Analytic Representation of Bayes Labeling and Bayes Clustering Operators for Random Labeled Point Processes

Lori A. Dalton, *Member, IEEE*, Marco E. Benalcázar, Marcel Brun, and Edward R. Dougherty, *Fellow, IEEE*

***Abstract*—Clustering algorithms typically group points based on some similarity criterion, but without reference to an underlying random process to make clustering algorithms rigorously predictive. In fact, there exists a probabilistic theory of clustering in the context of random labeled point sets in which clustering error is defined in terms of the process. In the present paper, given an underlying point process we develop a general analytic procedure for finding an optimal clustering operator, the Bayes clusterer, that corresponds to the Bayes classifier in classification theory. We provide detailed solutions under Gaussian models. Owing to computational complexity we also develop approximations of the Bayes clusterer.**

***Index Terms*—Bayes classification, Bayesian estimation, clustering, pattern recognition, small samples.**

I. Introduction

CLUSTERING algorithms typically group objects based on some notion of similarity with the hope of gaining knowledge about the underlying classes in a problem. Clustering is ubiquitous in genomics, where it has been employed since the earliest days of expression microarrays [1]. For instance, time-series clustering can be used to group genes whose expression levels exhibit similar behavior through time, similarity indicating possible co-regulation. But only a few years after the first use of clustering in the microarray era, the question of its scientific content was raised by Kerr and Churchill [2] when they asked, "How does one make statistical inferences based on the results of clustering?" Indeed, how do we mathematically interpret a data partition created by a clustering algorithm? Jain, *et al.* [3] go so far as to say that "clustering is a subjective process." For scientific knowledge, subjective interpretations are unacceptable.

Our interest in this paper is the characterization of optimal clustering algorithms. As such, we confront the most basic problem of signal processing: a signal (random process) is observed and the aim is to find an operator that optimally predicts a desired signal based on some objective criterion. This optimization takes place within the context of a random-process model that must relate joint characteristics of the observed and true signals, such as the covariance function in the case of optimal linear filtering. In classification, the process consists of random vectors, the process model is the feature-label distribution, operators predict labels, and the objective criterion is the misclassification rate. Clustering algorithms operate on random labeled point sets. Thus, the process consists of random labeled point sets, the process model is the probabilistic structure of these random labeled point sets, operators predict partitions of the point sets, and the objective criterion is the number of errors in the partitioning [4]. As in the case of Wiener filtering, classification and other signal optimization paradigms, the structure of the clustering problem is dictated by the underlying random process, operator class and signal model.

Clustering theory is inherently more difficult than classification because clusterers operate on random point sets instead of random vectors. Whereas a random vector is fully characterized by a probability distribution function, which may be statistically estimated from realizations, in general characterization of random sets requires Choquet capacities via the Choquet-Matheron-Kendall Theorem (see [5]). In one dimension, a probability distribution function involves probabilities of the half-infinite intervals, $(-\infty, b]$ for $b \in \mathbb{R}$, while a capacity functional involves probabilities for all compact sets in $\mathbb{R}$ [5]–[7], thus making modeling and parameter estimation much more difficult. Nevertheless, one has no option other than to study clustering in the framework of random sets if the aim is a general characterization of clustering performance and the subsequent discovery of optimal clustering algorithms. This is not to say that principled formulations of clustering outside the general framework of random sets have not been proposed (we will discuss some of these shortly); however, while these may indeed be useful in particular settings, they do not address the general theory of optimization in the context of random processes.

Manuscript received September 04, 2014; revised January 13, 2015; accepted January 18, 2015. Date of publication February 03, 2015; date of current version February 20, 2015. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Dmitry Malioutov. The work of L. A. Dalton is supported by the National Science Foundation (CCF-1422631). M. E. Benalcázar also acknowledges support from Secretaría Nacional de Educación Superior, Ciencia, Tecnología e Innovación (SENESCYT), Ecuador, and from Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Argentina.

L. A. Dalton is with the Department of Electrical and Computer Engineering and the Department of Biomedical Informatics, The Ohio State University, Columbus, OH 43210 USA (e-mail: dalton@ece.osu.edu).

M. E. Benalcázar is with the Secretaría Nacional de Educación Superior, Ciencia, Tecnología e Innovación (SENESCYT), Ecuador, the Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Argentina, and the Facultad de Ingeniería, Universidad Nacional de Mar del Plata, Mar del Plata, Argentina (e-mail: marco_benalcazar@hotmail.com).

M. Brun is with the Facultad de Ingeniería, Universidad Nacional de Mar del Plata, Mar del Plata, Argentina (e-mail: marcelbrun@gmail.com).

E. R. Dougherty is with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843 USA (e-mail: edward@ece.tamu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2015.2399870

Proceeding from the underlying probability structure and error definition, the most basic issue in classification theory is the existence of an optimal (minimal-error) classifier, called a "Bayes classifier," along with its error, called the "Bayes error." The salient role of a Bayes classifier in the development of classification theory is evident when one opens the pages of *A Probabilistic Theory of Pattern Recognition*, by Devroye, *et al.* [8]. The first section following the Introduction has the title, "The Bayes Problem," and in less than a page of text the authors provide an expression for a Bayes classifier. One cannot get off the ground without it. Not only are basic classifiers such as linear and quadratic discriminant analysis obtained by plugging sample statistics into the Bayes classifier, the overall analysis of design cost is framed in terms of increased error above the Bayes error. Specifically, the expected error of a designed classifier chosen from a class $\mathcal{C}$ of classifiers is decomposed according to

$$\mathrm{E}[\varepsilon_n] = \varepsilon_{\mathrm{Bayes}} + \Delta_{\mathcal{C}} + \mathrm{E}[\Delta_{n,\mathcal{C}}], \tag{1}$$

where ε_n, $\varepsilon_{\mathrm{Bayes}}$, $\Delta_{\mathcal{C}}$, and $\Delta_{n,\mathcal{C}}$ are the error of the classifier designed from a sample of size n, the Bayes error, the cost of constraint (difference between the error of the optimal classifier in class $\mathcal{C}$ and the Bayes error), and the design cost (difference between the errors of the designed classifier and the optimal classifier in $\mathcal{C}$) [9]. The trade-off between constraint and sample size is expressed in the decomposition. Extending these ideas to clustering, we build on the probabilistic framework introduced in [4] by developing an analytical representation of the Bayes (optimal) clusterer given a random-labeled-point-set (signal) model. Further, we present solutions for optimal operators and their errors for several classes of Gaussian models, which provide the first fundamental limits of performance in clustering under known models.

We will show that the Bayes clusterer can be formulated as a discrete optimization problem among all partitions of the data set, where, given a finite set of sample points, the objective is to find the best partition according to some quality metric. This idea is not entirely new, for instance the k-means objective function aims to find the best partition based on the geometry of points. This optimization problem is in general NP hard, thus it is common to resort to suboptimal partition search algorithms, either by constraining the space of the search, as in the standard iterative k-means algorithm, or by constructing a relaxation of the original problem as in spectral clustering [10]. Whereas prior works based on combinatorial optimization essentially start with some intuitive ad-hoc optimization function and focus on devising suboptimal algorithmic methods to optimize this function, our approach first defines a probabilistic model (a point generating stochastic process) and a definition of error. We then show that the optimal clustering operator is equivalent to a combinatorial optimization problem with a specified form. By stepping back and explicitly stating basic assumptions about the point process itself and how clustering performance should be evaluated, clustering transforms from a subjective activity to an objective operation. Furthermore, the Bayes clusterer can illuminate implicit assumptions that are made when applying an objective function.

Mixture models assume observations are drawn from a mixture distribution, representing the presence of sub-populations within an overall population. Mixture models can be used to make inferences about sub-populations given only unlabeled observations of the pooled population by estimating model parameters, for instance by estimating the parameters of Gaussian mixture densities via the expectation-maximization (EM) algorithm [11], [12]. Although these works are based on principled modeling, estimation and optimization methods, they do not optimize clustering relative to clustering error. Rather, inferences based on these methods have been limited to fitting models and evaluating performance using heuristic metrics. In contrast, the Bayes clusterer defines a clustering error, finds the optimal partition that minimizes this error, and reports the optimal error itself. Although the Gaussian models we propose are essentially Bayesian Gaussian mixture models, we have developed a theory that not only provides an optimal clusterer with performance superior to the EM algorithm, but also, for the first time, provides the clustering error itself.

Dirichlet processes (DP) are nonparametric Bayesian models for grouped data that essentially define a probability distribution over (almost surely discrete) probability distributions [13]. When used in clustering, the main advantage of a DP is that it does not require one to specify the number of clusters. As with mixture models, approaches based on DP focus on modeling, with inference limited to fitting models using Markov-Chain Monte-Carlo (MCMC) sampling and evaluating performance using heuristic metrics. Optimal clustering algorithms relative to a clustering error have not been studied for these models, and there is almost no work on directly evaluating the error rate. In contrast, our interest is in finding optimal clustering algorithms that minimize clustering error, and evaluating the error itself. Furthermore, our work is based on a very general point process model, which includes DP models as special cases. While in this paper we have developed optimal clusterers for only a few simple Gaussian models to illustrate our methods, future work will focus on the important and practical problem of modeling.

There has been some work on PAC-Bayesian generalization bounds for several types of clustering problems, see for instance [14]. There are important distinctions between Bayes clustering and a PAC-based approach, most notably: (1) PAC bounds are based on a heuristic objective function (e.g., KL divergence) whereas Bayes clustering is based on a natural definition of clustering error, and (2) PAC-based analysis and optimization is relative to distribution free bounds on the objective function, with no guarantee that these bounds are tight. On the other hand, Bayes clustering is relative to the exact error under a specified random labeled point process.

There have been attempts to address the subjectivity of clustering by cataloging clustering methods for different applications and settings. For instance, [15] points out, "clustering should not be treated as an application-independent mathematical problem, but should always be studied in the context of its end-use. Different techniques to evaluate clustering algorithms have to be developed for different uses of clustering." This does not contradict the main conclusions of our paper. However, rather than heuristically building a "taxonomy" of clustering problems and corresponding evaluation procedures, we argue that one should specify a random labeled point process depending on the application and purpose, and optimize the clustering error with respect to this process. Different random labeled point processes should be developed for different uses

of clustering, but once the model is specified, the clustering procedure is optimized. The current work addresses the optimization problem itself; future work will address the issue of robust model selection and learning models from examples.

Although we have omitted detailed discussions on many other methods, for instance hierarchical clustering, graph-based algorithms [16]–[18], non-negative matrix factorization [19], and others [20], the issues discussed above are ubiquitous throughout the literature.

II. The Probabilistic Framework

The probabilistic theory of clustering introduced in [4] formalizes how point sets are generated, the definition of label and clustering operators, and the definition of errors. The goal is to find a clustering operator that will assign a partition (or labels) to an observed point set with minimal expected error. We begin by reviewing theory in [4].

Given a point set $S \subset \mathbb{R}^d$, where d is the dimension of the space, let $\eta(S)$ denote the number of points in S. A *random labeled point process* (RLPP) is a pair (Ξ, Λ), where Ξ is a point process generating a point set S and Λ generates random labels on points in S. Ξ maps from a probability space to $[\mathbf{N}; \mathcal{N}]$, where $\mathbf{N}$ is the family of finite sequences in $\mathbb{R}^d$ and $\mathcal{N}$ is the smallest σ-algebra on $\mathbf{N}$ such that for any Borel set B in $\mathbb{R}^d$, the mapping $S \mapsto \eta(S \cap B)$ is measurable. The probability measure, ν, of Ξ is determined by the probabilities $\nu(Y)$ for $Y \in \mathcal{N}$, or (thanks to the Choquet-Matheron-Kendall theorem), the system of probabilities $P(\Xi \cap K \neq \emptyset)$ over all compact sets, K, in $\mathbb{R}^d$. A random labeling is a family, $\Lambda = \{\Phi_S : S \in \mathbf{N}\}$, where Φ_S is a random label function on the point set S in $\mathbf{N}$. Denoting the set of labels on individual points by $L = \{1, 2, \ldots, l\}$, Φ_S has a probability mass function P_S on L^S defined by $P_S(\phi_S) = P(\Phi_S = \phi_S | \Xi = S)$, where $\phi_S : S \to L$ is a deterministic label function assigning a label to each point in S.

A label operator λ maps point sets to label functions, $\lambda(S) = \phi_{S,\lambda} \in L^S$. For any set S, label function ϕ_S and label operator λ, define the *label mismatch error* by

$$\varepsilon_\lambda(S, \phi_S) = \frac{1}{\eta(S)} \sum_{\mathbf{x} \in S} I_{\phi_S(\mathbf{x}) \neq \phi_{S,\lambda}(\mathbf{x})}, \tag{2}$$

where I_A is an indicator function equal to 1 if A is true and 0 otherwise. In the notation of the original paper [4], the *error of label function* $\lambda(S)$ is given by

$$\begin{aligned} \varepsilon_\lambda(S) &= E_{\Phi_S}\left[\varepsilon_\lambda(S, \Phi_S) | S\right] \\ &= \frac{1}{\eta(S)} \sum_{\mathbf{x} \in S} P\left(\Phi_S(\mathbf{x}) \neq \phi_{S,\lambda}(\mathbf{x}) | S\right), \end{aligned} \tag{3}$$

and the *error of label operator* λ is then defined by

$$\varepsilon[\lambda] = E_\Xi E_{\Phi_\Xi}\left[\varepsilon_\lambda(\Xi, \Phi_\Xi)\right]. \tag{4}$$

It was shown in [4] that the *Bayes label operator*, λ^*, minimizing both $\varepsilon_\lambda(S)$ and $\varepsilon[\lambda]$, is given by:

$$\phi_{S,\lambda^*}(\mathbf{x}) = \arg\max_{j \in L} P\left(\Phi_S(\mathbf{x}) = j | S\right). \tag{5}$$

That is, we minimize labeling error by assigning each point the label corresponding to the maximum marginal probability.

Clustering involves identifying partitions of a point set rather than the actual labeling, where a partition of S into l clusters has the form $\mathcal{P}_S = \{S_1, S_2, \ldots, S_l\}$ such that the S_i are disjoint and $S = \bigcup_{i=1}^{l} S_i$. A cluster operator ζ maps point sets to partitions, $\zeta(S) = \mathcal{P}_{S,\zeta}$. Clustering is affected by a label switching problem: every clustering operator, ζ, has associated with it a family, F_ζ, of label operators that all always induce the same partitions as ζ. That is, $\lambda \in F_\zeta$ if and only if $\phi_{S,\lambda}$ induces the same partition as $\zeta(S)$ for all $S \in \mathbf{N}$, where a label function ϕ_S induces partition $\mathcal{P}_S = \{S_1, S_2, \ldots, S_l\}$ if $S_i = \{\mathbf{x} \in S : \phi_S(\mathbf{x}) = l_i\}$ for distinct $l_i \in L$. For any set S, label function ϕ_S and cluster operator ζ, define the *cluster mismatch error* by

$$\varepsilon_\zeta(S, \phi_S) = \min_{\lambda \in F_\zeta} \varepsilon_\lambda(S, \phi_S), \tag{6}$$

the *error of partition* $\zeta(S)$ by

$$\varepsilon_\zeta(S) = E_{\Phi_S}\left[\varepsilon_\zeta(S, \Phi_S) | S\right] = E_{\Phi_S}\left[\min_{\lambda \in F_\zeta} \varepsilon_\lambda(S, \Phi_S) \Big| S\right] \tag{7}$$

and the *error of cluster operator* ζ by

$$\varepsilon[\zeta] = E_\Xi E_{\Phi_\Xi}\left[\varepsilon_\zeta(\Xi, \Phi_\Xi)\right] = E_\Xi E_{\Phi_\Xi}\left[\min_{\lambda \in F_\zeta} \varepsilon_\lambda(\Xi, \Phi_\Xi)\right]. \tag{8}$$

In [4], the expectations, $E_\Xi E_{\Phi_\Xi}$, in (8) were inadvertently brought inside the minimum; however, all simulations in both [4] and [21] were done correctly with the expectations outside the minimum. A theoretical consequence of this correction is that an optimal label operator does not induce an optimal clustering operator, a point we will expand on later.

III. The Bayes Clusterer

This section contains the main contribution of the paper: representation of the Bayes clusterer. We begin by showing how the error definitions in Section II can be represented in terms of risk with intuitive cost functions. Although we will focus on the cost functions that fall out of the original definitions of error put forth in [4] (we call these "natural cost functions"), this new formulation illuminates how one might generalize definitions of performance by using different cost functions. A practical form for the Bayes clusterer and Bayes error, which could only be defined abstractly in [4], then falls out naturally from these representations.

A. Error of a Label Function $\lambda(S)$ in Terms of Risk

Via the probability mass function, P_S, (3) becomes

$$\varepsilon_\lambda(S) = \sum_{\phi_S \in L^S} c_S(\phi_{S,\lambda}, \phi_S) P_S(\phi_S), \tag{9}$$

where we define the *label cost function* by

$$c_S\left(\phi_S^1, \phi_S^2\right) = \frac{1}{\eta(S)} \sum_{\mathbf{x} \in S} I_{\phi_S^2(\mathbf{x}) \neq \phi_S^1(\mathbf{x})}, \tag{10}$$

which defines the cost in assigning labeling ϕ_S^1 when the true label is ϕ_S^2 as the proportion of mislabeled points in S. Thus, $\varepsilon_\lambda(S)$, the error of $\lambda(S)$, may be viewed as the average risk

of assigning labeling $\phi_{S,\lambda}$ under the label cost function. c_S depends on neither the model, which only affects the probabilities P_S, nor the actual points in S. It depends only on the relative labels $\phi_S^1(\mathbf{x})$ and $\phi_S^2(\mathbf{x})$ for each point $\mathbf{x} \in S$. That is, given a set $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_{\eta(S)}\}$ and letting $J = \{1, 2, \ldots, \eta(S)\}$, for a labeling ϕ_S of S define a corresponding labeling ϕ_J on J where $\phi_J(j) = \phi_S(\mathbf{x}_j)$. Then $c_J(\phi_J^1, \phi_J^2) = c_S(\phi_S^1, \phi_S^2)$.

The error of a label function may also be viewed as average risk over all points in S. This view utilizes a more intuitive cost function and is computationally more tractable. From (3),

$$\begin{aligned}\varepsilon_\lambda(S) &= \frac{1}{\eta(S)} \sum_{\mathbf{x}\in S} \sum_{\substack{i=1,\ldots,l \\ i\neq \phi_{S,\lambda}(\mathbf{x})}} \sum_{\substack{\phi_S \in L^S \\ \phi_S(\mathbf{x})=i}} P_S(\phi_S) \\ &= \frac{1}{\eta(S)} \sum_{\mathbf{x}\in S} \varepsilon_{\phi_{S,\lambda}}(\mathbf{x}), \end{aligned} \tag{11}$$

where for any label function ϕ_S,

$$\varepsilon_{\phi_S}(\mathbf{x}) = \sum_{i=1}^{l} c_{\mathbf{x}}\left(\phi_S(\mathbf{x}), i\right) P_{\mathbf{x}}(i),$$

$c_{\mathbf{x}}(i,j)$ is a 0-1 cost function, i.e., $c_{\mathbf{x}}(i,j) = 0$ if $i = j$ and $c_{\mathbf{x}}(i,j) = 1$ if $i \neq j$, and

$$P_{\mathbf{x}}(i) = \sum_{\substack{\phi_S \in L^S \\ \phi_S(\mathbf{x})=i}} P_S(\phi_S)$$

is the probability that point $\mathbf{x}$ has label $i \in L$. In other words, $\varepsilon_{\phi_{S,\lambda}}(\mathbf{x})$ is the average risk in assigning label $\phi_{S,\lambda}(\mathbf{x})$ to point $\mathbf{x}$ under 0-1 label cost, and the error of label function $\lambda(S)$ is the average of these risks over all points $\mathbf{x} \in S$.

Under either interpretation, the cost functions may be pre-computed independently of the model and the actual points of S, while the probabilities, $P_S(\phi_S)$, determine the labeling error for a given S.

B. Error of a Partition $\zeta(S)$ in Terms of Risk

For a given point set, S, and labeling, ϕ_S, following (6),

$$\varepsilon_\zeta(S, \phi_S) = \frac{1}{\eta(S)} \min_{\lambda \in F_\zeta} \sum_{\mathbf{x}\in S} I_{\phi_S(\mathbf{x}) \neq \phi_{S,\lambda}(\mathbf{x})}. \tag{12}$$

The minimum above, and also in (6), (7) and (8), appears to be taken over the infinite set F_ζ (the family of label operators inducing clusterer ζ), while in fact all that is needed is to minimize over the finite set $\{\phi_{S,\lambda} | \lambda \in F_\zeta\} \subseteq L^S$ (the family of label functions inducing partition $\zeta(S)$). Define $G_{\mathcal{P}_S}$ such that $\phi_S \in G_{\mathcal{P}_S}$ if and only if ϕ_S induces $\mathcal{P}_S$. Then,

$$\varepsilon_\zeta(S, \phi_S) = \frac{1}{\eta(S)} \min_{\phi_{S,\zeta} \in G_{\zeta(S)}} \sum_{\mathbf{x}\in S} I_{\phi_S(\mathbf{x}) \neq \phi_{S,\zeta}(\mathbf{x})}. \tag{13}$$

The error of partition $\zeta(S)$ under probability mass P_S on the set, L^S, of all labels is

$$\varepsilon_\zeta(S) = \sum_{\phi_S \in L^S} \left(\frac{1}{\eta(S)} \min_{\phi_{S,\zeta} \in G_{\zeta(S)}} \sum_{\mathbf{x}\in S} I_{\phi_S(\mathbf{x}) \neq \phi_{S,\zeta}(\mathbf{x})} \right) P_S(\phi_S).$$

Letting $\mathcal{K}_S$ be the set of all possible partitions of S,

$$\varepsilon_\zeta(S) = \sum_{\mathcal{P}_S \in \mathcal{K}_S} \sum_{\phi_S \in G_{\mathcal{P}_S}} \left(\frac{1}{\eta(S)} \min_{\phi_{S,\zeta} \in G_{\zeta(S)}} \sum_{\mathbf{x}\in S} I_{\phi_S(\mathbf{x}) \neq \phi_{S,\zeta}(\mathbf{x})} \right) P_S(\phi_S). \tag{14}$$

The term inside the parentheses is constant for all $\phi_S \in G_{\mathcal{P}_S}$ since the minimum essentially resolves the label switching problem relative to ϕ_S, and the ϕ_S in $G_{\mathcal{P}_S}$ are all identical except for permutations of the labels. Hence,

$$\varepsilon_\zeta(S) = \sum_{\mathcal{P}_S \in \mathcal{K}_S} \left(\frac{1}{\eta(S)} \min_{\phi_{S,\zeta} \in G_{\zeta(S)}} \sum_{\mathbf{x}\in S} I_{\phi_{S,\mathcal{P}_S}(\mathbf{x}) \neq \phi_{S,\zeta}(\mathbf{x})} \right) \times \sum_{\phi_S \in G_{\mathcal{P}_S}} P_S(\phi_S), \tag{15}$$

where $\phi_{S,\mathcal{P}_S}$ is any fixed member of $G_{\mathcal{P}_S}$. Finally, we write

$$\varepsilon_\zeta(S) = \sum_{\mathcal{P}_S \in \mathcal{K}_S} c_S(\zeta(S), \mathcal{P}_S) P_S(\mathcal{P}_S), \tag{16}$$

where we define the *partition cost function*,

$$c_S(\mathcal{Q}_S, \mathcal{P}_S) = \frac{1}{\eta(S)} \min_{\phi_{S,\mathcal{Q}_S} \in G_{\mathcal{Q}_S}} \sum_{\mathbf{x}\in S} I_{\phi_{S,\mathcal{P}_S}(\mathbf{x}) \neq \phi_{S,\mathcal{Q}_S}(\mathbf{x})},$$

with $\phi_{S,\mathcal{P}_S}$ being any member of $G_{\mathcal{P}_S}$, and

$$P_S(\mathcal{P}_S) = \sum_{\phi_S \in G_{\mathcal{P}_S}} P_S(\phi_S) \tag{17}$$

is the probability mass function on partitions $\mathcal{P}_S \in \mathcal{K}_S$ of S. As in the case of label functions, the cost c_S between two partitions depends on neither the model nor the actual points in S or their order. To illustrate, given a set $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_{\eta(S)}\}$, let $J = \{1, 2, \ldots, \eta(S)\}$ and for any partition $\mathcal{P}_S = \{S_1, S_2, \ldots, S_l\}$ of S define a corresponding partition $\mathcal{P}_J = \{J_1, J_2, \ldots, J_l\}$ on J, where $j \in J_i$ if and only if $\mathbf{x}_j \in S_i$. Then

$$c_J(\mathcal{Q}_J, \mathcal{P}_J) = c_S(\mathcal{Q}_S, \mathcal{P}_S). \tag{18}$$

Hence, c_J may be pre-computed as a matrix and utilized for any model or point set S, each row, say, corresponding to a *candidate* partition, $\mathcal{Q}_S$, and each column corresponding to a *reference* partition, $\mathcal{P}_S$, to which the candidates are compared.

c_S is also a metric on $\mathcal{K}_S$. That is, c_S has the following properties for all $\mathcal{P}^1, \mathcal{P}^2, \mathcal{P}^3 \in \mathcal{K}_S$: (i) $c_S(\mathcal{P}^1, \mathcal{P}^2) \geq 0$ (in particular, $\eta(S) c_S(\mathcal{P}^1, \mathcal{P}^2)$ is an integer between 0 and $\eta(S) - 1$); (ii) $c_S(\mathcal{P}^1, \mathcal{P}^2) = 0$ if and only if $\mathcal{P}^1 = \mathcal{P}^2$; (iii) $c_S(\mathcal{P}^1, \mathcal{P}^2) = c_S(\mathcal{P}^2, \mathcal{P}^1)$; and (iv) $c_S(\mathcal{P}^1, \mathcal{P}^3) \leq c_S(\mathcal{P}^1, \mathcal{P}^2) + c_S(\mathcal{P}^2, \mathcal{P}^3)$. Regarding property (iv), the triangle inequality, for any fixed $\phi^2 \in G_{\mathcal{P}^2}$,

$$\begin{aligned} &\eta(S)\left(c_S(\mathcal{P}^1, \mathcal{P}^2) + c_S(\mathcal{P}^2, \mathcal{P}^3)\right) \\ &\quad = \min_{\phi^1 \in G_{\mathcal{P}^1}} \min_{\phi^3 \in G_{\mathcal{P}^3}} \sum_{\mathbf{x}\in S} \left(I_{\phi^1(\mathbf{x}) \neq \phi^2(\mathbf{x})} + I_{\phi^2(\mathbf{x}) \neq \phi^3(\mathbf{x})}\right) \\ &\quad \geq \min_{\phi^1 \in G_{\mathcal{P}^1}} \min_{\phi^3 \in G_{\mathcal{P}^3}} \sum_{\mathbf{x}\in S} I_{\phi^1(\mathbf{x}) \neq \phi^3(\mathbf{x})} = \eta(S) c_S(\mathcal{P}^1, \mathcal{P}^3), \end{aligned}$$

where the inequality holds since for any $a, b, c \in L$, $I_{a\neq b} + I_{b\neq c} \geq I_{a\neq c}$.

C. Bayes Clusterer and Bayes Clustering Error

A *Bayes cluster operator*, ζ^*, is a clusterer with minimal error, $\varepsilon[\zeta^*]$. Since $\varepsilon[\zeta] = E_\Xi[\varepsilon_\zeta(\Xi)]$ and $\varepsilon_\zeta(S)$ depends on the clusterer ζ only at the point $\zeta(S)$, minimization of $\varepsilon[\zeta]$ can be achieved by minimizing $\varepsilon_\zeta(S)$ for every $S \in \mathbf{N}$ separately. Hence, a Bayes cluster operator is defined by a *Bayes partition*, $\zeta^*(S)$, for each set $S \in \mathbf{N}$. By (16),

$$\begin{aligned}\zeta^*(S) &= \arg\min_{\zeta(S)\in\mathcal{K}_S} \varepsilon_\zeta(S) \\ &= \arg\min_{\zeta(S)\in\mathcal{K}_S} \sum_{\mathcal{P}_S\in\mathcal{K}_S} c_S\left(\zeta(S), \mathcal{P}_S\right) P_S(\mathcal{P}_S). \quad (19)\end{aligned}$$

Provided that $P_S(\mathcal{P}_S)$ can be found in a given model and the size of $\mathcal{K}_S$ is feasible, by (19) it is now possible to find a Bayes clusterer. Furthermore, we call $\varepsilon_{\zeta^*}(S)$ the *Bayes partitioning error* and $\varepsilon[\zeta^*] = E_\Xi[\varepsilon_{\zeta^*}(\Xi)]$ the *Bayes clustering error*. This is all parallel to classification: a RLPP corresponds to a feature-label distribution, ζ^* corresponds to a Bayes classifier, $\varepsilon_\zeta(S)$ corresponds the true error for an arbitrary classifier at a fixed point, and $\varepsilon[\zeta]$ corresponds to the true error for an arbitrary classifier. Since the Bayes clusterer may be solved for each fixed S individually, from here forward we sometimes write partitions and label functions without a subscript S, with the understanding that they are relative to a fixed point set S. We also write expectations and probabilities without conditioning on S, with the understanding that conditioning on the sample is implicit throughout.

Consider identifying the best partition of a point set S consisting of $n = \eta(S)$ points among all possible partitions given known probabilities for each partition. Let $\mathcal{C}_S \subseteq \mathcal{K}_S$ be the set of *candidate* partitions, which comprise the search space, and $\mathcal{R}_S \subseteq \mathcal{K}_S$ be the set of *reference* partitions, which have known probabilities. Suppose that $\mathcal{C}_S$ is indexed such that $\mathcal{C}_S = \{\mathcal{Q}^1, \ldots, \mathcal{Q}^{|\mathcal{C}_S|}\}$, and that $\mathcal{R}_S$ is indexed such that $\mathcal{R}_S = \{\mathcal{P}^1, \ldots, \mathcal{P}^{|\mathcal{R}_S|}\}$. Then the optimal clustering problem can be formulated relative to a column vector of probabilities, $\mathbf{p} = \{p_j\}$, where $p_j = P_S(\mathcal{P}^j)$ for $j = 1, \ldots, |\mathcal{R}_S|$, a cost matrix, $C = \{c_{ij}\}$, where $c_{ij} = c_S(\mathcal{Q}^i, \mathcal{P}^j)$ for $i = 1, \ldots, |\mathcal{C}_S|$ and $j = 1, \ldots, |\mathcal{R}_S|$, and a column vector of the candidate partition errors, $\mathbf{e} = \{e_i\}$, where e_i is the error of candidate partition $\mathcal{Q}^i$ for $i = 1, \ldots, |\mathcal{C}_S|$. From (16), $\mathbf{e} = C\mathbf{p}$. The Bayes partition for S is then given by $\mathcal{Q}^{i^*}$, where $i^* = \arg\min_{i=1,\ldots,|\mathcal{C}_S|} e_i$, and the Bayes partitioning error, equivalent to the error of the Bayes partition, $\mathcal{Q}^{i^*}$, is simply e_{i^*}. Interestingly, $\mathcal{Q}^{i^*}$ can be a partition with probability zero ($p_{i^*} = 0$), provided e_{i^*} is minimal.

IV. Optimal Computation Reduction

The representation $\mathbf{e} = C\mathbf{p}$ is problematic owing to the size of $\mathcal{C}_S$ and $\mathcal{R}_S$. In a brute force search, we set $\mathcal{C}_S = \mathcal{K}_S$ and $\mathcal{R}_S = \mathcal{K}_S$, requiring a cost matrix C of size $|\mathcal{K}_S| \times |\mathcal{K}_S|$, which can be prohibitively large even for moderate size point sets. In this section, we will use following theorem to alleviate this problem without sacrificing optimality.

Theorem 1: Let $i^* \in \{1, \ldots, |\mathcal{C}_S|\}$ index the Bayes partition. Let $\mathcal{C}'_S = \{\mathcal{Q}^1, \ldots, \mathcal{Q}^{|\mathcal{C}'_S|}\} \subseteq \mathcal{C}_S$ be any subset of candidate partitions, and $\mathcal{R}'_S = \{\mathcal{P}^1, \ldots, \mathcal{P}^{|\mathcal{R}'_S|}\} \subseteq \mathcal{R}_S$ be any subset of reference partitions. Then

$$\sum_{j=1}^{|\mathcal{R}'_S|} c_{i^*j} p_j \leq \left(\min_{1\leq i\leq |\mathcal{C}'_S|} \sum_{j=1}^{|\mathcal{R}'_S|} c_{ij} p_j \right) + M \left(1 - \sum_{j=1}^{|\mathcal{R}'_S|} p_j \right) \quad (20)$$

whenever $c_{ij} \leq M$ for all i and j.

Proof: For all $i \in \{1, \ldots, |\mathcal{C}_S|\}$,

$$\sum_{j=1}^{|\mathcal{R}_S|} c_{ij} p_j = \sum_{j=1}^{|\mathcal{R}'_S|} c_{ij} p_j + \sum_{j=|\mathcal{R}'_S|+1}^{|\mathcal{R}_S|} c_{ij} p_j, \quad (21)$$

so that

$$\sum_{j=1}^{|\mathcal{R}'_S|} c_{ij} p_j \leq \sum_{j=1}^{|\mathcal{R}_S|} c_{ij} p_j \leq \sum_{j=1}^{|\mathcal{R}'_S|} c_{ij} p_j + M \left(1 - \sum_{j=1}^{|\mathcal{R}'_S|} p_j \right). \quad (22)$$

For the Bayes partition, i^*, and for all $i \in \{1, \ldots, |\mathcal{C}_S|\}$,

$$\sum_{j=1}^{|\mathcal{R}_S|} c_{i^*j} p_j \leq \sum_{j=1}^{|\mathcal{R}_S|} c_{ij} p_j. \quad (23)$$

Applying (22) on both sides of this equation,

$$\sum_{j=1}^{|\mathcal{R}'_S|} c_{i^*j} p_j \leq \sum_{j=1}^{|\mathcal{R}'_S|} c_{ij} p_j + M \left(1 - \sum_{j=1}^{|\mathcal{R}'_S|} p_j \right). \quad (24)$$

Since this is true for all i, we may use the tightest bound among any subset $\mathcal{C}'_S \subseteq \mathcal{C}_S$. ■

With the natural cost function in Theorem 1, the constant $M = \frac{n-1}{n}$ can always be used ($c_{ij} = \frac{n-1}{n}$ if partition i assigns every point to the same cluster but partition j assigns every point to a distinct cluster). With two clusters we may use a tighter bound $M = \frac{1}{2}$ for n even or $M = \frac{n-1}{2n}$ for n odd.

This theorem is very flexible, since there are no requirements on the partitions included in $\mathcal{C}'_S$ and $\mathcal{R}'_S$, and in the following subsections we will present several examples of how it can be used to reduce the size of $\mathcal{C}_S$ and $\mathcal{R}_S$. These methods tend to be most effective in problems with moderate to low Bayes error because the distribution P_S over partitions $\mathcal{P}_S$ is more concentrated on only a few partitions with high probability, p_j, and thus it is possible to focus on these few partitions representing most of the probability. With high Bayes error, this distribution tends to be more spread out so that no partition can be ignored.

A. Reducing the Space of Candidate Partitions

Theorem 1 can be used to eliminate candidate partitions from consideration without fully computing their error, and in some cases without any computation at all, by checking if (20) is violated. To illustrate, suppose the natural cost function is in use and $\mathcal{C}'_S = \mathcal{R}'_S = \{\mathcal{P}^1\}$, where $\mathcal{P}^1$ has probability p_1. By Theorem 1, any candidate partition, $\mathcal{P}^i$, can only be the Bayes partition if $c_{i1}p_1 \leq c_{11}p_1 + M(1-p_1) = M(1-p_1)$, where we have used the fact that $c_{11} = 0$. Furthermore, since $c_{i1} \geq 1/n$ for all $i \neq 1$, a partition $\mathcal{P}^i$ for $i \neq 1$ can only be the Bayes partition if $p_1 \leq \frac{Mn}{Mn+1}$. Equivalently, partition $\mathcal{P}^1$ is guaranteed to be the Bayes partition if $p_1 > \frac{Mn}{Mn+1}$. This inequality gives

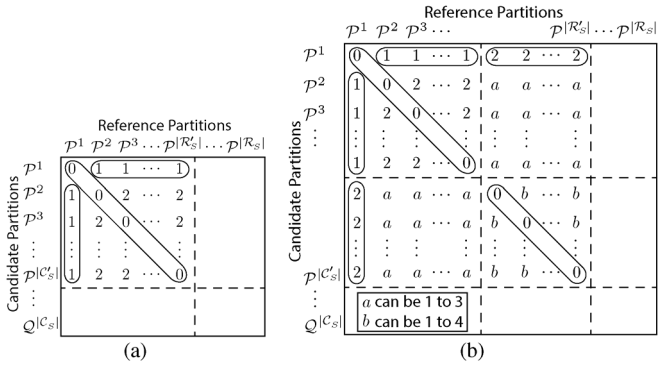


Fig. 1. Example cost matrices (all elements are scaled by a factor of n). (a) Arbitrary partition, $\mathcal{P}^1$, listed first, followed by all partitions with exactly one mismatch from $\mathcal{P}^1$; (b) Arbitrary partition, $\mathcal{P}^1$, listed first, followed by all partitions with exactly one mismatch from $\mathcal{P}^1$, followed by all partitions with exactly two mismatches from $\mathcal{P}^1$.

a test to check if $\mathcal{P}^1$ has a probability that is so high that it is guaranteed to be the Bayes partition.

Next, suppose $\mathcal{C}'_S = \mathcal{R}'_S = \{\mathcal{P}^1, \ldots, \mathcal{P}^{|\mathcal{R}'_S|}\}$, where $\mathcal{P}^1$ is any partition and $\mathcal{P}^2, \ldots, \mathcal{P}^{|\mathcal{R}'_S|}$ correspond to all partitions that have one point mismatched relative to $\mathcal{P}^1$. An illustration of the cost matrix is shown in Fig. 1(a). We have forced a structure on the cost matrix, where the upper left hand square has an all-zero diagonal, the non-diagonal elements in the first row and column are $1/n$, and all remaining elements are $2/n$. In this case,

$$\min_{1\le i\le|\mathcal{C}'_S|} \sum_{j=1}^{|\mathcal{R}'_S|} c_{ij}p_j \le \sum_{j=1}^{|\mathcal{R}'_S|} c_{1j}p_j = \frac{1}{n}\sum_{j=2}^{|\mathcal{R}'_S|} p_j, \tag{25}$$

so that by Theorem 1 the Bayes partition must satisfy

$$\sum_{j=1}^{|\mathcal{R}'_S|} c_{i^*j}p_j \le M - Mp_1 - \left(M - \frac{1}{n}\right)\sum_{j=2}^{|\mathcal{R}'_S|} p_j. \tag{26}$$

The imposed cost-matrix structure assures $c_{i1} \ge 2/n$ and $c_{ij} \ge 1/n$ for $i \notin \mathcal{C}'_S$ and $j = 2, \ldots, |\mathcal{R}'_S|$. Thus, for $i \notin \mathcal{C}'_S$,

$$\sum_{j=1}^{|\mathcal{R}'_S|} c_{ij}p_j \ge \frac{2}{n}p_1 + \frac{1}{n}\sum_{j=2}^{|\mathcal{R}'_S|} p_j. \tag{27}$$

Hence, if the Bayes partition is not in $\mathcal{C}'_S$, then by combining (26) and (27), we have

$$\frac{2}{n}p_1 + \frac{1}{n}\sum_{j=2}^{|\mathcal{R}'_S|} p_j \le M - Mp_1 - \left(M - \frac{1}{n}\right)\sum_{j=2}^{|\mathcal{R}'_S|} p_j. \tag{28}$$

Taking the contrapositive, we can guarantee that $i^* \in \mathcal{C}'_S$ if

$$p_1 > \frac{Mn}{Mn+2}\left(1 - \sum_{j=2}^{|\mathcal{R}'_S|} p_j\right), \tag{29}$$

and hence avoid computing e_i for all $i \notin \mathcal{C}'_S$.

The next theorem generalizes this procedure, so that we may successively add partitions to $\mathcal{C}'_S$ until it is guaranteed that $i^* \in \mathcal{C}'_S$.

Theorem 2: Let $i^* \in \{1, \ldots, |\mathcal{C}_S|\}$ index the Bayes partition, under the natural cost function, among all candidate partitions, $d \ge 1$ be an integer, and $\mathcal{C}'_S = \mathcal{R}'_S$ consist of an arbitrary partition, $\mathcal{P}^1$, and all partitions with cost at most d/n relative to $\mathcal{P}^1$. Moreover, suppose that for each k, $0 \le k \le d$, a_k is the set of all indices for partitions with cost k/n relative to $\mathcal{P}^1$. Suppose the minimum in (20) is achieved by i', that is,

$$i' = \arg\min_{1\le i\le|\mathcal{C}'_S|} \sum_{j=1}^{|\mathcal{R}'_S|} c_{ij}p_j. \tag{30}$$

Then $i^* \in \mathcal{C}'_S$ if $c_{ij} \le M$ for all i and j and

$$p_1 > \frac{Mn - \sum_{k=1}^{d}\sum_{j\in a_k}(Mn+d+1-k-c_{i'j}n)p_j}{Mn + d + 1 - c_{i'1}n}. \tag{31}$$

Furthermore, the same conclusion holds for the following weaker bound, not requiring i':

$$p_1 > \frac{Mn - \sum_{k=1}^{d}(Mn + d - 2k + 1)\sum_{j\in a_k} p_j}{Mn + d + 1}. \tag{32}$$

Proof: An example of the structure imposed on the cost function for $d = 2$ is illustrated in Fig. 1(b), where we assume the partitions in $\mathcal{R}'_S$ are ordered by their cost relative to $\mathcal{P}^1$. If $i \notin \mathcal{C}'_S$, then $c_{ij} \ge (d + 1 - k)/n$ for $j \in a_k$ and $k = 0, \ldots, d$. Thus, by construction,

$$\sum_{j=1}^{|\mathcal{R}'_S|} c_{ij}p_j \ge \sum_{k=0}^{d}\frac{d+1-k}{n}\sum_{j\in a_k} p_j. \tag{33}$$

Combining i' and (33) with Theorem 1 guarantees $i^* \in \mathcal{C}'_S$ if

$$\sum_{k=0}^{d}\frac{d+1-k}{n}\sum_{j\in a_k} p_j \le \sum_{j=1}^{|\mathcal{R}'_S|} c_{i'j}p_j + M\left(1 - \sum_{j=1}^{|\mathcal{R}'_S|} p_j\right).$$

Solving for p_1 yields (31). Rather than finding i', weaker bounds on the minimum in Theorem 1 are obtained by noting

$$\min_{1\le i\le|\mathcal{C}'_S|}\sum_{k=0}^{d}\sum_{j\in a_k} c_{ij}p_j \le \sum_{k=0}^{d}\sum_{j\in a_k} c_{1j}p_j = \sum_{k=1}^{d}\sum_{j\in a_k}\frac{k}{n}p_j.$$

Combining this and (33) with Theorem 1 and solving for p_1 results in (32). ∎

The flow chart in Fig. 2 uses the preceding methods to determine a subset of candidate partitions guaranteed to contain the Bayes partition. There are many other methods for reducing the search space by enforcing a structure on the cost matrix C and applying Theorems 1 or 2; however, given the length of the current paper we will pursue this no further.

B. Reducing the Space of Reference Partitions

Any reference partition with probability zero can be omitted from consideration. Can we reduce the set of reference partitions

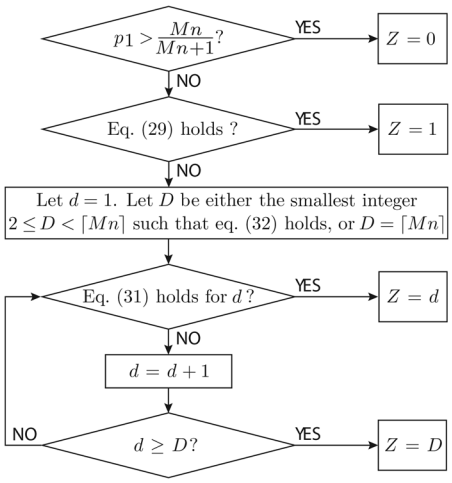


Fig. 2. A flow chart for determining a subset of candidate partitions containing the optimal partition. The final subset consists of an initial partition, $\mathcal{P}^1$, and all partitions with at most Z points mismatched relative to $\mathcal{P}^1$.

further? Suppose we have identified a list, $\mathcal{C}_S^0$, of candidate partitions that must contain the Bayes partition. Rather than evaluate the exact error for every partition in $\mathcal{C}_S^0$ using all reference partitions, we would like to know if a subset of reference partitions, $\mathcal{R}_S^0 = \{\mathcal{P}^1, \ldots, \mathcal{P}^{|\mathcal{R}_S^0|}\}$, can be used to further shrink $\mathcal{C}_S^0$ or even identify the Bayes partition. By Theorem 1, it is evident that any candidate partition $i \in \mathcal{C}_S^0$ for which

$$\sum_{j=1}^{|\mathcal{R}_S^0|} c_{ij} p_j > \left(\min_{1 \le k \le |\mathcal{C}_S^0|} \sum_{j=1}^{|\mathcal{R}_S^0|} c_{kj} p_j \right) + M \left(1 - \sum_{j=1}^{|\mathcal{R}_S^0|} p_j \right) \quad (34)$$

may be eliminated from consideration.

This may be done iteratively: for each integer $t > 0$, define $\mathcal{C}_S^t \subseteq \mathcal{C}_S^{t-1}$ to be the set of candidate partitions not yet eliminated, meanwhile keeping track of $\sum_{j=1}^{|\mathcal{R}_S^{t-1}|} c_{ij} p_j$ for each $\mathcal{P}^i \in \mathcal{C}_S^t$, as well as $\sum_{j=1}^{|\mathcal{R}_S^{t-1}|} p_j$. Then define a new larger set of reference partitions, $\mathcal{R}_S^t \supset \mathcal{R}_S^{t-1}$ (in our implementation we add the highest probability partitions one at a time), and update

$$\sum_{j=1}^{|\mathcal{R}_S^t|} c_{ij} p_j = \sum_{j=1}^{|\mathcal{R}_S^{t-1}|} c_{ij} p_j + \sum_{j=|\mathcal{R}_S^{t-1}|+1}^{|\mathcal{R}_S^t|} c_{ij} p_j \quad (35)$$

for each $\mathcal{P}^i \in \mathcal{C}_S^t$ and $\sum_{j=1}^{|\mathcal{R}_S^t|} p_j = \sum_{j=1}^{|\mathcal{R}_S^{t-1}|} p_j + \sum_{j=|\mathcal{R}_S^{t-1}|+1}^{|\mathcal{R}_S^t|} p_j$. Then eliminate partitions in $\mathcal{C}_S^t$ where (34) holds with $\mathcal{C}_S^t$ and $\mathcal{R}_S^t$ in place of $\mathcal{C}_S^0$ and $\mathcal{R}_S^0$. Repeat until $\mathcal{C}_S^t$ contains only one candidate partition, which may occur with a small reference set, $\mathcal{R}_S^t$, and greatly reduced computation.

C. Finding the Bayes Error and Arbitrary Partition Error

At each stage of the above algorithm to reduce the space of reference partitions, we obtain successively tighter bounds on the Bayes partitioning error: if the procedure ends at step $T \ge 0$, having identified i^* as the index of the Bayes partition, then the Bayes error satisfies

$$\sum_{j=1}^{|\mathcal{R}_S^T|} c_{i^* j} p_j \le e_{i^*} \le \sum_{j=1}^{|\mathcal{R}_S^T|} c_{i^* j} p_j + M \left(1 - \sum_{j=1}^{|\mathcal{R}_S^T|} p_j \right), \quad (36)$$

where $\sum_{j=1}^{|\mathcal{R}_S^T|} c_{i^* j} p_j$ and $\sum_{j=1}^{|\mathcal{R}_S^T|} p_j$ are automatically supplied by the above algorithm.

If these bounds are not tight enough for the application, they can be made as tight as desired by expanding the set of reference partitions further, i.e., with $\mathcal{R}_S^{T+1} \supset \mathcal{R}_S^T$. In the extreme, where $\mathcal{R}_S^{T+1} = \mathcal{R}_S$, we obtain the exact Bayes error,

$$e_{i^*} = \sum_{j=1}^{|\mathcal{R}_S^T|} c_{i^* j} p_j + \sum_{j=|\mathcal{R}_S^T|+1}^{|\mathcal{R}_S|} c_{i^* j} p_j. \quad (37)$$

Likewise, the exact theoretical error for any arbitrary candidate partition indexed by i (say the partition obtained by an alternative clusterer) under the assumed model can be bounded at each stage of the algorithm:

$$\sum_{j=1}^{|\mathcal{R}_S^t|} c_{ij} p_j \le e_i \le \sum_{j=1}^{|\mathcal{R}_S^t|} c_{ij} p_j + M \left(1 - \sum_{j=1}^{|\mathcal{R}_S^t|} p_j \right), \quad (38)$$

where $\sum_{j=1}^{|\mathcal{R}_S^t|} p_j$ is already being tallied and $\sum_{j=1}^{|\mathcal{R}_S^t|} c_{ij} p_j$ can also be updated at each stage. If at the end of the search where $t = T$ these bounds are not tight enough for the application, they can also be made as tight as desired by expanding the set of reference partitions further, i.e., with $\mathcal{R}_S^{T+1} \supset \mathcal{R}_S^T$, until we achieve the exact partition error when $\mathcal{R}_S^{T+1} = \mathcal{R}_S$:

$$e_i = \sum_{j=1}^{|\mathcal{R}_S^T|} c_{ij} p_j + \sum_{j=|\mathcal{R}_S^T|+1}^{|\mathcal{R}_S|} c_{ij} p_j. \quad (39)$$

V. Suboptimal Computation Reduction

Although the above methods to compute the Bayes clusterer greatly reduce computational complexity, as the point set size grows the problem will eventually become intractable. Ignoring the size of the cost function, even computing the probability of all possible partitions can itself be infeasible. Hence, we consider suboptimal algorithms to approximate the Bayes clusterer, the idea being to constrain the space of candidate and reference partitions to a subset of partitions representing a high concentration of the probability mass over all partitions, without requiring that we evaluate the probability for every partition. We implicitly assume that partitions "near" each other have close probabilities, so that we can search for high probability partitions based on "nearness" to other high probability partitions. We measure the distance between two partitions, say $\mathcal{P}^1$ and $\mathcal{P}^2$, by the minimum Hamming distance between labels inducing the partitions or, equivalently, the scaled natural partition cost, $\eta(S) c_S(\mathcal{P}^1, \mathcal{P}^2)$, which we have shown to be a valid metric. We loosely refer to this as the Hamming distance between partitions. The maximum distance between any two partitions is finite and we assume the maximum distance is Mn, where M is some positive constant.

We will outline several suboptimal clustering algorithms in Section IX. All of these are essentially based on two algorithms. Algorithm 1 implements a greedy search for the highest probability partition by evaluating the scaled probability of a seed partition (typically partition probabilities can be directly computed only up to an unknown normalization factor), evaluating

Algorithm 1: Search for the Maximum Probability Partition

Data: $\mathcal{P}^0$ = seed partition
k = Hamming distance
Result: $\mathcal{P}$ = local maximum probability partition
$i = 0$;
p_0 = scaled probability of $\mathcal{P}^0$;
repeat
  $i = i + 1$;
  $\{\mathcal{Q}^1, \ldots, \mathcal{Q}^s\}$ = Algorithm 2$(\mathcal{P}^{i-1}, k)$;
  for $j = 1$ *to* s **do**
    q_j = scaled probability of $\mathcal{Q}^j$;
  end
  $j^* = \arg\max(q_1, q_2, \ldots, q_s)$;
  if $q_{j^*} > p_{i-1}$ **then**
    $\mathcal{P}^i = \mathcal{Q}^{j^*}$;
    $p_i = q_{j^*}$;
  end
until $q_{j^*} \leq p_{i-1}$;
$\mathcal{P} = \mathcal{P}^{i-1}$;

Algorithm 2: Find the Set of Partitions in a Closed Ball of Radius h Centered on a Seed Partition $\mathcal{P}$

Data: $\mathcal{P}$ = seed reference partition
h = maximum Hamming distance
Result: $\mathcal{R}_S$ = set of unique partitions with Hamming distance at most h from $\mathcal{P}$
$\mathcal{R}_S = \{\}$;
for $j = 1$ *to* h **do**
  $\mathcal{X}_S$ = set of all partitions of the form $\mathcal{P}$ with j points flipped;
  $\mathcal{R}_S = \{\mathcal{R}_S, \mathcal{X}_S\}$;
end

a scaled probability of all partitions within a closed ball of radius k centered on the seed partition for a given fixed integer k (call these partitions $\mathcal{Q}^1, \ldots, \mathcal{Q}^s$), identifying the partition among these, including the seed, with highest probability, and repeating using the highest probability partition as the new seed until there is no improvement. This procedure is guaranteed to converge to a local maximum in a finite bounded number of steps, since the search is over a finite number of partitions and the probabilities can only increase. It is also guaranteed to find the maximum probability partition when $k = Mn$ because a closed ball of this radius contains the whole space. The entire procedure may be repeated a number of times with different seeds, and a final partition with highest scaled probability selected. Algorithm 2 finds the set of partitions in a closed ball centered on a given seed partition.

When the Bayes error is not high, most of the probability mass over partitions tends to be concentrated in a neighborhood of the maximum probability partition. Hence, we will typically choose a set of reference partitions using Algorithm 2 with a fixed integer radius h and some available seed partition, e.g, the true maximum probability partition or an approximation found from Algorithm 1. The constant h controls the complexity of the Bayes partition search, and when Algorithm 1 is used to select a seed, the constant k controls the complexity of the maximal probability partition search. If $h = Mn$ then the space of reference partitions includes all partitions, and if $h = 0$ then the only reference partition is the seed partition.

Given a suboptimal set of reference partitions and their scaled probabilities, we take a naïve approach to normalizing them by assuming that non-reference partitions have probability zero. We also initialize the space of candidate partitions to be either the set of all partitions, or the same as the set of reference partitions. Given a set of reference partitions, normalized partition probabilities, and a set of candidate partitions, we approximate the optimal Bayes partition, where all methods in Sections IV-A and IV-B to improve computational complexity may be applied.

One can approximate the Bayes partition error of any candidate partition $\mathcal{P}^i$ using only the probabilities that have been computed by $\widehat{e}'_i = \sum_{j=1}^{|\mathcal{R}'_S|} c_{ij} \widehat{p}_j(\mathcal{R}'_S)$, where $\mathcal{R}'_S$ is a set of reference partitions and $\widehat{p}_j(\mathcal{R}'_S)$ is the probability of reference partition j normalized over only the partitions in $\mathcal{R}'_S$. This tends to be a low biased approximation if $\mathcal{P}^i \in \mathcal{R}'_S$, primarily due to the naïve scaling method. In fact, the reported Bayes error is upper bounded by the maximum cost between partitions in the candidate and reference set, which is upper bounded by $2h/n$ when using Algorithm 2 with distance h. Put another way, we must choose $h \geq ne^*/2$, where e^* is the Bayes partitioning error, for it to even be possible to report the Bayes error correctly. Thus, problems with higher Bayes error require higher computational complexity for accurate error estimation. To approximate the partition error for an arbitrary partition we recommend computing the probabilities for a larger collection of reference partitions, $\mathcal{R}''_S \supset \mathcal{R}'_S$, than used for finding the Bayes partition. Then, under our naïve scaling rule, $\widehat{p}_j(\mathcal{R}''_S) = C\widehat{p}_j(\mathcal{R}'_S)$ for all $j \in \{1, \ldots, |\mathcal{R}'_S|\}$ for some constant $C < 1$ and we have the approximate error

$$\widehat{e}''_i = \sum_{j=1}^{|\mathcal{R}''_S|} c_{ij} \widehat{p}_j(\mathcal{R}''_S) = C\widehat{e}'_i + \sum_{j=|\mathcal{R}'_S|+1}^{|\mathcal{R}''_S|} c_{ij} \widehat{p}_j(\mathcal{R}''_S). \quad (40)$$

VI. Partition Probabilities for Gaussian Models

To this point, we have made no assumptions on the RLPP, except to say that the probabilities $P_S(\phi_S)$ (in the case of the optimal label operator) or $P_S(\mathcal{P}_S)$ (the p_j in the case of the optimal cluster operator) are available. To evaluate either of these probabilities requires finding $P_S(\phi_S)$ for an arbitrary label function under the assumed RLPP model. For $\phi_S \in L^S$,

$$P_S(\phi_S) = P(\phi_S|S) \propto P(\phi_S) f(S|\phi_S). \quad (41)$$

We assume a RLPP where $P(\phi_S)$ is a known prior probability on labels, which does not depend on the specific points in S. For instance, under random sampling with n points and l possible labels, $P(\phi_S)$ is uniform over all l^n possible label functions. Under stratified sampling, label functions with the correct number of points in each class are equally likely, with all other label functions having zero prior probability.

We further assume that given a label function, ϕ_S, and a collection of distribution parameters, $\rho = \{\rho_1, \ldots, \rho_l\}$, where ρ_i is a parameter associated with label i, each point $\mathbf{x} \in S$ having label $i = \phi_S(\mathbf{x})$ is independently drawn from a label-i-conditional distribution, $f_i(\mathbf{x}; \rho_i)$. We also assume the ρ_i are *a priori*

independent from each other and from ϕ_S with prior density $f(\rho_i)$. Hence,

$$P_S(\phi_S) \propto P(\phi_S) \int f(S|\phi_S,\rho) f(\rho) d\rho = P(\phi_S) \prod_{\substack{i=1 \\ n_i \geq 1}}^{l} \int \left(\prod_{\mathbf{x} \in S_i} f_i(\mathbf{x};\rho_i) \right) f(\rho_i) d\rho_i, \quad (42)$$

where $S_i = \{\mathbf{x} \in S : \phi_S(\mathbf{x}) = i\}$ is the set of all points in S assigned label i by ϕ_S and n_i is the cardinality of S_i.

We next introduce three Gaussian models. In the next Section we extend these to generalized RLPPs with improper distributions on model parameters, and show invariance of generalized RLPPs to linear transformations of the space.

A. Known Means and Covariances

Under this model, point sets are generated by l Gaussian distributions with known parameters. In particular, for each $i \in \{1,\ldots,l\}$ let $\rho_i = \{\mu_i, \Sigma_i\}$, where μ_i is a length d real vector and Σ_i is a symmetric positive definite $d \times d$ matrix. Then a point $\mathbf{x} \in S$ with label i has a Gaussian distribution with mean μ_i and covariance Σ_i. That is, $f_i(\mathbf{x};\rho_i) \sim \mathcal{N}(\mu_i, \Sigma_i)$. For $n_i \geq 1$,

$$\prod_{\mathbf{x} \in S_i} f_i(\mathbf{x};\rho_i) = \frac{1}{(2\pi)^{\frac{dn_i}{2}} |\Sigma_i|^{\frac{n_i}{2}}} \times \exp\left(-\frac{1}{2} \sum_{\mathbf{x} \in S_i} (\mathbf{x}-\mu_i)^T \Sigma_i^{-1} (\mathbf{x}-\mu_i) \right), \quad (43)$$

where $|\bullet|$ denotes a determinant. If $n_i \geq 2$, let $\widehat{\mu}_i$ and $\widehat{\Sigma}_i$ be the sample mean and covariance of points in S_i. Then,

$$\sum_{\mathbf{x} \in S_i} (\mathbf{x}-\mu_i)^T \Sigma_i^{-1} (\mathbf{x}-\mu_i) = \mathrm{tr}\left(\Phi_i^* \Sigma_i^{-1}\right), \quad (44)$$

where $\mathrm{tr}(\bullet)$ denotes a trace and

$$\Phi_i^* = (n_i - 1)\widehat{\Sigma}_i + n_i(\mu_i - \widehat{\mu}_i)(\mu_i - \widehat{\mu}_i)^T.$$

If $n_i = 1$ then (44) holds with $\Phi_i^* = (\mu_i - \widehat{\mu}_i)(\mu_i - \widehat{\mu}_i)^T$. Hence,

$$\prod_{\mathbf{x} \in S_i} f_i(\mathbf{x};\rho_i) = \frac{1}{(2\pi)^{\frac{dn_i}{2}} |\Sigma_i|^{\frac{n_i}{2}}} \exp\left(-\frac{\mathrm{tr}\left(\Phi_i^* \Sigma_i^{-1}\right)}{2} \right). \quad (45)$$

Since the ρ_i are fixed, $f(\rho_i)$ is a point mass at $\rho_i = \{\mu_i, \Sigma_i\}$. Hence, from (42),

$$P_S(\phi_S) \propto P(\phi_S) \prod_{\substack{i=1 \\ n_i \geq 1}}^{l} \frac{1}{(2\pi)^{\frac{dn_i}{2}} |\Sigma_i|^{\frac{n_i}{2}}} \exp\left(-\frac{1}{2} \mathrm{tr}\left(\Phi_i^* \Sigma_i^{-1}\right) \right) \propto P(\phi_S) \left(\prod_{i=1}^{l} |\Sigma_i|^{n_i} \right)^{-\frac{1}{2}} \exp\left(-\frac{1}{2} \sum_{i=1}^{l} \mathrm{tr}\left(\Phi_i^* \Sigma_i^{-1}\right) \right),$$

where by convention we set Φ_i^* to a $d \times d$ zero matrix if $n_i = 0$. The Bayes clusterer is thus given by (19) and (17) with $P_S(\phi_S)$ provided above. If $P(\phi_S) \propto 1$ for all ϕ_S considered, and we either have that all n_i are fixed for all ϕ_S considered or equal covariances for each label ($\Sigma_1 = \Sigma_2 = \ldots = \Sigma_l$), this may be simplified further:

$$P_S(\phi_S) \propto \exp\left(-\frac{1}{2} \sum_{i=1}^{l} \mathrm{tr}\left(\Phi_i^* \Sigma_i^{-1}\right) \right). \quad (46)$$

The probability of a label function is larger if the sample means, $\widehat{\mu}_i$, are close to the true means, μ_i (so Φ_i^* is smaller), and if the shape of the sample covariances, $\widehat{\Sigma}_i$, are close to the known covariances, Σ_i, in the sense that $\widehat{\Sigma}_i \approx \alpha\Sigma_i$ (so $\mathrm{tr}(\Phi_i^* \Sigma_i^{-1})$ is smaller). This probability is also larger if $\widehat{\Sigma}_i$ is smaller, in the sense that $\widehat{\Sigma}_i \approx \alpha\Sigma_i$ is better for smaller α. Thus "tighter" clusters have higher probability.

B. Gaussian Means and Known Covariances

In this model, point sets are generated by distributions with random means and known covariances. For each $i \in \{1,\ldots,l\}$, we define two fixed hyperparameters: a real number $\nu_i > 0$ and a length d real vector $\mathbf{m}_i$. We define parameters $\rho_i = \{\mu_i, \Sigma_i\}$, where the Σ_i are fixed symmetric positive definite $d \times d$ matrices and the μ_i have independent Gaussian distributions with mean $\mathbf{m}_i$ and covariance $\frac{1}{\nu_i}\Sigma_i$. A point $\mathbf{x} \in S$ having label i is drawn from a Gaussian distribution with mean μ_i and covariance Σ_i, so that $f_i(\mathbf{x};\rho_i) \sim \mathcal{N}(\mu_i, \Sigma_i)$. Define

$$\mathcal{L}_i(S_i) = \int \left(\prod_{\mathbf{x} \in S_i} f_i(\mathbf{x};\rho_i) \right) f(\rho_i) d\rho_i$$

when $n_i \geq 1$. For a proper distribution on the μ_i where $\nu_i > 0$ for all i, applying (45) for $n_i \geq 2$,

$$\mathcal{L}_i(S_i) = \frac{|\nu_i|^{\frac{d}{2}}}{(2\pi)^{\frac{d(n_i+1)}{2}} |\Sigma_i|^{\frac{(n_i+1)}{2}}} \exp\left(-\frac{1}{2} \mathrm{tr}\left((n_i - 1)\widehat{\Sigma}_i \Sigma_i^{-1} \right) \right) \times \int \exp\left(-\frac{n_i}{2} (\mu_i - \widehat{\mu}_i)^T \Sigma_i^{-1} (\mu_i - \widehat{\mu}_i) - \frac{\nu_i}{2} (\mu_i - \mathbf{m}_i)^T \Sigma_i^{-1} (\mu_i - \mathbf{m}_i) \right) d\mu_i.$$

Applying the fact that

$$n_i(\mu_i - \widehat{\mu}_i)^T \Sigma_i^{-1} (\mu_i - \widehat{\mu}_i) + \nu_i (\mu_i - \mathbf{m}_i)^T \Sigma_i^{-1} (\mu_i - \mathbf{m}_i) = (n_i + \nu_i) \left(\mu_i - \frac{n_i \widehat{\mu}_i + \nu_i \mathbf{m}_i}{n_i + \nu_i} \right)^T \Sigma_i^{-1} \left(\mu_i - \frac{n_i \widehat{\mu}_i + \nu_i \mathbf{m}_i}{n_i + \nu_i} \right) + \frac{n_i \nu_i}{n_i + \nu_i} (\widehat{\mu}_i - \mathbf{m}_i)^T \Sigma_i^{-1} (\widehat{\mu}_i - \mathbf{m}_i),$$

and integrating out a Gaussian distribution on μ_i,

$$\mathcal{L}_i(S_i) = \frac{|\nu_i|^{\frac{d}{2}}}{|n_i + \nu_i|^{\frac{d}{2}} (2\pi)^{\frac{dn_i}{2}} |\Sigma_i|^{\frac{n_i}{2}}} \times \exp\left(-\frac{1}{2} \mathrm{tr}\left((n_i - 1)\widehat{\Sigma}_i \Sigma_i^{-1} \right) \right) \times \exp\left(-\frac{1}{2} \cdot \frac{n_i \nu_i}{n_i + \nu_i} (\widehat{\mu}_i - \mathbf{m}_i)^T \Sigma_i^{-1} (\widehat{\mu}_i - \mathbf{m}_i) \right) = \frac{|\nu_i|^{\frac{d}{2}}}{|n_i + \nu_i|^{\frac{d}{2}} (2\pi)^{\frac{dn_i}{2}} |\Sigma_i|^{\frac{n_i}{2}}} \exp\left(-\frac{1}{2} \mathrm{tr}\left(\Psi_i^* \Sigma_i^{-1} \right) \right), \quad (47)$$

where

$$\Psi_i^* = (n_i - 1)\widehat{\Sigma}_i + \frac{n_i \nu_i}{n_i + \nu_i}(\widehat{\mu}_i - \mathbf{m}_i)(\widehat{\mu}_i - \mathbf{m}_i)^T. \quad (48)$$

In the case where $n_i = 1$, (47) holds with $\Psi_i^* = \frac{\nu_i}{\nu_i+1}(\widehat{\mu}_i - \mathbf{m}_i)(\widehat{\mu}_i - \mathbf{m}_i)^T$. Finally, from (42) and (47),

$$P_S(\phi_S) \propto P(\phi_S) \left(\prod_{i=1}^{l} \frac{|\nu_i|^d}{|n_i + \nu_i|^d |\Sigma_i|^{n_i}} \right)^{\frac{1}{2}} \times \exp\left(-\frac{1}{2} \sum_{i=1}^{l} \mathrm{tr}\left(\Psi_i^* \Sigma_i^{-1} \right) \right), \quad (49)$$

where by convention we set Ψ_i^* to a $d \times d$ zero matrix if $n_i = 0$. The Bayes clusterer is thus given by (19) and (17) with $P_S(\phi_S)$ provided above.

If $P(\phi_S) \propto 1$ for all ϕ_S considered and all n_i are fixed for all ϕ_S considered, then

$$P_S(\phi_S) \propto \exp\left(-\frac{1}{2} \sum_{i=1}^{l} \mathrm{tr}\left(\Psi_i^* \Sigma_i^{-1} \right) \right). \quad (50)$$

A label function probability is larger if the sample means, $\widehat{\mu}_i$, are close to the expected means, $\mathbf{m}_i$, and if the sample covariances, $\widehat{\Sigma}_i$, are "tighter" with a shape close to the known covariances, Σ_i, in the sense that $\widehat{\Sigma}_i \approx \alpha \Sigma_i$ for small α.

C. Normal-Inverse-Wishart Means and Covariances

In this model, the means and variances of each class-conditional distribution are random. For each $i \in \{1, \ldots, l\}$, define a length d real vector $\mathbf{m}_i$, real numbers $\nu_i > 0$ and $\kappa_i > d-1$, and a symmetric positive definite matrix Ψ_i as hyperparameters. Define parameters $\rho_i = \{\mu_i, \Sigma_i\}$ having independent normal-inverse-Wishart distributions. That is, Σ_i is inverse-Wishart,

$$f(\Sigma_i) = \frac{|\Psi_i|^{\frac{\kappa_i}{2}}}{2^{\frac{\kappa_i d}{2}} \Gamma_d\left(\frac{\kappa_i}{2}\right)} |\Sigma_i|^{-\frac{\kappa_i + d + 1}{2}} \exp\left(-\frac{1}{2} \mathrm{tr}\left(\Psi_i \Sigma_i^{-1} \right) \right),$$

where Γ_d is the multivariate Gamma function and, given Σ_i, the μ_i have Gaussian distributions with mean $\mathbf{m}_i$ and covariance $\frac{1}{\nu_i}\Sigma_i$. Note if $\kappa_i > d+1$, the mean of Σ_i is $\frac{1}{\kappa_i - d - 1}\Psi_i$. With ρ_i fixed, each $\mathbf{x} \in S$, having label i, is Gaussian with $f_i(\mathbf{x}; \rho_i) \sim \mathcal{N}(\mu_i, \Sigma_i)$. ρ_i has a proper density, and when $n_i \geq 1$ we define

$$\begin{aligned} \mathcal{L}_i(S_i) &= \int \int \left(\prod_{\mathbf{x} \in S_i} f_i(\mathbf{x}; \mu_i, \Sigma_i) \right) f(\mu_i | \Sigma_i) d\mu_i f(\Sigma_i) d\Sigma_i \\ &= \int \frac{|\nu_i|^{\frac{d}{2}}}{|n_i + \nu_i|^{\frac{d}{2}} (2\pi)^{\frac{d n_i}{2}} |\Sigma_i|^{\frac{n_i}{2}}} \\ &\quad \times \exp\left(-\frac{1}{2} \mathrm{tr}\left(\Psi_i^* \Sigma_i^{-1} \right) \right) f(\Sigma_i) d\Sigma_i, \end{aligned} \quad (51)$$

where in the last line we use (47) for the previous model with fixed covariances, and Ψ_i^* is defined as in (48) if $n_i \geq 1$. Proceeding,

$$\begin{aligned} \mathcal{L}_i(S_i) &= \int \frac{|\nu_i|^{\frac{d}{2}}}{|n_i + \nu_i|^{\frac{d}{2}} (2\pi)^{\frac{d n_i}{2}} |\Sigma_i|^{\frac{n_i}{2}}} \exp\left(-\frac{1}{2} \mathrm{tr}\left(\Psi_i^* \Sigma_i^{-1} \right) \right) \\ &\quad \times \frac{|\Psi_i|^{\frac{\kappa_i}{2}}}{2^{\frac{\kappa_i d}{2}} \Gamma_d\left(\frac{\kappa_i}{2}\right)} |\Sigma_i|^{-\frac{\kappa_i + d + 1}{2}} \\ &\quad \times \exp\left(-\frac{1}{2} \mathrm{tr}\left(\Psi_i \Sigma_i^{-1} \right) \right) d\Sigma_i \\ &= \frac{|\nu_i|^{\frac{d}{2}}}{|n_i + \nu_i|^{\frac{d}{2}} (2\pi)^{\frac{d n_i}{2}}} \times \frac{|\Psi_i|^{\frac{\kappa_i}{2}}}{2^{\frac{\kappa_i d}{2}} \Gamma_d\left(\frac{\kappa_i}{2}\right)} \\ &\quad \times \int |\Sigma_i|^{-\frac{\kappa_i + n_i + d + 1}{2}} \\ &\quad \times \exp\left(-\frac{1}{2} \mathrm{tr}\left(\left(\Psi_i + \Psi_i^* \right) \Sigma_i^{-1} \right) \right) d\Sigma_i. \end{aligned}$$

This is essentially an inverse-Wishart integral with updated parameters $\kappa_i + n_i$ and $\Psi_i + \Psi_i^*$. Thus,

$$\mathcal{L}_i(S_i) = \frac{|\nu_i|^{\frac{d}{2}}}{|n_i + \nu_i|^{\frac{d}{2}} (2\pi)^{\frac{d n_i}{2}}} \frac{|\Psi_i|^{\frac{\kappa_i}{2}}}{2^{\frac{\kappa_i d}{2}} \Gamma_d\left(\frac{\kappa_i}{2}\right)} \frac{2^{\frac{(\kappa_i + n_i) d}{2}} \Gamma_d\left(\frac{\kappa_i + n_i}{2}\right)}{|\Psi_i + \Psi_i^*|^{\frac{\kappa_i + n_i}{2}}}. \quad (52)$$

From (42) and (52), after scaling across all ϕ_S, we have

$$P_S(\phi_S) \propto P(\phi_S) \prod_{i=1}^{l} \frac{|\nu_i|^{\frac{d}{2}} \Gamma_d\left(\frac{\kappa_i + n_i}{2}\right) |\Psi_i|^{\frac{\kappa_i}{2}}}{|n_i + \nu_i|^{\frac{d}{2}} \Gamma_d\left(\frac{\kappa_i}{2}\right) |\Psi_i + \Psi_i^*|^{\frac{\kappa_i + n_i}{2}}}, \quad (53)$$

where by convention we set Ψ_i^* to a $d \times d$ zero matrix if $n_i = 0$. The Bayes clusterer is again given by (19) and (17) with $P_S(\phi_S)$ provided above. If $P(\phi_S) \propto 1$ for all ϕ_S considered and we have all n_i are fixed for all ϕ_S considered, then

$$P_S(\phi_S) \propto \prod_{i=1}^{l} |\Psi_i + \Psi_i^*|^{-\frac{\kappa_i + n_i}{2}}. \quad (54)$$

VII. Generalized Gaussian Models With Improper Distributions on Model Parameters

In a Bayesian setting, it can be useful to assume Gaussianity with a "non-informative" prior on the parameters. We take a similar approach to formulate generalized Gaussian RLPPs, where the means and/or covariances are random and governed by improper distributions. To formalize this, the random point process, Ξ, now maps from a measure space, $(\Omega, \mathcal{A}, \upsilon)$, to $[\mathbf{N}; \mathcal{N}]$, where $\mathcal{A}$ is a σ-algebra and υ is a σ-finite measure. ν is now a measure on Ξ determined by $\nu(Y) = \upsilon(\Xi^{-1}(Y))$ for $Y \in \mathcal{N}$. Strictly speaking, this RLPP is not realizable since it is not based on a probability space. That being said, it may still be possible to find a meaningful probability mass function over all possible labelings, conditioned on a given point set S. If used to find a Bayes clusterer, we call the result a *generalized Bayes cluster operator*. One consequence is that (42) must be modified such that $P_S(\phi_S)$ is proportional to:

$$P(\phi_S) \prod_{\substack{i=1 \\ n_i \geq 1}}^{l} \int \left(\prod_{\mathbf{x} \in S_i} f_i(\mathbf{x}; \rho_i) \right) f(\rho_i) d\rho_i \prod_{\substack{i=1 \\ n_i = 0}}^{l} \int f(\rho_i) d\rho_i.$$

If $n_i = 0$ for any i, then the last product above is not defined. Hence, the generalized theory requires that we only consider label functions that assign at least one point to every label, and to do so we set $P(\phi_S) = 0$ otherwise.

A. Gaussian Means and Known Covariances

Returning to our Gaussian model with unknown means and known covariances, suppose we set $\nu_i = 0$ for all i, so that the μ_i have improper distributions of the form $f(\rho_i) = f(\mu_i) \propto |\Sigma_i|^{-\frac{1}{2}}$, where Σ_i is known. Although these distributions are improper, it is possible to carry out the derivation of $\mathcal{L}_i(S_i) = \int(\prod_{\mathbf{x}\in S_i} f_i(\mathbf{x};\rho_i))f(\rho_i)d\rho_i$ for each i where $n_i \geq 1$, using $|\Sigma_i|^{-\frac{1}{2}}$ in place of $f(\rho_i)$. The steps are exactly the same and the result is essentially the same:

$$\mathcal{L}_i(S_i) \propto \frac{1}{|n_i|^{\frac{d}{2}}(2\pi)^{\frac{dn_i}{2}}|\Sigma_i|^{\frac{n_i}{2}}} \exp\left(-\frac{1}{2}\mathrm{tr}\left(\Psi_i^*\Sigma_i^{-1}\right)\right), \quad (55)$$

where $\Psi_i^* = (n_i - 1)\widehat{\Sigma}_i$ for $n_i \geq 2$ and $\Psi_i^* = 0$ for $n_i = 1$. As long as ϕ_S is such that $n_i \geq 1$ for all i, $P_S(\phi_S)$ is derived in the usual way from (55) and (42), and, analogously to (49) under proper priors, may be simplified to

$$P_S(\phi_S) \propto P(\phi_S)\left(\prod_{i=1}^{l}|n_i|^d|\Sigma_i|^{n_i}\right)^{-\frac{1}{2}} \times \exp\left(-\frac{1}{2}\sum_{i=1}^{l}\mathrm{tr}\left(\Psi_i^*\Sigma_i^{-1}\right)\right). \quad (56)$$

$P_S(\phi_S)$ is normalizable and provides a probability mass function that can be used in a generalized Bayes cluster operator.

The standard k-means objective function is optimal for a 0-1 cost function (i.e., maximum *a posteriori* clustering), a Gaussian model with unknown means having an improper flat distribution, identical scaled identity covariances (the scaling factor need not be specified), and a specific form for $P(\phi_S)$ [22]. Thus, the new theory reveals implicit assumptions in applying the k-means objective function, and provides a framework to compute the exact partition error (with any cost function) under the given assumptions.

B. Normal-Inverse-Wishart Means and Covariances

Next consider our Gaussian model with unknown means and unknown covariances. If we set $\nu_i = 0$ for all i, as before the μ_i given Σ_i have improper distributions of the form $f(\mu_i|\Sigma_i) \propto |\Sigma_i|^{-\frac{1}{2}}$, and just as before substituting $|\Sigma_i|^{-\frac{1}{2}}$ in place of $f(\mu_i|\Sigma_i)$ in the derivation of $\int(\prod_{\mathbf{x}\in S_i} f_i(\mathbf{x};\rho_i))f(\mu_i|\Sigma_i)d\mu_i$ for each i where $n_i \geq 1$ gives (55). For the covariance we require $\kappa_i \in \mathbb{R}$ and $\Psi_i = 0$ for all i, resulting in an improper distribution:

$$f(\Sigma_i) \propto \frac{|\Sigma_i|^{-\frac{\kappa_i+d+1}{2}}}{2^{\frac{\kappa_i d}{2}}\Gamma_d\left(\frac{\kappa_i}{2}\right)}. \quad (57)$$

Substituting the right hand side above in place of $f(\Sigma_i)$ and the result for $\int(\prod_{\mathbf{x}\in S_i} f_i(\mathbf{x};\rho_i))f(\mu_i|\Sigma_i)d\mu_i$ into $\mathcal{L}_i(S_i) = \int(\prod_{\mathbf{x}\in S_i} f_i(\mathbf{x};\rho_i))f(\rho_i)d\rho_i$, we again obtain essentially the same result:

$$\mathcal{L}_i(S_i) \propto \frac{\Gamma_d\left(\frac{\kappa_i+n_i}{2}\right)}{\pi^{\frac{dn_i}{2}}\Gamma_d\left(\frac{\kappa_i}{2}\right)|n_i|^{\frac{d}{2}}|\Psi_i^*|^{\frac{\kappa_i+n_i}{2}}}, \quad (58)$$

where $\Psi_i^* = (n_i - 1)\widehat{\Sigma}_i$ for $n_i \geq 2$ and $\Psi_i^* = 0$ for $n_i = 1$. For any labeling in which $n_i \geq 1$ for all i, $P_S(\phi_S)$ is derived in the usual way from (58) and (42), and, analogously to (53) under proper priors, may be simplified to

$$P_S(\phi_S) \propto P(\phi_S)\prod_{i=1}^{l}\frac{\Gamma_d\left(\frac{\kappa_i+n_i}{2}\right)}{|n_i|^{\frac{d}{2}}|\Psi_i^*|^{\frac{\kappa_i+n_i}{2}}}. \quad (59)$$

C. Invariance to Affine Transformations

Aside from avoiding informative assumptions on the means and/or covariances, generalized Bayes clusterers enjoy several invariance properties for linear transformations of the sample space. Suppose we apply an affine transformation to a point set S, resulting in the point set $\overline{S} = A(S - \mathbf{x}_0)$, where $\mathbf{x}_0 \in \mathbb{R}^d$ and A is a $d \times d$ matrix. Let ϕ_S be an arbitrary label function on S and $\phi_{\overline{S}}$ be the corresponding label function on $\overline{S}$, where $\phi_{\overline{S}}(\overline{\mathbf{x}}) = \phi_S(\mathbf{x})$ for $\overline{\mathbf{x}} = A(\mathbf{x} - \mathbf{x}_0) \in \overline{S}$. In both generalized Gaussian models, $\overline{\Psi}_i^* = \sum_{\overline{\mathbf{x}}\in\overline{S}_i}(\overline{\mathbf{x}} - \overline{\mu}_i)(\overline{\mathbf{x}} - \overline{\mu}_i)^T$, where $\overline{\mu}_i$ is the sample mean of points in $\overline{S}_i$. One can easily verify $\overline{\mu}_i = A(\widehat{\mu}_i - \mathbf{x}_0)$, and $\overline{\Psi}_i^* = A\Psi_i^*A^T$.

For a Gaussian model with unknown means and known covariances, from (56) observe that $P_{\overline{S}}(\phi_{\overline{S}}) = P_S(\phi_S)$ if

$$\mathrm{tr}\left(\overline{\Psi}_i^*\Sigma_i^{-1}\right) \equiv \mathrm{tr}\left(A\Psi_i^*A^T\Sigma_i^{-1}\right) = \mathrm{tr}\left(\Psi_i^*\Sigma_i^{-1}\right).$$

First, suppose $A = I_d$, where I_d is the $d \times d$ identity matrix. In this case, the preceding equation holds trivially. Hence the generalized Bayes clusterer for unknown means and known covariances is invariant to translations of the point set. Next, suppose A is a rotation matrix, where $A^TA = I_d$, but the model assumes $\Sigma_i = \sigma_i^2 I_d$ for scalars σ_i^2 for all $i \in L$. Then

$$\mathrm{tr}\left(\overline{\Psi}_i^*\Sigma_i^{-1}\right) = \frac{1}{\sigma_i^2}\mathrm{tr}\left(\Psi_i^*\right) = \mathrm{tr}\left(\Psi_i^*\Sigma_i^{-1}\right).$$

Thus, the generalized Bayes clusterer assuming known scaled identity covariances is invariant to translations and rotations.

For the Gaussian model with unknown means and covariances, examining (59) it is clear that $P_{\overline{S}}(\phi_{\overline{S}}) = P_S(\phi_S)$ if

$$\left|\overline{\Psi}_i^*\right| \equiv \left|A\Psi_i^*A^T\right| \propto \left|\Psi_i^*\right|, \quad (60)$$

which holds for any matrix A such that $|A| \neq 0$. Thus, the generalized Bayes clusterer for unknown means and covariances is invariant to translations, rotations, scaling, and any other one-to-one linear transformation of the sample space.

VIII. Clustering is Distinct From Labeling

Earlier we noted that an optimal label operator does not necessarily induce an optimal clustering operator. To illustrate, assume that for any partition $\mathcal{P}_S$ with non-zero probability, $P(\phi_S)$ is constant for all ϕ_S in $G_{\mathcal{P}_S}$, that is, all label functions inducing $\mathcal{P}_S$ are equally likely. First consider a generalized Gaussian model with unknown means and known covariances satisfying $\Sigma_i = \Sigma$ for all $i \in L$. Then for any non-zero probability partition, $\mathcal{P}_S$, and for every label function $\phi_S \in G_{\mathcal{P}_S}$, from (56) with $\Psi_i^* = (n_i - 1)\widehat{\Sigma}_i$,

$$P_S(\phi_S) \propto \exp\left(-\frac{1}{2}\sum_{i=1}^{l}(n_i - 1)\mathrm{tr}(\widehat{\Sigma}_i\Sigma^{-1})\right)$$

where the proportionality holds over all $\phi_S \in G_{\mathcal{P}_S}$. The terms in the sum, and hence the $P_S(\phi_S)$, are equivalent for every ϕ_S in $G_{\mathcal{P}_S}$. Recall from (17) that $P_S(\mathcal{P}_S) = \sum_{\phi_S \in G_{\mathcal{P}_S}} P_S(\phi_S)$, so that $\sum_{\substack{\phi_S \in G_{\mathcal{P}_S} \\ \phi_S(\mathbf{x})=i}} P_S(\phi_S) = \frac{1}{l} P_S(\mathcal{P}_S)$. Furthermore,

$$\sum_{\substack{\phi_S \in L^S \\ \phi_S(\mathbf{x})=i}} P_S(\phi_S) = \sum_{\mathcal{P}_S \in \mathcal{K}_S} \sum_{\substack{\phi_S \in G_{\mathcal{P}_S} \\ \phi_S(\mathbf{x})=i}} P_S(\phi_S) = \sum_{\mathcal{P}_S \in \mathcal{K}_S} \frac{1}{l} P_S(\mathcal{P}_S) = \frac{1}{l}.$$

In particular, this quantity is not a function of i. From (11),

$$\varepsilon_\lambda(S) = \frac{1}{\eta(S)} \sum_{\mathbf{x} \in S} \sum_{\substack{i=1 \\ i \neq \phi_{S,\lambda}(\mathbf{x})}}^{l} \frac{1}{l} = \frac{l-1}{l}.$$

This is the same labeling error achieved by random guessing.

For our generalized Gaussian model with unknown means and covariances, suppose $\kappa_i = \kappa$ for all $i \in L$. From (59) with $\Psi_i^* = (n_i - 1)\widehat{\Sigma}_i$, for each label function $\phi_S \in G_{\mathcal{P}_S}$,

$$P_S(\phi_S) \propto \prod_{i=1}^{l} \left| (n_i - 1)\widehat{\Sigma}_i \right|^{-\frac{\kappa + n_i}{2}}.$$

The terms in the product, and hence the $P_S(\phi_S)$, are equivalent for every ϕ_S in $G_{\mathcal{P}_S}$. Just as before, $\varepsilon_\lambda(S) = \frac{l-1}{l}$, so that even optimal labeling can do no better than random guessing.

IX. Experimental Results

This section presents performance for the optimal Bayes clusterer and several suboptimal algorithms relative to classical algorithms. All results are obtained from synthetic data generated under the proposed Gaussian RLPP models.

A. Algorithms

We implement several clustering algorithms: (**Optimal**) the exact optimal solution for the given RLPP—only used in small point sets; (**Subopt. Pmax**) Algorithm 2 where the seed is the true maximum probability partition, $\mathcal{P}_{max}$; (**Subopt. Pseed**) Algorithm 2 where the seed partition, $\mathcal{P}_{seed}$, is found from Algorithm 1; (**FCM**) fuzzy c-means; (**KM**) k-means; (**Hier. (Si)**) hierarchical clustering with single linkage; (**Hier. (Co)**) hierarchical clustering with complete linkage; (**Random**) randomly clustered points; and (**Classifier**) the clusterer induced by the optimal linear classifier—only used in models with no uncertainty in the mean and covariance of the distributions that compose the RLPP. We will also introduce two suboptimal algorithms tailored for very large point sets in Section IX-E.

We make a few simplifying modeling assumptions to reduce computational burden in the optimal, Pmax and Pseed algorithms. First, we restrict our analysis to only $l = 2$ clusters. For the optimal clusterer, the number of candidate partitions for a given point set S with n points is 2^{n-1}. We also assume that we know the cluster sizes. If there are n_1 and n_2 points in the clusters, where labels can be switched, the number of reference partitions is thus reduced to $\frac{1}{2}\frac{n!}{n_1!n_2!}$ if $n_1 = n_2$ or to $\frac{n!}{n_1!n_2!}$ otherwise.

The suboptimal Pmax algorithm constrains the set of reference partitions to a subset over which most of the probability mass is concentrated using Algorithm 2 with seed partition $\mathcal{P}_{max}$ and threshold h. We further constrain the reference set to partitions assigning the correct number of points to each cluster and apply no constraints to the set of candidate partitions. When h reaches the maximum value for a given clustering problem, Pmax turns into the optimal algorithm.

Suboptimal Pmax still requires computing the probabilities for all reference partitions in order to find $\mathcal{P}_{max}$. To further reduce computational cost, Pseed applies two changes:

1) We use Algorithm 1 with $k = 1$ and repeat the algorithm 5 times with random seeds to iteratively find a partition $\mathcal{P}_{seed}$ close to $\mathcal{P}_{max}$. We may find a local maximum.
2) We apply Algorithm 2 with seed $\mathcal{P}_{seed}$ and threshold h. The same set is used for both the reference and candidate sets, and is constrained to have correct cluster sizes.

Even with the maximum value for h, Pseed might not return the optimal partition since candidate partitions have been constrained to have the correct proportion of points.

Where applicable, Optimal, Pmax and Pseed use all bounds in Fig. 2 and Section IV-B to reduce the set of candidate and reference partitions, respectively. We set partition $\mathcal{P}^1$ (Figs. 1(a) and 1(b)) to the highest probability partition available, $\mathcal{P}_{max}$ or $\mathcal{P}_{seed}$. The complexity of the search is often significantly reduced with these bounds, particularly in models with low or moderate Bayes errors.

B. Description of Experiments

We consider 2-dimensional RLPPs based on the Gaussian mixture models in Section VI with $l = 2$ clusters: (**Model 0**) fixed known means and covariances; (**Model 1**) Gaussian means and fixed known covariances; and (**Model 2**) Gaussian means and inverse-Wishart covariances. Experiments are divided into three steps: point set generation, clustering, and performance evaluation.

In point set generation, we first draw two random distributions based on the RLPP model, and then draw a testing set, S, from these distributions. We fix *a priori* the number of points in each cluster, n_1 and n_2. We will evaluate the performance of optimal clustering and the optimal clustering error on small point sets of size $n = n_1 + n_2 = 20$, the performance of suboptimal clustering and clustering error estimation on moderate point sets of size $n = 70$ points, and the performance of clustering on very large point sets of size up to $n = 10000$. We will consider several equal and unequal cluster size settings. For unequal sized clusters, where $n_1 \neq n_2$, by convention half of the point sets are generated with n_1 points from distribution 1 and n_2 from distribution 2, and vice-versa for the other half. In all cases, each testing set is accompanied by a vector of labels indicating the distribution from which each point comes.

We apply the algorithms listed in Section IX-A to each testing set. A two-cluster partition is found by assigning a vector of labels based on the clustering algorithm. Only the optimal, suboptimal Pmax, suboptimal Pseed, and clusterer induced by the optimal linear classifier assume a RLPP model; classical clustering algorithms are model-free.

For each set, we compare the output (label vector) of each algorithm with the actual label vector by counting the number

TABLE I
SIMULATIONS BASED ON 3 MODELS AND 2 DIFFERENT BALANCING OF SAMPLES: 10:10 AND 12:8. IN THE COVARIANCE MATRICES COLUMN, I_d REPRESENTS THE $d \times d$ IDENTITY MATRIX. THE HYPERPARAMETERS $\mathbf{m}_i$, Ψ_i ν_i AND κ_i ARE DEFINED IN SECTIONS VI-B AND VI-C

Sim ID	Model	n_1	n_2	Repeats	Mean Vectors	Covariance Matrices	ν	κ
1	0	10	10	1000	$\mu_1 = (0,0)$, $\mu_2 = (1.5,1.5)$	$\Sigma_1 = \Sigma_2 = 1 \cdot I_d$	—	—
2	1	10	10	500	$\mathbf{m}_1 = (0,0)$, $\mathbf{m}_2 = (1.5,1.5)$	$\Sigma_1 = \Sigma_2 = 0.5 \cdot I_d$	$\nu_1 = 1$, $\nu_2 = 2$	—
3	2	10	10	500	$\mathbf{m}_1 = (0,0)$, $\mathbf{m}_2 = (1.5,1.5)$	$\Psi_1 = \Psi_2 = 0.5 \cdot I_d$	$\nu_1 = 1$, $\nu_2 = 2$	$\kappa_1 = 2$, $\kappa_2 = 3$
4	0	12	8	1000	$\mu_1 = (0,0)$, $\mu_2 = (1.5,1.5)$	$\Sigma_1 = \Sigma_2 = 1 \cdot I_d$	—	—
5	1	12	8	500	$\mathbf{m}_1 = (0,0)$, $\mathbf{m}_2 = (1.5,1.5)$	$\Sigma_1 = \Sigma_2 = 0.5 \cdot I_d$	$\nu_1 = 1$, $\nu_2 = 2$	—
6	2	12	8	500	$\mathbf{m}_1 = (0,0)$, $\mathbf{m}_2 = (1.5,1.5)$	$\Psi_1 = \Psi_2 = 0.5 \cdot I_d$	$\nu_1 = 1$, $\nu_2 = 2$	$\kappa_1 = 2$, $\kappa_2 = 3$

of points assigned to a cluster different from the actual one and dividing by n. Averaging these quotients across all point sets yields the empirical clustering error for each algorithm. When comparing vectors of labels, we account for the label switching issue (see Section II). Since we know the RLPP, for small point sets we can find the optimal partition, calculate its theoretical error, calculate the errors for the partitions output by the other algorithms, and make comparisons.

C. Results for Small Point Sets ($n = 20$)

Table I shows settings used for simulations on small point sets. In all cases, parameters are selected to obtain a Bayes clustering error close to 0.1. Additional results are available on the companion website for clustering error close to 0.2.

1) Clustering Operator Performance: To evaluate empirical errors, we use 1000 testing sets for Model 0 and 500 testing sets for Models 1 and 2. Fig. 3 displays average empirical errors as a function of the Hamming distance threshold h used in Algorithm 2 to define the set of reference partitions in Pmax and Pseed. The optimal and suboptimal algorithms are superior to all classical algorithms, particularly under Model 2, see Fig. 3(c) and (f). Performance is nearly optimal for both Pmax and Pseed even with very small h. For Pmax, this indicates that the partition with maximum probability is often the optimal partition. For Pseed, this leads to two additional conclusions: (1) Algorithm 1 with $k = 1$ often finds the maximum probability partition rather than getting trapped in local maxima, and (2) constraining the set of candidate partitions to the set of partitions with correct cluster sizes does not significantly degrade performance. The average difference between theoretical and empirical errors, provided on the companion website, has been found to be small in all cases.

2) Clustering Error Approximation Performance: Fig. 4 shows the root-mean-square (RMS) error over all samples in Simulation 1 between the approximate error for each algorithm (found using the probabilities of partitions within Hamming distance at most h from $\mathcal{P}_{seed}$) and the exact error (found using the probabilities of all partitions). Similar results for the other simulations are available on the companion website. In all cases the quality of error estimation is poor for small Hamming distances. This is in contrast with Fig. 3, where the quality of the clusters computed using $\mathcal{P}_{seed}$ remains quite stable across all values of h. This indicates that *clustering is easier than cluster error estimation*: suboptimal clusterers reach good clusters, but may not estimate their error accurately. The situation is analogous to classification, where a small sample may yield a decent classifier, but without the ability to obtain a satisfactory error estimate [23]. In addition, the best algorithms, the ones that produce better clusters, tend to suffer more from the estimation issue with larger RMS.

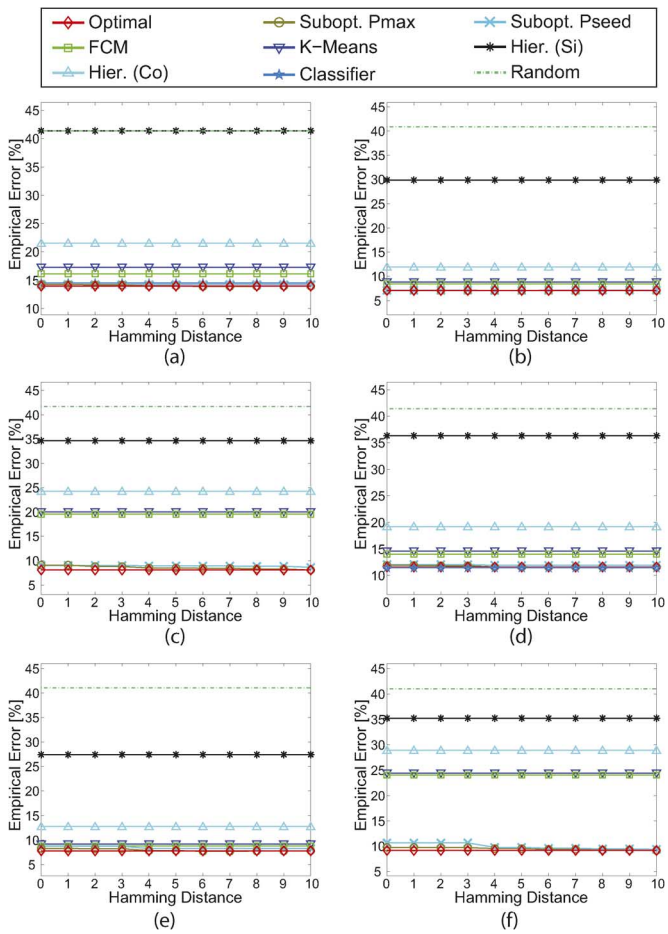


Fig. 3. Average empirical errors: (a) Simulation 1, (b) Simulation 2, (c) Simulation 3, (d) Simulation 4, (e) Simulation 5, (f) Simulation 6.

3) RAM Memory Usage and Processing Time: RAM memory usage and processing time to run all clustering algorithms in all simulations are provided in the supplementary materials as functions of the threshold h used in the suboptimal Pmax and Pseed algorithms. Pseed in particular requires very little time and memory for small h, where it is comparable to or even better than classical algorithms for $h \leq 5$. This is typical for all simulation settings we tested for small sample sizes. In some cases, the processing time and RAM memory usage for Pmax at maximum h can slightly exceed the optimal algorithm, although they produce the same results. This is because Pmax requires additional overhead when finding the maximum probability partition and its neighbors.

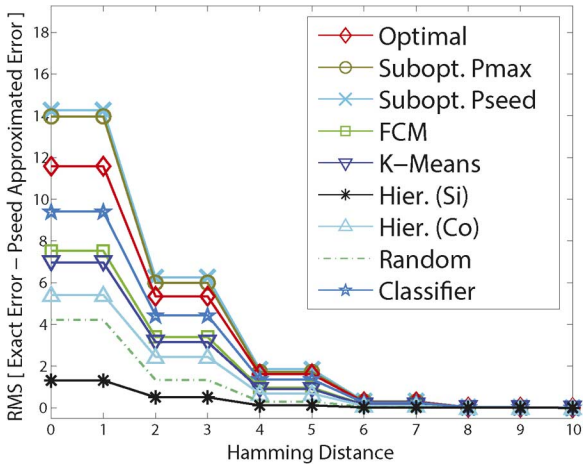


Fig. 4. RMS performance of error estimation for Simulation 1.

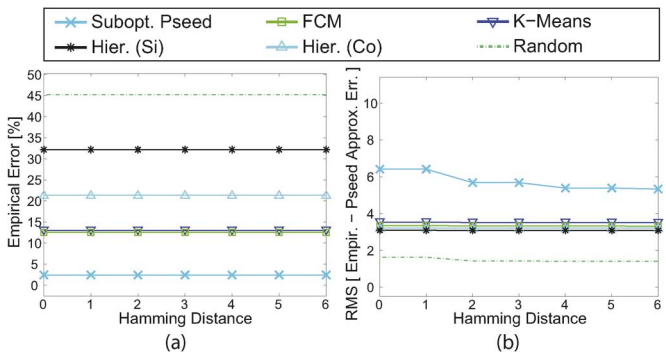


Fig. 5. Performance for moderate sized point sets drawn from Model 2: (a) average empirical errors, (b) RMS performance of error estimation.

D. Results for Moderate Sized Point Sets ($n = 70$)

The maximum size of the set of candidate and reference partitions grows exponentially with the number of points that compose a given point set to be clustered. Employing the optimal and Pmax algorithms for any but very small point sets is a computationally prohibitive task. Therefore, for this case we only test the performance of the suboptimal Pseed algorithm to cluster and estimate the clustering error. For this task, we draw 500 testing sets from Model 2 using the following settings: $n_1 = n_2 = 35$, $\mathbf{m}_1 = (0,0)$, $\mathbf{m}_2 = (6,6)$, $\Psi_1 = \Psi_2 = 3I_d$, $\nu_1 = \nu_2 = 2$, $\kappa_1 = 2$ and $\kappa_2 = 3$. The approximate Bayes clustering error in this model is close to 0.1. Results for 11 additional simulation settings are available on the companion website.

Fig. 5(a) shows average empirical errors for all algorithms tested. The averages are computed as a function of the Hamming distance threshold h used to define the set of candidate and reference partitions for the Pseed algorithm. Results are shown up to $h = 6$; using larger h considerably increases both the processing time and memory usage. Performance of the Pseed algorithm is superior to all the classical clustering algorithms tested and is almost constant as h increases.

Fig. 5(b) shows graphs of the RMS values between the approximate errors (found using the probabilities of partitions with a Hamming distance at most h from $\mathcal{P}_{seed}$) and empirical errors with respect to h. While with a small point set the exact error can be found using the probabilities of all partitions, for a large point set this is impossible because of the huge size of the set of reference partitions. Instead, we substitute the empirical error, or the Hamming distance between the true and predicted labels, for the exact error to approximate the RMS. As in the case of small point sets, error estimation using the Pseed suboptimal algorithm is more difficult than clustering itself.

Finally, graphs of memory usage and processing time for all clustering algorithms are available on the companion website.

E. Results for Large Point Sets (n Up to 10000)

We have shown that optimal clustering on small point sets can significantly outperform classical clustering algorithms, and that suboptimal clustering algorithms with Pseed are nearly optimal on small point sets and significantly outperform classical clustering algorithms on moderately large point sets. Thus far, the suboptimal methods we have proposed have been relatively computationally intensive owing to the desire to evaluate the Bayes clustering error. In this section we do not attempt to approximate clustering error, but instead focus on computationally efficient methods to perform clustering itself and how the new theory can be practically applied to very large point sets.

Given a partition $\mathcal{P}_{\text{seed}}$ having high probability found using Algorithm 1, we have seen that the Suboptimal Pseed algorithm performs quite well and is very fast when the maximum Hamming distance, h, is set to zero. Essentially, the key to good clustering, at least in our Gaussian models, is to focus on finding a maximum probability partition. We thus propose the following method, **Subopt. Pseed Fast**, to identify high probability partitions under very large point sets:

1) Select a random subset, S_{small}, of 100 points from the original large point set, S.
2) Initialize a partition on S_{small} with $\mathcal{P}_{\text{initial}}$. Here we use fuzzy c-means, since this method is quite fast and performs well in the Gaussian models we have tested.
3) Let $\mathcal{P}_{\text{equal}}$ be a partition having a designated number of points in each cluster (n_1 and n_2), chosen to keep the size of clusters proportional to known sizes of clusters in S, if available. This step is optional, but can improve clustering in models where the size of the clusters are known.
   a) Select one cluster in $\mathcal{P}_{\text{initial}}$; call it cluster A and the other cluster B.
   b) Evaluate the probability of all partitions having a Hamming distance 1 from $\mathcal{P}_{\text{initial}}$; note that each point in each cluster corresponds to a probability.
   c) Let $\mathcal{P}_1$ be the partition found by flipping points corresponding to the highest probability until cluster A has exactly n_1 points as follows: if cluster A has fewer than n_1 points, we flip points in cluster B corresponding to the highest probabilities, and if cluster A has more than n_1 points, we flip points in cluster A corresponding to highest probabilities.
   d) Similarly, let $\mathcal{P}_2$ be the partition found by flipping points corresponding to the highest probability until cluster A has exactly n_2 points.
   e) Evaluate the probabilities of $\mathcal{P}_1$ and $\mathcal{P}_2$.
   f) If the probability of $\mathcal{P}_1$ is greater than $\mathcal{P}_2$, then let $\mathcal{P}_{\text{equal}} = \mathcal{P}_1$; otherwise let $\mathcal{P}_{\text{equal}} = \mathcal{P}_2$.
4) Let $\mathcal{P}_{\text{small}}$ be the result of Algorithm 1 with input seed $\mathcal{P}_{\text{equal}}$ and Hamming distance $k = 2$.
5) Train a quadratic discriminant classifier by evaluating the sample mean and sample covariance of clusters formed

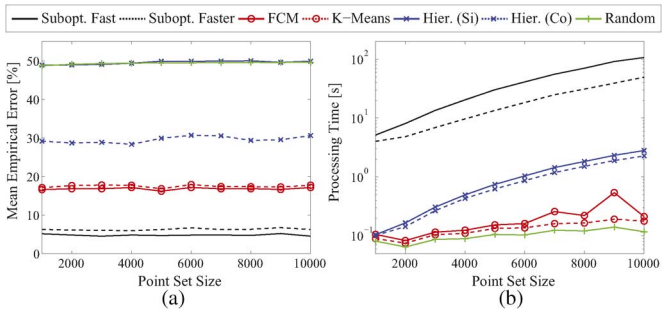


Fig. 6. Performance of clustering methods on large point sets drawn from Model 2: (a) average empirical errors, (b) average processing time.

by $\mathcal{P}_{\text{small}}$ on the small point set. Let $\mathcal{P}_{\text{large}}$ be the partition found by classifying all points in S with the above quadratic discriminant, where the classifier threshold is selected so that the cluster sizes match the known cluster sizes.

6) Let $\mathcal{P}_{\text{polish}}$ be the result of Algorithm 1 with input seed $\mathcal{P}_{\text{large}}$ and Hamming distance $k = 2$.
7) Repeat 10 times and select a final partition, $\mathcal{P}_{\text{seed}}$, having highest probability among the $\mathcal{P}_{\text{polish}}$.

The keys to this algorithm are in step (5), where we generalize clusters on a small point set to the full point set, and in steps (4) and (6), where we use Algorithm 1 to search for improved clusters using equations based on the optimal clustering theory. We also implement a modification of the above algorithm, which we call **Subopt. Pseed Faster**, where in step (6) we apply a modification of Algorithm 1 in which, naming the clusters A and B, one first selects the best single point to flip in cluster A based on partition probabilities, selects the best point to flip in cluster B, and iterates until there is no improvement.

To evaluate these algorithms, we draw 500 testing sets from Model 2 using the following settings: $n_1 = n_2$, where $n_1 + n_2$ varies from 1000 to 10000, $\mathbf{m}_1 = (0, 0)$, $\mathbf{m}_2 = (1.5, 1.5)$, $\Psi_1 = \Psi_2 = 0.5I_d$, $\nu_1 = 1$, $\nu_2 = 2$, $\kappa_1 = 2$, and $\kappa_2 = 3$. Except for sample sizes, these are the same settings from Simulation 3 in Fig. 3(c), where we observed a Bayes clustering error around 8% on point sets of size 20.

Fig. 6(a) shows the average empirical error with respect to point set size for all classical clustering algorithms, as well as Subopt. Pseed Fast and Subopt. Pseed Faster, described above. Both algorithms based on equations for the optimal clusterer significantly outperform all classical algorithms tested, and in particular they maintain error rates around 5–6% across all point set sizes, which is in fact slightly better than the optimal clusterer on 20 points. The optimal error may actually decrease with point set size since larger point sets carry more information, i.e., consider clustering a few points versus clustering when one essentially knows the Gaussian mixture distribution that generated the points. Thus, these suboptimal algorithms appear to suffer very little performance degradation relative to the optimal clusterer, even on very large point sets. The next best algorithms in terms of accuracy are fuzzy c-means and k-means, which achieve error rates around 17%, followed by hierarchical clustering with complete linkage at around 30%, and hierarchical clustering with single linkage, which performs as poorly as random clustering.

Graphs of average processing time are provided in Fig. 6(b), where the y axis is shown in log scale. A similar graph for average memory usage is available in the companion website. Subopt. Pseed Fast takes between 37 and 50 times longer than either hierarchical method, and Subopt. Pseed Faster takes between 17 and 40 times longer. Memory requirements for our methods are, in fact, less than hierarchical methods on very large point sets. Relative to fuzzy c-means and k-means, Subopt. Pseed Fast and Subopt. Pseed Faster do require substantially increased processing time and memory, but the requirements are not prohibitive. On a desktop PC clustering 10000 points with Subopt. Pseed Fast typically took 106 seconds and 100 MB of RAM, while Subopt. Pseed Faster typically took 50 seconds and 9 MB of RAM. In return, we achieve a 3 to 4 fold performance improvement over the best existing methods in this example. Furthermore, all processing time and memory results we have presented may be somewhat pessimistic, as all classical methods have been implemented using built-in MATLAB functions with optimized code.

X. Conclusion

Bayes decision theory for clustering parallels classification theory, but in the more difficult environment of random point sets. We have shown that Bayes clustering is equivalent to a Bayes risk model in which one considers all possible partitions of a point set along with a cost function. This leads to a fundamental difficulty in optimal clustering in which one must enumerate all possible partitions of a data set, something that is prohibitive even for moderately sized point sets. Thus, we have proposed approaches to reduce this computational complexity. Going a step beyond clustering itself, we have addressed methods to evaluate clustering error. Error estimation tends to be more difficult than optimal clustering itself, in the sense that error estimation requires examination of a much larger subset of the highest probability partitions than it takes to simply identify the best partition.

Given the results in this paper, two basic requirements for developing a rigorous theory of clustering are now in place: an appropriate probabilistic framework, including a measure of error [4], and a Bayes decision theory for clustering within that framework. Owing to the importance of clustering for science, engineering, and medicine, research should now proceed to develop a rigorous clustering theory that addresses many questions heretofore considered in classification theory and which allows clustering to be used in a scientific context rather than subjectively.

Two issues require immediate attention. First, in analogy with classification, the Bayes clusterer requires a known model. Since model uncertainty is commonplace in many situations, we need to develop the theory of robust clustering, that is, finding an optimal clusterer in the presence of model uncertainty. Although the problem has been solved under fairly general circumstances for nonrecursive filtering [24] and classification [25], the solution for clustering will be quite different because the signal in clustering is a random set and the objective is set partitioning rather than predicting a signal. Beyond that, owing to the difficulty in error estimation, as in the case of small-sample classification [26], we need to develop a theory of minimum-mean-square-error (MMSE) error estimation

that provides an error estimate that is optimal relative to the (uncertain) model and the data.

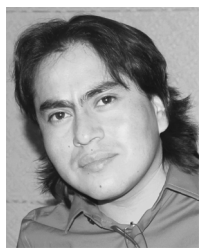
ACKNOWLEDGMENT

A companion website is available at: http://ece.osu.edu/~dalton/supplementary/2014BayesClustering.

REFERENCES

[1] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proc. Natl. Acad. Sci. USA*, vol. 95, pp. 14 863–14 868, 1998.
[2] M. K. Kerr and G. A. Churchill, "Bootstrapping cluster analysis: Assessing the reliability of conclusions from microarray experiments," *Proc. Natl. Acad. Sci. USA*, vol. 98, no. 16, pp. 8961–8966, 2001.
[3] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, 1999.
[4] E. R. Dougherty and M. Brun, "A probabilistic theory of clustering," *Pattern Recognit.*, vol. 37, no. 5, pp. 917–925, 2004.
[5] E. R. Dougherty, *Random Processes for Image and Signal Processing*. Bellingham, WA, USA: IEEE/SPIE Presses, 1999.
[6] D. Stoyan, W. S. Kendall, and J. Mecke, *Stochastic Geometry and Its Application*. Chichester, U.K.: Wiley, 1987.
[7] N. Cressie and G. M. Lasslett, "Random set theory and problems of modeling," *SIAM Rev.*, vol. 29, no. 4, pp. 557–554, 1987.
[8] L. Devroye, L. Gyorfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*. New York, NY, USA: Springer-Verlag, 1996.
[9] E. R. Dougherty and U. Braga-Neto, "Epistemology of computational biology: Mathematical models and experimental prediction as the basis of their validity," *Biologic. Syst.*, vol. 14, no. 1, pp. 65–90, 2006.
[10] U. Von Luxburg, M. Belkin, and O. Bousquet, "Consistency of spectral clustering," *Ann. Statist.*, pp. 555–586, 2008.
[11] K. Y. Yeung, C. Fraley, A. Murua, A. E. Raftery, and W. L. Ruzzo, "Model-based clustering and data transformations for gene expression data," *Bioinformatics*, vol. 17, no. 10, pp. 977–987, 2001.
[12] C. Fraley and A. E. Raftery, "Model-based clustering, discriminant analysis, density estimation," *J. Amer. Statist. Assoc.*, vol. 97, no. 458, pp. 611–631, 2002.
[13] T. S. Ferguson, "A Bayesian analysis of some nonparametric problems," *Ann. Statist.*, pp. 209–230, 1973.
[14] Y. Seldin and N. Tishby, "PAC-bayesian analysis of co-clustering and beyond," *J. Mach. Learn. Res.*, vol. 11, pp. 3595–3646, 2010.
[15] U. von Luxburg, R. C. Williamson, and I. Guyon, "Clustering: Science or art?," in *JMLR: Workshop and Conf. Proc.*, 2012, vol. 27, pp. 65–79, ser. Workshop on Unsupervised and Transfer Learning.
[16] E. Hartuv and R. Shamir, "A clustering algorithm based on graph connectivity," *Inf. Process. Lett.*, vol. 76, no. 4, pp. 175–181, 2000.
[17] R. Sharan and R. Shamir, "Click: A clustering algorithm with applications to gene expression analysis," in *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, 2000, vol. 8, no. 307, p. 16.
[18] A. Ben-Dor, R. Shamir, and Z. Yakhini, "Clustering gene expression patterns," *J. Computat. Biol.*, vol. 6, no. 3–4, pp. 281–297, 1999.
[19] J.-P. Brunet, P. Tamayo, T. R. Golub, and J. P. Mesirov, "Metagenes and molecular pattern discovery using matrix factorization," *Proc. Nat. Acad. Sci.*, vol. 101, no. 12, pp. 4164–4169, 2004.
[20] H. Chipman, T. J. Hastie, and R. Tibshirani, "Clustering microarray data," *Statist. Anal. Gene Express. Microarray Data*, vol. 1, pp. 159–200, 2003.
[21] M. Brun, C. Sima, J. Hua, J. Lowey, B. Carroll, E. Suh, and E. R. Dougherty, "Model-based evaluation of clustering validation measures," *Pattern Recognit.*, vol. 40, no. 3, pp. 807–824, 2007.
[22] L. A. Dalton, "On the optimality of K-means clustering," in *Proc. IEEE Int. Workshop Genomic Signal Process. Statist.*, Nov. 2013.
[23] E. R. Dougherty, A. Zollanvari, and U. M. Braga-Neto, "The illusion of distribution-free small-sample classification in genomics," *Current Genomics*, vol. 12, no. 5, pp. 333–341, 2011.
[24] L. A. Dalton and E. R. Dougherty, "Intrinsically optimal Bayesian robust filtering," *IEEE Trans. Signal Process.*, vol. 62, no. 3, pp. 657–670, 2014.
[25] L. A. Dalton and E. R. Dougherty, "Optimal classifiers with minimum expected error within a Bayesian framework-Part II: Properties and performance analysis," *Pattern Recognit.*, vol. 46, no. 5, pp. 1301–1314, 2013.
[26] 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 Trans. Signal Process.*, vol. 59, no. 1, pp. 115–129, 2011.

Lori A. Dalton (S'10–M'13) received the B.S., M.S., and Ph.D. degrees in electrical engineering at Texas A&M University, College Station, in 2001, 2002 and 2012, respectively. He is currently an Assistant Professor in the Department of Electrical and Computer Engineering and the Department of Biomedical Informatics at The Ohio State University in Columbus, OH. She was awarded an NSF graduate research fellowship in 2001, and she was awarded the Association of Former Students Distinguished Graduate Student Masters Research Award in 2003. Her current research interests include pattern recognition, classification, clustering, and error estimation.



Marco E. Benalcázar earned a Ph.D. in electronic engineering from the Universidad Nacional de Mar del Plata, Argentina, in 2014. In 2012, he received an M.S. in solar photovoltaic energy systems from the Universidad Internacional de Andalucía, Spain. From September 2012 to February 2013, he did research on clustering in the Genomic Signal Processing Laboratory at Texas A&M University in College Station, Texas, USA. In 2010, he was awarded a graduate fellowship from the Fundación Carolina, Spain. He also received fellowships from the Consejo Nacional de Investigaciones Científicas y Técnicas, Argentina, and the Secretaría Nacional de Educación Superior, Ciencia, Tecnología e Innovación, Ecuador. His current research interests include pattern recognition, machine learning, artificial vision, and digital image processing.

Marcel Brun received his Ph.D. in computer science from Universidad de San Pablo, Brazil, in 2002. Currently, he is Professor of the Department of Mathematics in the Engineering School at UNMDP, Argentina. He does research in the Digital Image Processing Group of UNMDP. His past research experience include postdoctoral stages at Texas A&M University (College Station, Texas), University of Louisville (Louisville, Kentucky) and the Translational Genomics Research Institute, TGen (Phoenix, Arizona). His areas of interest include pattern recognition applied to genomic signal processing and biomedical image processing. He is the author and coauthor of more than 30 journal articles.

Edward R. Dougherty (M'05–SM'09–F'12) is a Distinguished Professor in the Department of Electrical and Computer Engineering at Texas A&M University in College Station, TX, where he holds the Robert M. Kennedy 26 Chair in Electrical Engineering and is Scientific Director of the Center for Bioinformatics and Genomic Systems Engineering. He holds a Ph.D. in mathematics from Rutgers University and an M.S. in Computer Science from Stevens Institute of Technology, and has been awarded the Doctor Honoris Causa by the Tampere University of Technology in Finland. He is a fellow of SPIE, has received the SPIE Presidents Award.