+1)|\Pi^0|} \\ -\alpha_0^2\varepsilon_0^2\frac{1}{b^2(b+1)|\Pi^0|} & \alpha_0^2\varepsilon_0^2\frac{b-1}{b^2(b+1)|\Pi^0|} \end{bmatrix}. \tag{A.24}$$

Similarly, we can write for the joint distribution of $(p_1(k_1), p_1(k_2))$.

A.3.2 p−Point Class

Since we have partitions in this model, we need to know whether two states belong to the same partition or not. First, suppose that $P^0(k_1) \neq P^0(k_2)$. Then,

$$\Pr(\pi(k_1) = x_1, \pi(k_2) = x_2) = \Pr(\pi(k_1) = x_1)\Pr(\pi(k_2) = x_2), \tag{A.25}$$

from which we get

$$\boldsymbol{\mu}^0_{k_1,k_2} = \begin{bmatrix} \frac{\omega^0_{P^0(k_1)}\alpha_0}{|s^0_{P^0(k_1)}|} \\ \frac{\omega^0_{P^0(k_2)}\alpha_0}{|s^0_{P^0(k_2)}|} \end{bmatrix}, \tag{A.26}$$

$$\boldsymbol{\Sigma}^0_{k_1,k_2} = \begin{bmatrix} \alpha_0^2(\omega^0_{P^0(k_1)})^2\frac{|s^0_{P^0(k_2)}|-1}{|s^0_{P^0(k_1)}|^2(|s^0_{P^0(k_1)}|+1)|\Pi^0|} & 0 \\ 0 & \alpha_0^2(\omega^0_{P^0(k_2)})^2\frac{|s^0_{P^0(k_2)}|-1}{|s^0_{P^0(k_2)}|^2(|s^0_{P^0(k_2)}|+1)|\Pi^0|} \end{bmatrix}. \tag{A.27}$$

Now, suppose $P^0(k_1) = P^0(k_2) = m_{k_1k_2}$. Then

$$\mathrm{Cov}[\pi(k_1), \pi(k_2)] = \frac{-(\omega^0_{m_{k_1k_2}})^2}{|s^0_{m_{k_1k_2}}|^2(|s^0_{m_{k_1k_2}}|+1)}, \tag{A.28}$$

and we have

$$\boldsymbol{\mu}^0_{k_1,k_2} = \begin{bmatrix} \frac{\omega^0_{m_{k_1k_2}}\alpha_0}{|s^0_{m_{k_1k_2}}|} \\ \frac{\omega^0_{m_{k_1k_2}}\alpha_0}{|s^0_{m_{k_1k_2}}|} \end{bmatrix}, \tag{A.29}$$

$$\mathbf{\Sigma}^0_{k_1,k_2} = \begin{bmatrix} \alpha_0^2(\omega^0_{m_{k_1k_2}})^2 \frac{|s^0_{m_{k_1k_2}}|-1}{|s^0_{m_{k_1k_2}}|^2(|s^0_{m_{k_1k_2}}|+1)|\Pi^0|} & -\alpha_0^2(\omega^0_{m_{k_1k_2}})^2 \frac{1}{|s^0_{m_{k_1k_2}}|^2(|s^0_{m_{k_1k_2}}|+1)|\Pi^0|} \\ -\alpha_0^2(\omega^0_{m_{k_1k_2}})^2 \frac{1}{|s^0_{m_{k_1k_2}}|^2(|s^0_{m_{k_1k_2}}|+1)|\Pi^0|} & \alpha_0^2(\omega^0_{m_{k_1k_2}})^2 \frac{|s^0_{m_{k_1k_2}}|-1}{|s^0_{m_{k_1k_2}}|^2(|s^0_{m_{k_1k_2}}|+1)|\Pi^0|} \end{bmatrix}. \tag{A.30}$$

In the following, $\Pr(p_0(k_1) = \alpha, p_0(k_2) = \beta)$ and $\Pr(p_1(k_1) = \alpha, p_1(k_2) = \beta)$ will be denoted by $F^0_{k_1,k_2}(\alpha,\beta)$ and $F^1_{k_1,k_2}(\alpha,\beta)$, respectively. Now, we start by computing the pmf of $(\zeta^0_{k_1,t_1^0}, \zeta^0_{k_2,t_2^0})$. After quite some computation we obtain

$$\Pr(\underline{\zeta}^0_{k_1,t_1^0} = m_1, \underline{\zeta}^0_{k_2,t_2^0} = m_2) = \begin{cases} \mathrm{Int}^0(\theta^0_{k_1,L}, \theta^0_{k_1,U}; \theta^0_{k_2,L}, \theta^0_{k_2,U}); m_1, m_2 \neq 0 \\ \mathrm{Int}^0(-\infty, -g_0 t_1^0; \theta^0_{k_2,L}, \theta^0_{k_2,U}); m_1 = 0, m_2 \neq 0 \\ \mathrm{Int}^0(\theta^0_{k_1,L}, \theta^0_{k_1,U}; -\infty, -g_0 t_2^0); m_2 = 0, m_1 \neq 0 \\ \mathrm{Int}^0(-\infty, -g_0 t_1^0; -\infty, -g_0 t_2^0); m_1 = m_2 = 0 \end{cases} \tag{A.31}$$

$$\Pr(\underline{\zeta}^1_{k_1,t_1^1} = m_1, \underline{\zeta}^1_{k_2,t_2^1} = m_2) = \begin{cases} \mathrm{Int}^1(\theta^1_{k_1,L}, \theta^1_{k_1,U}; \theta^1_{k_2,L}, \theta^1_{k_2,U}); m_1, m_2 \neq 0 \\ \mathrm{Int}^1(-\infty, -g_1 t_1^1; \theta^1_{k_2,L}, \theta^1_{k_2,U}); m_1 = 0, m_2 \neq 0 \\ \mathrm{Int}^1(\theta_{k_1,L}, \theta^1_{k_1,U}; -\infty, -g_1 t_2^1); m_2 = 0, m_1 \neq 0 \\ \mathrm{Int}^1(-\infty, -g_1 t_1^1; -\infty, -g_1 t_2^1); m_1 = m_2 = 0. \end{cases} \tag{A.32}$$

Furthermore, we have

$$\Pr(\overline{\zeta}^0_{k_1,t_1^0} = m_1, \underline{\zeta}^0_{k_2,t_2^0} = m_2) = \begin{cases} \mathrm{Int}^0(\overline{\theta}^0_{k_1,L}, \overline{\theta}^0_{k_1,U}; \overline{\theta}^0_{k_2,L}, \overline{\theta}^0_{k_2,U}); \\ m_1 \neq n_1, m_2 \neq 0 \\ \mathrm{Int}^0(\theta^0_{k_1,L}, \overline{\theta}^0_{k_1,U}; -\infty, -g_0 t_2^0); \\ m_2 = 0, m_1 \neq n_1 \\ \mathrm{Int}^0(-\infty, -g_0 t_1^0; g_1(n_1 - 1) - g_0 s, \infty); \\ m_1 = n_1, m_2 \neq 0 \\ \mathrm{Int}^0(g_1(n_1 - 1) - g_0 t_1^0, \infty; -\infty, -g_0 t_2^0); \\ m_1 = n_1, m_2 = 0 \end{cases} \tag{A.33}$$

$$\Pr(\overline{\zeta}^1_{k_1,t_1^1} = m_1, \underline{\zeta}^1_{k_2,t_2^1} = m_2) = \begin{cases} \mathrm{Int}^1(\overline{\theta}^1_{k_1,L}, \overline{\theta}^1_{k_1,U}; \overline{\theta}^1_{k_2,L}, \overline{\theta}^1_{k_2,U}); \\ m_1 \neq n_1, m_2 \neq 0 \\ \mathrm{Int}^1(\overline{\theta}_{k_1,L}, \overline{\theta}^1_{k_1,U}; -\infty, -g_1 s); \\ m_2 = 0, m_1 \neq n_1 \\ \mathrm{Int}^1(-\infty, -g_1 t_1^1; g_0(n_0 - 1) - g_1 t_2^1, \infty); \\ m_1 = n_1, m_2 \neq 0 \\ \mathrm{Int}^1(g_0(n_0 - 1) - g_1 t_1^1, \infty; -\infty, -g_1 t_2^1); \\ m_1 = n_0, m_2 = 0 \end{cases} \tag{A.34}$$

In equations (A.31)-(A.34) we use the following definitions (the notation $\int\!\!\!\!\int.$ is used to denote $\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}.$)

$$\begin{aligned}
\text{Int}^0(b_L^{k_1}, b_U^{k_1}; b_L^{k_2}, b_U^{k_2}) &:= \int\!\!\!\!\int \Pr\left[\begin{pmatrix} \alpha + b_L^{k_1} \\ \beta + b_L^{k_2} \end{pmatrix} \preceq \begin{pmatrix} p_0(k_1) \\ p_0(k_2) \end{pmatrix} \preceq \begin{pmatrix} \alpha + b_U^{k_1} \\ \beta + b_U^{k_2} \end{pmatrix}\right] \\
&\quad \times F^1_{k_1,k_2}(\alpha,\beta) d\alpha d\beta \\
\text{Int}^1(b_L^{k_1}, b_U^{k_1}; b_L^{k_2}, b_U^{k_2}) &:= \int\!\!\!\!\int \Pr\left[\begin{pmatrix} \alpha + b_L^{k_1} \\ \beta + b_L^{k_2} \end{pmatrix} \preceq \begin{pmatrix} p_1(k_1) \\ p_1(k_2) \end{pmatrix} \preceq \begin{pmatrix} \alpha + b_U^{k_2} \\ \beta + b_U^{k_2} \end{pmatrix}\right] \\
&\quad \times F^0_{k_1,k_2}(\alpha,\beta) d\alpha d\beta
\end{aligned} \tag{A.35}$$

Table D.1 shows the parameters used in equations (A.31)- (A.34).

$\theta^0_{k_1,L} = g_1(m_1-1) - g_0 t_1^0$	$\theta^0_{k_1,U} = g_1 m_1 - g_0 t_1^0$	$\theta^0_{k_2,L} = g_1(m_2-1) - g_0 t_2^0$	$\theta^0_{k_2,U} = g_1 m_2 - g_0 t_2^0$
$\theta^1_{k_1,L} = g_0(m_1-1) - g_1 t_1^1$	$\theta^1_{k_1,U} = g_0 m_1 - g_1 t_1^1$	$\theta^1_{k_2,L} = g_0(m_2-1) - g_1 t_2^1$	$\theta^1_{k_2,U} = g_0 m_2 - g_1 t_2^1$
$\overline{\theta}^0_{k_1,L} = g_1 m_1 - g_0 t_1^0$	$\overline{\theta}^0_{k_1,U} = g_1(m_1+1) - g_0 t_1^0$	$\overline{\theta}^0_{k_2,L} = g_1(m_2-1) - g_0 t_2^0$	$\overline{\theta}^0_{k_2,U} = g_1 m_2 - g_0 t_2^0$
$\overline{\theta}^1_{k_1,L} = g_0 m_1 - g_1 t_1^1$	$\overline{\theta}^1_{k_1,U} = g_0(m_1+1) - g_1 t_1^1$	$\overline{\theta}^1_{k_2,L} = g_0(m_2-1) - g_1 t_2^1$	$\overline{\theta}^1_{k_2,U} = g_0 m_2 - g_1 t_2^1$

Table A.1: Defined parameters.

APPENDIX B

GENERATING UNCERTAINTY CLASSES FROM PATHWAYS

We provide a simple example of how a set of biological pathways can generate an uncertainty class of stochastic network models. Consider three pathways describing the dynamical behavior of two binary genes A and B:

$$B = 1 \implies A = 0 \tag{B.1}$$

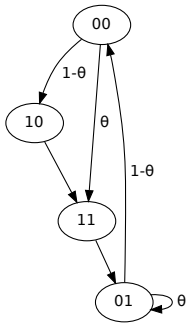
$$B = 0 \implies A = 1 \tag{B.2}$$

$$A = 1 \implies B = 1 \tag{B.3}$$

This simple system is almost completely specified, but when gene A is in state 0, the dynamical behavior of gene B is unspecified. In [1], Layek *et al.* show how to generate an uncertainty class of deterministic networks from a set of pathways by relaxing timing considerations. Knight *et al.* in [74] use a stochastic approach to generate a single Markov chain from a set of pathways and validate the approach using pathways for the NF-κB transcription factor system. In [73] it is shown that the earlier approach in [74] can be generalized to produce a parameterized uncertainty class of Markov chains from a given set of pathways.

Based on [73] we generate the parameterized state transition graph in Fig.B.1. By choosing $\theta \in [0, 1]$ we fix the stochastic evolution of gene B and this characterizes a single Markov chain. We can therefore think of the graph in Fig. B.1 as the state transition graph of an entire uncertainty class of Markov chains.

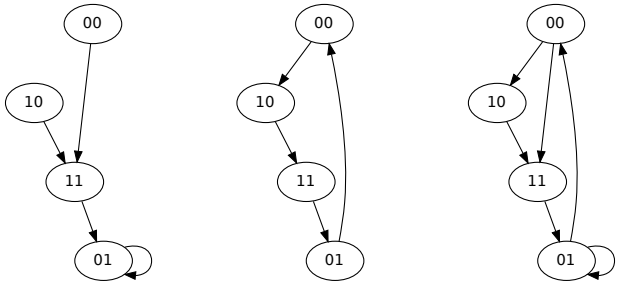
The regularized maximum likelihood classification technique requires a finite uncertainty class so we effectively sample this uncountably infinite, parameterized un-



(a)

Figure B.1: The parameterized state transition graph of a two gene, four state Markov chain system derived from three pathways. The node labels should be read [A,B]. The parameter θ determines the evolution of gene B when gene A$= 0$.

certainty class of Markov chains by discretizing the values of θ. Specifically in this simple example, we use $\theta \in \{0, 0.5, 1\}$ to consider three networks where the behavior of gene B is deterministically up-regulated, deterministically down-regulated, and is a mixture of the two behaviors. The mixture case can be understood to encompass more complex biological regulation such as time-varying pulses or more complex, non-linear stochastic regulation where we only care about the long-run activity. These three networks are shown in Figure B.2.



(a) $\theta = 0$ (b) $\theta = 1$ (c) $\theta = 0.5$

Figure B.2: The state transition graphs of the Markov chains for different values of θ, where all outgoing edges from a given node are equiprobable. Depending on the value of θ the network can have a singleton atttractor state, a large attractor cycle, or a mixture of these two long run behaviors.

APPENDIX C

PROOFS IN SECTION 4

C.1 Conditional Entropy as a Function of Precision Matrix Components

C.1.1 Covaraince Matrix Containing $\bar{R}_x$

From the matrix inversion lemma, $(\mathbf{\Lambda}^{-1}_{\bar{R}_x})_{R_x} = [\mathbf{\Lambda}_{R_x} - \mathbf{\Lambda}_{12}\mathbf{\Lambda}_x^{-1}\mathbf{\Lambda}_{21}]^{-1}$, which is called the *Schur complement* of the $\mathbf{\Lambda}_g$. Moreover, from the properties of the Schur complement we know that ([93]-Appendix 4.4)

$$\log|\mathbf{\Lambda}_{R_x} - \mathbf{\Lambda}_{12}\mathbf{\Lambda}_x^{-1}\mathbf{\Lambda}_{21}| = -\log|\mathbf{\Lambda}_x| + \log|\mathbf{\Lambda}_{\bar{R}_x}|.$$

Hence from the equality in equation (4.13), we obtain $\mathrm{H}(x|R_x) = \log(2\pi e) - \log|\mathbf{\Lambda}_x|$.

C.1.2 $\bar{R}_x$ with Other Genes in $\mathcal{G}$

Denote the precision matrix as in equation (4.16), $\mathbf{\Sigma} = \mathbf{\Lambda}^{-1}$. A 4-block representation of the precision matrix given by

$$\mathbf{\Lambda} = \begin{bmatrix} \mathbf{T}_{11} & \mathbf{T}_{12} \\ \mathbf{T}_{21} & \mathbf{T}_{22} \end{bmatrix}, \tag{C.1}$$

where

$$\mathbf{T}_{11} = \begin{bmatrix} \mathbf{\Lambda}_{R_x} & \mathbf{\Lambda}_{12} \\ \mathbf{\Lambda}_{21} & \mathbf{\Lambda}_x \end{bmatrix}; \mathbf{T}_{12} = \begin{bmatrix} \mathbf{\Lambda}_{13} \\ \mathbf{\Lambda}_{23} \end{bmatrix}; \mathbf{T}_{21} = \begin{bmatrix} \mathbf{\Lambda}_{31} & \mathbf{\Lambda}_{32} \end{bmatrix}; \mathbf{T}_{21} = \mathbf{\Lambda}_{33}.$$

Similarly, denote the covariance matrix by $\mathbf{\Sigma}$ and its 9-block representation as in equation (4.16), where the difference in notation is that $\mathbf{\Lambda}$ in block components is

replaced with $\mathbf{\Sigma}$. Furthermore, denote

$$\mathbf{\Sigma} = \begin{bmatrix} \mathbf{\Sigma}_{R_x} & \mathbf{\Sigma}_{12} & \mathbf{\Sigma}_{13} \\ \mathbf{\Sigma}_{21} & \mathbf{\Sigma}_{x} & \mathbf{\Sigma}_{23} \\ \mathbf{\Sigma}_{31} & \mathbf{\Sigma}_{32} & \mathbf{\Sigma}_{33} \end{bmatrix}.$$

Now, similar to above, the entropy is written as

$$\mathrm{H}(x|R_x) = \log 2\pi e + \log |\begin{bmatrix} \mathbf{\Sigma}_{R_x} & \mathbf{\Sigma}_{12} \\ \mathbf{\Sigma}_{21} & \mathbf{\Sigma}_{x} \end{bmatrix}| - \log |\mathbf{\Sigma}_{R_x}|, \tag{C.2}$$

which needs to be rewritten as a function of the precision matrix. From the matrix inversion lemma, we know that $\begin{bmatrix} \mathbf{\Sigma}_{R_x} & \mathbf{\Sigma}_{12} \\ \mathbf{\Sigma}_{21} & \mathbf{\Sigma}_{x} \end{bmatrix} = (\mathbf{T}_{11} - \mathbf{T}_{12}\mathbf{T}_{22}^{-1}\mathbf{T}_{21})^{-1}$. From the basic linear algebra ([93]-Appendix 4.4),

$$\log |\mathbf{\Lambda}| = \log |\begin{bmatrix} \mathbf{\Lambda}_{x} & \mathbf{\Lambda}_{23} \\ \mathbf{\Lambda}_{32} & \mathbf{\Lambda}_{33} \end{bmatrix}| - \log |\mathbf{\Sigma}_{R_x}|, \tag{C.3}$$

$$-\log |\mathbf{\Lambda}| = \log |\mathbf{\Sigma}| = \log |\begin{bmatrix} \mathbf{\Sigma}_{R_x} & \mathbf{\Sigma}_{12} \\ \mathbf{\Sigma}_{21} & \mathbf{\Sigma}_{x} \end{bmatrix}| - \log |\mathbf{\Lambda}_{33}|. \tag{C.4}$$

Combining equations (C.3)-(C.4) with equation (C.2), we obtain

$$\mathrm{H}(x|R_x) = \log 2\pi e + \log |\mathbf{\Lambda}_{33}| - \log |\begin{bmatrix} \mathbf{\Lambda}_{x} & \mathbf{\Lambda}_{23} \\ \mathbf{\Lambda}_{32} & \mathbf{\Lambda}_{33} \end{bmatrix}|,$$

which can be written as

$$\mathrm{H}_{\mathbf{\Lambda}}(x|R_x) = \log 2\pi e - \log |\mathbf{\Lambda}_{x} - \mathbf{\Lambda}_{23}\mathbf{\Lambda}_{33}^{-1}\mathbf{\Lambda}_{32}|. \tag{C.5}$$

Now, we need to determine the expected conditional entropy:

$$\mathrm{E}_{\mathbf{\Lambda}}[\mathrm{H}_{\mathbf{\Lambda}}(x|R_x)] = \log 2\pi e - \mathrm{E}_{\mathbf{\Lambda}} \log |\mathbf{\Lambda}_x - \mathbf{\Lambda}_{23}\mathbf{\Lambda}_{33}^{-1}\mathbf{\Lambda}_{32}|.$$

From $\mathbf{\Lambda} \sim \mathcal{W}(\mathbf{W}, \kappa)$, we have $\begin{bmatrix} \mathbf{\Lambda}_x & \mathbf{\Lambda}_{23} \\ \mathbf{\Lambda}_{32} & \mathbf{\Lambda}_{33} \end{bmatrix} \sim \mathcal{W}(\begin{bmatrix} \mathbf{W}_x & \mathbf{W}_{23} \\ \mathbf{W}_{32} & \mathbf{W}_{33} \end{bmatrix}, \kappa)$, from which we have $\begin{bmatrix} \mathbf{\Lambda}_x & \mathbf{\Lambda}_{23} \\ \mathbf{\Lambda}_{32} & \mathbf{\Lambda}_{33} \end{bmatrix}^{-1} \sim \mathcal{W}^{-1}(\begin{bmatrix} \mathbf{W}_x & \mathbf{W}_{23} \\ \mathbf{W}_{32} & \mathbf{W}_{33} \end{bmatrix}^{-1}, \kappa)$. Hence,

$$\left[\begin{bmatrix} \mathbf{\Lambda}_x & \mathbf{\Lambda}_{23} \\ \mathbf{\Lambda}_{32} & \mathbf{\Lambda}_{33} \end{bmatrix}^{-1}\right]_{11} \sim \mathcal{W}^{-1}((\mathbf{W}_x - \mathbf{W}_{23}\mathbf{W}_{33}^{-1}\mathbf{W}_{32})^{-1}, \kappa - \dim(\mathbf{W}_{33})). \quad \text{(C.6)}$$

On the other hand we know that $(\mathbf{\Lambda}_x - \mathbf{\Lambda}_{23}\mathbf{\Lambda}_{33}^{-1}\mathbf{\Lambda}_{32})^{-1} = \left[\begin{bmatrix} \mathbf{\Lambda}_x & \mathbf{\Lambda}_{23} \\ \mathbf{\Lambda}_{32} & \mathbf{\Lambda}_{33} \end{bmatrix}^{-1}\right]_{11}$, from which we obtain $(\mathbf{\Lambda}_x - \mathbf{\Lambda}_{23}\mathbf{\Lambda}_{33}^{-1}\mathbf{\Lambda}_{32})^{-1} \sim \mathcal{W}^{-1}((\mathbf{W}_x - \mathbf{W}_{23}\mathbf{W}_{33}^{-1}\mathbf{W}_{32})^{-1}, \kappa - \dim(\mathbf{W}_{33}))$. Finally we have $(\mathbf{\Lambda}_x - \mathbf{\Lambda}_{23}\mathbf{\Lambda}_{33}^{-1}\mathbf{\Lambda}_{32}) \sim \mathcal{W}((\mathbf{W}_x - \mathbf{W}_{23}\mathbf{W}_{33}^{-1}\mathbf{W}_{32}), \kappa - \dim(\mathbf{W}_{33}))$. $\dim(\mathbf{W}_{33})$ is equivalent to $p - |R_x| - 1$. Thus, using the expression for the entropy of a Wishart distributed random matrix (from [90]), we obtain the desired expression.

C.2 Proof of Lemma 3

Similar to the CP_1, the objective function is convex; hence, we only need to show that the constraints are also convex. From the Schur complement properties,

$$\log |\mathbf{W}_x - \mathbf{W}_{23}\mathbf{W}_{33}^{-1}\mathbf{W}_{32}| + \log |\mathbf{W}_{33}| = \log |\begin{bmatrix} \mathbf{W}_x & \mathbf{W}_{23} \\ \mathbf{W}_{32} & \mathbf{W}_{33} \end{bmatrix}|,$$

from which we can write the constraint as

$$-\log|\begin{bmatrix}\mathbf{W}_x & \mathbf{W}_{23}\\ \mathbf{W}_{32} & \mathbf{W}_{33}\end{bmatrix}|+\log|\mathbf{W}_{33}|-\psi(\frac{\kappa-(p-|R_x|-1)}{2})\leq\xi;\xi\geq\underline{\xi}=-\log\pi e.$$

Now, from [92] (Theorem 29), the first two log summands form a convex function and the proof.is complete

C.3 Calculus Required for Solving CP_2

We find the matrix $\mathbf{Hess}(\mathbf{z},\mathbf{y})=\boldsymbol{\nabla}^2(g(\mathbf{z}))-\sum_{l=1}^{3}y(i)\boldsymbol{\nabla}^2(h_l(\mathbf{z}))$. First,

$$\boldsymbol{\nabla}^2(g(\mathbf{z}))=\begin{bmatrix}\frac{1}{2}(1-\lambda)[\mathrm{tr}(\mathbf{W}^{-1}\mathbf{E}_i\mathbf{W}^{-1}\mathbf{E}_j)]_{i,j} & \boldsymbol{\emptyset}\\ \boldsymbol{\emptyset} & \boldsymbol{\emptyset}\end{bmatrix},\tag{C.7}$$

which is a square matrix with size $(\frac{p(p+1)}{2})^2+2$. Define

$$\mathbf{B}_1=\begin{bmatrix}\boldsymbol{\emptyset}_{|R_x|(|R_x|+1)/2} & \boldsymbol{\emptyset}\\ \boldsymbol{\emptyset} & [\mathrm{tr}(\overline{\mathbf{W}}^{-1}\mathbf{E}_i\overline{\mathbf{W}}^{-1}\mathbf{E}_j)]\end{bmatrix},\tag{C.8}$$

and

$$\mathbf{B}_2=\begin{bmatrix}\boldsymbol{\emptyset}_{|\bar{R}_x|(|\bar{R}_x|+1)/2} & \boldsymbol{\emptyset}\\ \boldsymbol{\emptyset} & -[\mathrm{tr}(\mathbf{W}_{33}^{-1}\mathbf{E}_i\mathbf{W}_{33}^{-1}\mathbf{E}_j)]\end{bmatrix},\tag{C.9}$$

where $\overline{\mathbf{W}}=\begin{bmatrix}\mathbf{W}_x & \mathbf{W}_{23}\\ \mathbf{W}_{32} & \mathbf{W}_{33}\end{bmatrix}$. Then

$$\boldsymbol{\nabla}^2(h_3(\mathbf{x}))=\begin{bmatrix}\mathbf{B}_1+\mathbf{B}_2 & \boldsymbol{\emptyset}\\ \boldsymbol{\emptyset} & \boldsymbol{\emptyset}\end{bmatrix}.$$

Denoting the Jacobian matrix of the constraints by $\mathbf{A}(\mathbf{z})$, we may write element-wise

$$\mathbf{A}_{2,i}(\mathbf{z}) = \begin{cases} \mathrm{tr}(\overline{\mathbf{W}}^{-1}\mathbf{E}_i) + \mathrm{tr}(\mathbf{W}_{33}^{-1}\mathbf{E}_i); i \geq |\bar{R}_x|(|\bar{R}_x|+1)/2 + 1 \\ \mathrm{tr}(\overline{\mathbf{W}}^{-1}\mathbf{E}_i); |R_x|(|R_x|+1)/2 + 1 \leq i \leq |\bar{R}_x|(|\bar{R}_x|+1)/2 \\ 1; i = p(p+1)/2 + 1. \end{cases}$$

Similarly, we have $\mathbf{A}_{1,:}(\mathbf{z}) = \mathbf{e}_{p(p+1)/2+1}^T$.

APPENDIX D

PRELIMINARIES AND PROOFS IN SECTION 5

D.1 Dirichlet Distribution: Definition and Properties

The ratio $\frac{\prod_{k=1}^{b}\Gamma(\alpha_k)}{\Gamma(\alpha_0)}$, called multinomial Beta function, is denoted by $B(\boldsymbol{\alpha})$ or similarly by $B([\alpha_1, ..., \alpha_b])$.

The five major properties of the Dirichlet distribution, frequently used in this dissertation, are listed below (for properties $\mathbf{P1} - \mathbf{P3}$, refer to [108]- [109])

P1. If $[p_1, p_2, ..., p_b] \sim \mathcal{D}ir(\alpha_1, \alpha_2, ..., \alpha_b)$ and $r_1, ..., r_l$ are integers such that $0 < l_1 < ... < r_l = b$, then

$$(\sum_{i=1}^{r_1} p_i, \sum_{i=r_1+1}^{r_2} p_i, ..., \sum_{i=r_{l-1}+1}^{r_l} p_i) \sim \mathcal{D}ir(\sum_{i=1}^{r_1} \alpha_i, \sum_{i=r_1+1}^{r_2} \alpha_i, ..., \sum_{i=r_{l-1}+1}^{r_l} \alpha_i). \quad \text{(D.1)}$$

P2. Assuming the assumption in I, then each p_i is (marginally) distributed as follows

$$p_i \sim \mathcal{B}eta(\alpha_i, \alpha_0 - \alpha_i). \quad \text{(D.2)}$$

Considering the assumptions above, if properties **P1.** and **P2.** are combined, then for some positive integer r_1, one obtains

$$\sum_{i=1}^{r_1} p_i \sim \mathcal{B}eta(\sum_{i=1}^{r_1} \alpha_i, \alpha_0 - \sum_{i=1}^{r_1} \alpha_i). \quad \text{(D.3)}$$

P3. If $[p_1, p_2, ..., p_b] \sim \mathcal{D}ir(\alpha_1, \alpha_2, ..., \alpha_b)$, then for the first and the second moments

we have

$$E[p_i] = \frac{\alpha_i}{\alpha_0}$$
$$E[p_i^2] = \frac{\alpha_i(\alpha_i + 1)}{\alpha_0(\alpha_0)}.$$

P4. If the random vector $\boldsymbol{p}$ is distributed according to the Dirichlet distribution $\mathcal{D}ir(\boldsymbol{\alpha})$, where $\alpha_0 = \sum_i \alpha_i$, then we have [90]

$$E[\log p_k] = \psi(\alpha_k) - \psi(\alpha_0),$$

where ψ is the digamma function.

Now, using the properties above, we prove two fundamental lemmas which are frequently used in our analysis:

Lemma 7. *Suppose that* $[p_1, p_2, \ldots, p_b] \sim \mathcal{D}ir(\alpha_1, \alpha_2, \ldots, \alpha_b)$*, then, for any Lebesgue-measurable function* $g : \mathcal{S}_{b-1} \to \mathbb{R}$*, we have*[*]

$$E_{\boldsymbol{p}}[p_i g(\boldsymbol{p})] = \frac{\alpha_i}{\sum_{k=1}^{b} \alpha_k} E_{\boldsymbol{p}'}[g(\boldsymbol{p}')],$$

in which

$$\boldsymbol{p}' \sim \mathcal{D}ir(\alpha_1', \alpha_2', \ldots, \alpha_b'); \alpha_i' = \alpha_i + 1, \alpha_k' = \alpha_k, k \neq i.$$

Proof. Without loss of generality, we prove the property for $i = 1$. Expanding the expectation as follows

[*] $\mathcal{S}_{b-1}$ denotes the unit simplex in the two-dimensional Euclidean space.

$$
\begin{aligned}
E_{\boldsymbol{p}}[p_1 \log g(\boldsymbol{p})] &= \int p_1 g(\boldsymbol{p}) \tfrac{\Gamma(\sum_{k=1}^b \alpha_k)}{\prod_{k=1}^b \Gamma(\alpha_k)} p_1^{\alpha_1-1} p_2^{\alpha_2-1} p_3^{\alpha_3-1} d\boldsymbol{p} \\
&= \tfrac{\Gamma(\alpha_1+1)\prod_{k=2}^b \Gamma(\alpha_k)}{\Gamma(\alpha_1+1+\sum_{k=2}^b \alpha_k)} \tfrac{\Gamma(\sum_{k=1}^b \alpha_k)}{\prod_{k=1}^b \Gamma(\alpha_k)} \int g(\boldsymbol{p}) \\
&\times \underbrace{\frac{\Gamma(\alpha_1+1+\sum_{k=2}^b \alpha_k)}{\Gamma(\alpha_1+1)\prod_{k=2}^b \Gamma(\alpha_k)} p_1^{(\alpha_1+1)-1} p_2^{\alpha_2-1} p_3^{\alpha_3-1} d\boldsymbol{p}}_{\mathcal{D}ir(\alpha_1+1,\alpha_2,\alpha_3)} \\
&= \tfrac{\alpha_1 \prod_{k=1}^b \Gamma(\alpha_k)}{(\sum_{k=1}^b \alpha_1)\Gamma(\sum_{k=1}^b \alpha_k)} \tfrac{\Gamma(\sum_{k=1}^b \alpha_k)}{\prod_{k=1}^b \Gamma(\alpha_k)} E_{\boldsymbol{p}'}[g(\boldsymbol{p}')];\ \boldsymbol{p}' \sim \mathcal{D}ir(\alpha_1+1,\alpha_2,\alpha_3),
\end{aligned}
\tag{D.4}
$$

where in the last equality, we used the fact that $\Gamma(x+1) = x\Gamma(x)$. Q.E.D.

D.1.1 Proof of Lemma 4

Define $C = \mathcal{X}\backslash(A \cup B)$. From property **P1** we have

$$
(\sum_{i\in A} p_i, \sum_{i\in B} p_i, \sum_{i\in C} p_i) \sim \mathcal{D}ir(\sum_{i\in A} \alpha_i, \sum_{i\in B} \alpha_i, \sum_{i\in C} \alpha_i).
$$

Hence, Lemma 7 indicates that

$$
E_{\boldsymbol{p}}[\frac{\sum_{i\in A} p_i}{\sum_{i\in A} p_i + \sum_{j\in B} p_j}] = \frac{\sum_{i\in A} p_i}{\sum_{i\in A} \alpha_i + \sum_{i\in B} \alpha_i + \sum_{i\in C} \alpha_i} E_{\boldsymbol{p}'}[\frac{1}{\sum_{i\in A} p'_i + \sum_{j\in B} p'_j}]
\tag{D.5}
$$

in which

$$
(\sum_{i\in A} p'_i, \sum_{i\in B} p'_i, \sum_{i\in C} p'_i) \sim \mathcal{D}ir(1 + \sum_{i\in A} \alpha_i, \sum_{i\in B} \alpha_i, \sum_{i\in C} \alpha_i).
\tag{D.6}
$$

Now, suppose $(u, v, w) \sim \mathcal{D}ir(\alpha_u, \alpha_v, \alpha_w)$ where $u + v > 1$. Then, since $(u+v, w) \sim \mathcal{D}ir(\alpha_u+\alpha_v, \alpha_w)$ (i.e. Beta-distributed), we may write $E[\frac{1}{u+v}] = E[\frac{1}{z}]$, where $(z, w) \sim \mathcal{D}ir(\alpha_u + \alpha_v, \alpha_w)$. Finally, we write

$$
\begin{aligned}
E[\tfrac{1}{u+v}] &= \tfrac{\Gamma(\alpha_u+\alpha_v+\alpha_w)}{\Gamma(\alpha_u+\alpha_v)\Gamma(\alpha_w)} \int \tfrac{1}{z} z^{\alpha_u+\alpha_v-1} w^{\alpha_w-1} dz dw = \tfrac{\Gamma(\alpha_u+\alpha_v+\alpha_w)}{\Gamma(\alpha_u+\alpha_v)\Gamma(\alpha_w)} \int z^{\alpha_u+\alpha_v-2} w^{\alpha_w-1} dz dw \\
&= \tfrac{\Gamma(\alpha_u+\alpha_v+\alpha_w)}{\Gamma(\alpha_u+\alpha_v)\Gamma(\alpha_w)} \tfrac{\Gamma(\alpha_u+\alpha_v-1)\Gamma(\alpha_w)}{\Gamma(\alpha_u+\alpha_v+\alpha_w-1)} = \tfrac{\alpha_u+\alpha_v+\alpha_w-1}{\alpha_u+\alpha_v-1}.
\end{aligned}
$$

According to equation (D.6), we have $\alpha_u = 1 + \sum_{i\in A}\alpha_i$, $\alpha_v = \sum_{i\in B}\alpha_i$, and $\alpha_w = \sum_{i\in C}\alpha_i$. Combining equations (D.5) and (D.7), the proof for the first moment is finished.

For the variance, we find the second moment and then equation (5.14) would be the direct result of combining the first two moments. Writing the second moment, we have

$$\begin{aligned} E_{\boldsymbol{p}}\Big[[\frac{\sum_{i\in A} p_i}{\sum_{i\in A} p_i+\sum_{j\in B} p_j}]^2\Big] &= \frac{\sum_{i\in A}\alpha_i(\sum_{i\in A}\alpha_i+1)}{(\sum_{i\in A\cup B\cup C}\alpha_i)(\sum_{i\in A\cup B\cup C}\alpha_i+1)} \\ &\times E_{\boldsymbol{p}''}\Big[[\frac{1}{\sum_{i\in A} p_i''+\sum_{j\in B} p_j''}]^2\Big] \end{aligned} \tag{D.7}$$

where $\boldsymbol{p}'' \sim \mathcal{D}ir(2+\sum_{i\in A}\alpha_i, \sum_{i\in B}\alpha_i, \sum_{i\in C}\alpha_i)$. In equation (D.7), the equality comes from applying Lemma 7 twice. Then similar to the approach leading to equation (D.7), we obtain

$$E_{\boldsymbol{p}}\Big[[\frac{\sum_{i\in A} p_i}{\sum_{i\in A} p_i+\sum_{j\in B} p_j}]^2\Big] = \frac{\sum_{i\in A}\alpha_i(\sum_{i\in A}\alpha_i+1)}{(\sum_{i\in A}\alpha_i+\sum_{j\in B}\alpha_j)(\sum_{i\in A}\alpha_i+\sum_{j\in B}\alpha_j+1)}$$

D.1.2 Proof of Lemma 5

First define $B_k^{0,y},\ k \in \{1,\ldots,2^M\}$ and $y \in \{0,1\}$ as in the Lemma. Moreover, we use $Z_{B_k^{0,y}}$ to denote $\sum_{i\in B_k^{0,y}} u_i$. Then, based on the assumptions, we would have

$$(Z_{B_k^{0,0}}, Z_{B_k^{0,1}}, 1 - Z_{B_k^{0,0}} - Z_{B_k^{0,1}}) \sim \mathcal{M}ult(\sum_{i\in B_k^{0,0}} p_i, \sum_{i\in B_k^{0,1}} p_i, 1 - \sum_{i\in B_k^{0,0}\cup B_k^{0,1}} p_i)$$

Now, we expand the expected conditional entropy as follows

$$\begin{aligned} E\Big[H[Z_{A_0}|Z_{A_1},\ldots,Z_{A_M}]\Big] &= \sum_{k=1}^{2^M}\sum_{y=0}^{1} E\Big[\Pr(Z_{A_0}=y, f_{dec}(Z_{A_1},\ldots,Z_{A_M})=k) \\ &\times \log\Pr(Z_{A_0}=y, f_{dec}(Z_{A_1},\ldots,Z_{A_M})=k)\Big] \\ &- E\Big[\Pr(Z_{A_0}=y, f_{dec}(Z_{A_1},\ldots,Z_{A_M})=k) \\ &\times \log\Pr(f_{dec}(Z_{A_1},\ldots,Z_{A_M})=k)\Big] \end{aligned} \tag{D.8}$$

where $f_{dec}(Z_{A_1}, Z_{A_2}, \ldots, Z_{A_M})$ is a function which maps the binary-valued vector $(Z_{A_1}, Z_{A_2}, \ldots, Z_{A_M})$ to its corresponding decimal number. Then, using the definitions so far, one may write

$$(Z_{A_0}=y, f_{dec}(Z_{A_1},\ldots,Z_{A_M})=k) \equiv^p Z_{B_k^{0,y}}.$$

Hence, equation (D.8) can be rewritten as follows

$$\begin{aligned} E\Big[H[Z_{A_0}|Z_{A_1},\ldots,Z_{A_M}]\Big] &= \sum_{k=1}^{2^M}\sum_{y=0}^{1} E\Big[(\textstyle\sum_{i\in B_k^{0,y}} p_i)\log(\sum_{i\in B_k^{0,y}} p_i)\Big] \\ &- E\Big[(\textstyle\sum_{i\in B_k^{0,y}} p_i)\log(\sum_{i\in B_k^{0,0}} p_i + \sum_{i\in B_k^{0,1}} p_i)\Big] \end{aligned} \tag{D.9}$$

where from property **P1**, we have $(\sum_{i\in B_k^{0,0}} p_i, \sum_{i\in B_k^{0,1}} p_i, 1-\sum_{i\in B_k^{0,0}} p_i-\sum_{i\in B_k^{0,1}} p_i) \sim \mathcal{D}ir(\sum_{i\in B_k^{0,0}} \alpha_i, \sum_{i\in B_k^{0,1}} \alpha_i, \alpha_0 - \sum_{i\in B_k^{0,0}} \alpha_i - \sum_{i\in B_k^{0,1}} \alpha_i)$. That being said, applying Lemma 4 to equation (D.9), knowing property **P4**, the result can be readily derived.

D.2 Maximum Entorpy and Maximal Data Information

D.2.1 Maximum Entropy Method

D.2.1.1 General Methodology

The MaxEnt prior distribution is the solution to the following problem

$$\max_{P:\ X\sim P} H_{\boldsymbol{\theta}}[\boldsymbol{\theta}] = -E_{\boldsymbol{\theta}}[\ln \pi(\boldsymbol{\theta})] \tag{D.10}$$
$$\text{Subject to: } E_{\boldsymbol{\theta}}[f_i(\boldsymbol{\theta})] = \beta_i; i = 1, ..., m,$$

where $f_i(.)$ are some measurable functions, and β_i are known due to prior knowledge. The solution to the optimization problem in equation (D.10) is given by

$$\pi(\boldsymbol{\theta}) = Z^{-1} \exp\{\sum_{i=1}^{m} \gamma_i f_i(\boldsymbol{\theta})\},$$

where Z and γ_i's are, respectively, normalizing factor and the Lagrange multipliers, computed using the constraints in equation (D.10).

D.2.1.2 Multinomial with Dirichlet prior

Considering the Dirichlet prior, when there is no prior information in the form of expectations available, the MaxEnt prior, provided that the parameter α_0 is known, is the solution to the following optimization problem:

$$\max_{\boldsymbol{\alpha}\in\mathcal{S}_{b-1}^{\alpha_0}} \sum_{k=1}^{b} \log\Gamma(\alpha_k) - (\alpha_k - 1)\psi(\alpha_k). \tag{D.11}$$

Solving equation (D.11) using Lagrange multiplier method, and knowing that

$\sum_{k=1}^{b} \alpha_k = \alpha_0$, one may obtain the Dirichlet prior shape as follows

$$\boldsymbol{\alpha} = \alpha_0 \mathbf{1}_b / b.$$

D.2.2 Maximal Data Information Prior

D.2.2.1 General Methodology

The simple MDIP is the solution to the following optimization problem

$$\max \; H[\boldsymbol{\theta}] - E_{\boldsymbol{\theta}}[H[f(x|\boldsymbol{\theta})]] \tag{D.12}$$

$$\text{Subject to: } E_{\boldsymbol{\theta}}[g_i(\boldsymbol{\theta})] = \beta_i; i = 1, \ldots, m \tag{D.13}$$

whose solution is given by

$$\pi(\boldsymbol{\theta}) \propto \exp\{-H[f(x|\boldsymbol{\theta})] + \sum_{i=1}^{m} \gamma_i g_i(\boldsymbol{\theta})\}$$

D.2.2.2 Multinomial with Dirichlet Prior

For example, for the multinomial model (discrete setting), the MDIP prior is given by

$$\pi(\boldsymbol{\theta}) \propto (1 - \sum_{i=1}^{b-1} \theta_j)^{1-\sum_{i=1}^{b-1} \theta_j} \prod_{i=1}^{b-1} \theta_i^{\theta_i} \exp\{\sum_{i=1}^{m} \gamma_i g_i(\boldsymbol{\theta})\}$$

where a normalization factor is needed to result in a proper MDIP. This factor is computed for binomial and trinomial cases in [107].

D.3 Regularization Parameter Selection via Cross-Validation

In order to find a proper regularization parameter, one would need to define a measure of performance. This measure assesses a constructed prior, when the regularization parameter is applied into the process of prior construction. The regularization

parameters $(\lambda^{fun}, \lambda^{reg})$ belong to the simplex $\lambda^{fun} + \lambda^{reg} \leq 1$. Here, we restrict the selection of parameters to a 2D grid of the space, denoted by $\mathbf{\Lambda}$, where we make the grid uniformly with steps of size 0.1 from 0 to 1, for one of the parameters, λ^{fun} and the other one takes values uniformly with the same step size from 0 to $1-\lambda^{fun}$. Then, the search is done in this restricted space for the regularization parameters. In this subsection, we first introduce an oracle regularization which is found by comparing the center of the constructed prior, $\hat{\boldsymbol{\alpha}}/\alpha_0$, with the true probability.

Here, we use cross-validation to choose a near-optimal regularization parameter, the difference would be the measure used in the cross-validation process. The first attempt to using cross-validation for determining the regularization parameter was in ridge regression problem [119, 120]. It was called generalized cross-validation.

In this appendix, we introduce a leave-one-out (LOO) approach to find an estimate for the (prior) expected mean log-likelihood (EMLL), denoted by $\varepsilon\ell$, of the validation points. Then, we choose regularization parameters yielding the largest LOO estimate of the EMLL.

First, we split the data points into two parts: prior construction and classifier training, denoted by $\boldsymbol{u}_{n_p}^{\text{prior}}$ and $\boldsymbol{u}_{n_t}^{\text{trian}}$. Denote the constructed REMLD prior by using $\boldsymbol{u}_{n_p}$, $\mathcal{G}$, and $\boldsymbol{\lambda}_y; y = 0, 1$ as sample points, pathways, and regularization parameter vectors, respectively, with $\boldsymbol{\pi}^y(\boldsymbol{u}_{n_p}, \mathcal{G}; \boldsymbol{\lambda}_0, \boldsymbol{\lambda}_1)$ (for sake of simplicity in notations we will use only $\boldsymbol{\pi}^y_{\boldsymbol{u}_{n_p}}$). Moreover, the sample set with one point held out from its $k-$th bin in class-y is denoted by $\boldsymbol{u}_{n_p-1}^{u_k^y \leftarrow u_k^y - 1}$. Owing to the discreteness nature of the data, fixing $\mathcal{G}, \boldsymbol{\lambda}_0,$ and $\boldsymbol{\lambda}_1$, we may write the LOO estimate of the EMLL for the class-y constructed prior as follows:

$$\hat{\varepsilon\ell}(\mathcal{G}, \boldsymbol{\lambda}_y) = \frac{1}{n_p} \sum_{k=1}^{b} u_k^y \psi(\boldsymbol{\pi}^y_{\boldsymbol{u}_{n_p-1}^{u_k^y \leftarrow u_k^y - 1}}[k]) \tag{D.14}$$

Computing the estimated error via equation (D.14), the regularization vector, for class-y is selected so that the largest LOO estimate is attained, i.e.,

$$\boldsymbol{\lambda}_y^{\text{loo}} = \arg\max_{\boldsymbol{\lambda}_y \in \boldsymbol{\Lambda}} \hat{\varepsilon\ell}(\mathcal{G}, \boldsymbol{\lambda}_y) \tag{D.15}$$

where, similar to above the maximization in equation (D.15) is found on the restricted set $\boldsymbol{\Lambda}$.

From the computational point of view, the number of iterations needed for implementing the LOO does not exceed number of bins. It is due to the structure of discrete (categorical) classification problem: all the points in one bin are seen the same by the optimization framework. Therefore, as equation (D.14) indicates, at each iteration of the LOO, it is enough to decrease number of points in a bin by one, and then solve the optimization framework. The resulting expected log-likelihood can be easily multiplied by the number of observations in that bin. Thus, the maximum number of iterations is the number of nonzero bins.