Di Liu
Department of Statistics
Carnegie Mellon University

Cancer Pathology Classification
Comparing Sets in High Dimensions

Version: April 20, 2011
Committee: William Cohen, Stephen E. Fienberg, Ann Lee

1 Introduction

We concern ourselves with a particular setting in clustering and classification problems in which we have multiple data sources and we observe multiple data vectors from each data source. We are interested in comparing, classifying, or clustering the data sources rather than the individual data vectors. This setting has many real applications, including image analysis [7], genetics [2], and medical imaging [12].

In order to clearly illustrate our setting, we outline our main application (see Section 2). These data come from a study aimed at classifying the cancer status of human tissue samples using medical imaging; a similar study was reported in Wang et al [12]. From each tissue sample, we observe many images of individual cells. In this case, each tissue sample is a data source, and the individual cell images are the data points. The goal of the study is to determine the lesion type (class) of the tissue samples using the cell images.

In addition to the source structure, these data have other features and challenges which motivate our approach. First, each cell image must be viewed as a very complex, high dimensional object. Second, while each tissue sample displays some individual characteristics, many images from different tissue samples – even those from different cancer classes – display similar properties. Third, while there are many available images, each individual cell cannot be individually assigned a ground truth by expert pathologists. These three properties motivate a general semi-supervised learning approach to the task.

We give methods related to data from this particular setting as my thesis work. Instead of using a common SVM voting approach we model each data source S with a conditional distribution on the input space: $p_{X|S}$. In the above setting, many of the $p_{X|S}$ are similar, since many of the individual cells are similar across sources and classes. Therefore we can learn more information about these distributions by learning the marginal distribution p_X. Motivated by this property, we propose a semi-supervised approach to classification of data sources. Specifically, we use a quantization of the input space learned from all data from all sources to create an estimate of the distribution $p_{X|S}$ for each source. We propose two practical strategies for creating such a quantization: kmeans and mixture models (Section 3.4). We clearly explain the set of assumptions under which our strategy is better suited than the popular majority voting strategy (Section 3.3). We then use these estimates to create a pairwise distance matrix between all the data sources. This matrix may then be applied to a clustering or classification task. We show our method displays good classification performance in several applications.

2 Introduction to the Cancer Pathology Dataset

Surgical biopsy is currently the dominant method to diagnose many types of cancer. Medical imaging promises to be a cost-effective, minimally invasive, and least uncomfortable alternative diagnostic method. For both surgical and imaging tools, a pathologist must examine dozens of images to determine a diagnosis. The nuclear features of individual cells have been shown to be effective visual diagnostic features. However, due to the inherent limitations of the human brain and visual system [4], this complex data often produces uncertain diagnoses — even for relatively common diseases. Recent work uses computer visual algorithms applied over thousands of individual cell images in the diagnosis of several types of cancers including prostate [10], cervix [1], thyroid [3], liver [6], and breast [8].

Our data come from a study aimed at classifying the cancer status of human tissue samples using medical imaging; these data were originally reported in Wang et al [12]. We study the diagnosis of two sub-classes of thyroid cancer: follicular adenoma of the thyroid (FA — a milder form) and follicular carcinoma of the thyroid (FTC — a potentially deadly form). We begin with digital images of human tissue samples obtained from the archives of the University of Pittsburgh Medical Center (Institutional Review Board approval #PRO09020278). The pathologists took many microscope images from both cancerous (class FA and FTC) and healthy (class N) thyroid tissue. We next processed each image with an image segmentation algorithm in order to isolate subcellular structures, particularly nuclei because of their importance in diagnosis. After this step, each nuclei image can be viewed as a complex, high dimensional object — see Section 4 for additional details pertaining to this point. Remember that each tissue sample is represented by observations of many nuclei. Our goal, therefore, is not to classiify individual nuclei — we are interested in labeling groups of nuclei instead. Moreover, while each tissue sample displays some individual characteristics, many images from different tissue samples – even those from different cancer classes – display similar properties. Also, while there are many available images, each individual nuclei cannot be individually assigned a ground truth by expert pathologists.

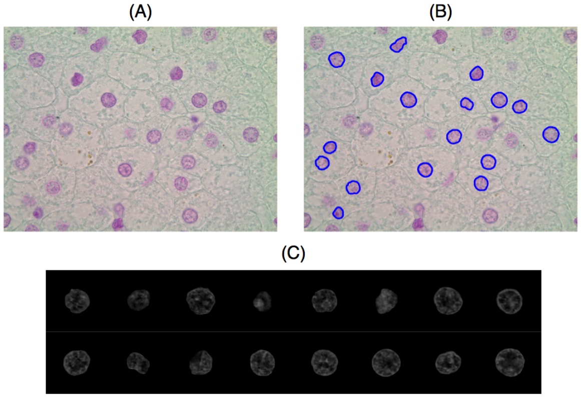

Figure 1: Summary of the image processing procedure (Figure taken from [12]). (A) Raw tissue sample image, (B) Segmented image, (C) Individual nuclei; converted to normalized grayscale.

	NA	**FA**	**FTC**
# Sources (Tissue Samples)	10	5	5
# Data vectors (Nuclei) from each source	35	70	70

Table 1: A summary of the cancer pathology data set. Note that each tissue sample (data source) has a different number of associated nuclei.

We now have a data set consisting of features from many images of individual nuclei, each coming from a particular tissue sample. Therefore, we let the tissue samples be the data sources and the nuclei image features be the individual data vectors. Table 2 summarizes the properties of the data set.

3 Classification and Clustering in the Set Setting

3.1 Classification Problems

In the traditional supervised learning setting, suppose we have data $\{(x_i, y_i)\}_{i=1}^n$. We seek a classifier g which maps any $x_i \in X$ to the classification label $y_i \in Y$. Our goal is to predict y_j when new x_j comes in. The ground truth $h : X- > Y$ is unknown, where X belongs to input space and Y is the classification label.

3.2 Notation

We now define the notation for our problem of interest: classification of data sources. Suppose we have data from $\mathcal{S}$ data sources, denoted $S_1, S_2, \ldots, S_{\mathcal{S}}$, with source S_i having $n^{(i)}$ data points in a common d dimensional space $\mathcal{X}$. Let $n = \sum_{i=1}^{\mathcal{S}} n^{(i)}$ be the total number of data vectors, and denote $\mathbf{x}_{ij}$ as the jth data vector from the data source S_i. We denote the resulting $n^{(i)} \times d$ data matrices as $\mathbf{X}_i$, and the $\mathbf{X}$ as the $n \times d$ matrix containing all data from all sources. Define $\mathcal{L}(\mathbf{x})$ as the function which returns the data source of $\mathbf{x}$; $\mathcal{L}(\mathbf{x})$ takes values in $\{S_1, S_2, \ldots, S_{\mathcal{S}}\}$. In the classification setting, we denote $\mathcal{Y}$ as the label space, where each source — rather than each data point — has a label. Thus, we have in total n examples $\mathbf{z}_{ij} = (y_i, \mathbf{x}_{ij})$. We denote p_X as the marginal distribution of the data vectors over $\mathcal{X}$, p_{XY} as the joint distribution of the $\mathbf{z}$ over $\mathcal{X} \times \mathcal{Y}$, and $p_{X|S_i}$ as the conditional distribution of the $\mathbf{x}$ over $\mathcal{X}$, given $\mathbf{x}$ comes from source S_i.

3.3 Assumptions on the Conditional Distributions

When $\mathcal{X}$ is high dimensional, classification tasks become more difficult, particularly if n is small relative to d. In this case, we propose a semi-supervised approach to the classification problem. In particular, we consider an assumption related to the manifold assumption, which has good theoretical support in semi-supervised learning. In our setting, we instead make a related assumption on the conditional distributions $\{p_{X|S_i}\}$:

Condition 3.1. *(Source Distribution Assumption) Suppose the marginal distribution on the input space places its mass on only a small subset of $\mathcal{X}$. That is, suppose the marginal distribution p_X has an associated probability measure μ_X such that for some small $\delta_X, \epsilon_X > 0$, there exists $\mathcal{M} \subset \mathcal{X}$ such that $\mu_X(\mathcal{M}) \geq 1 - \delta_X$ and $\mu(\mathcal{M}) < \epsilon_X$, where μ denotes Lebesgue measure on $\mathcal{X}$. We further consider three possible conditions on $\{p_{X|S}\}$:*

1. *The distributions $p_{X|S}$ fall into only a few classes. That is, suppose we have a collection of index sets $C = \{\mathcal{C}_c\}$, with $|C| < \mathcal{S}$ such that $\forall i, j \in \mathcal{C}_c, p_{X|S_i} = p_{X|S_j}$.*
2. *As above, the distributions $p_{X|S}$ fall into only $|C|$ classes. However, within each class, the distributions are similar rather than identical. We may formalize this for example, as: given a constant $E > 0$, $\forall c, \forall i, j \in \mathcal{C}_c \; \exists \, \epsilon_h \in (0, E)$ s.t. $d(p_{X|S_i}, p_{X|S_j}) < \epsilon_h$, where $d(\cdot,\cdot)$ denotes Hellinger distance.*
3. *The distributions $p_{X|S}$ are unrelated. As above, we may formalize this, for example, as: given a constant $E > 0 \; \forall c, \forall i, j \in \mathcal{C}_c \; \exists \, \epsilon_h > E$ s.t. $d(p_{X|S_i}, p_{X|S_j}) > \epsilon_h$, where $d(\cdot,\cdot)$ denotes Hellinger distance.*

Condition 3.1(1) corresponds to the following data generation scenario. For each class, we first draw a collection of data $\mathbf{X}$ according to $p_{X|Y}$. These data are then randomly assigned source labels. Therefore, given the data, the source and class labels are independent. We then might guess that the data sources give us little additional information. The following proposition formalizes this intuition:

Proposition 3.1. *Suppose Condition 3.1(1) holds. Then, when $C = 2$ in the classification setting, the optimal classifier for data source classification is obtained empirically using a voting strategy.*

Proof. Suppose we have $n^{(S)}$ data points from source S. The Bayes' rule for sources is defined as:

$$h_S^*(S) = \begin{cases} 1: & r_S(S) > \frac{1}{2} \\ 0: & \text{else} \end{cases}$$

Where the source regression function $r_S(S)$ can be written as follows:

$$\begin{aligned} r_S(S) &:= r_S(\mathbf{x}) \\ &= \mathbb{E}(Y|\mathcal{L}(\mathbf{x}) = S) \\ &= \mathbb{P}(Y = 1|\mathcal{L}(\mathbf{x}) = S) \\ &= \int_{\mathcal{X}} \mathbb{P}(Y = 1|\mathcal{L}(\mathbf{x}) = S, X = \mathbf{x}) p_{X|S}(\mathbf{x}) d\mathbf{x} \\ &= \mathbb{E}_{X|S} \mathbb{P}(Y = 1|\mathcal{L}(\mathbf{x}) = S, X = \mathbf{x}) \\ &= \mathbb{E}_{X|S} \mathbb{P}(Y = 1|X = \mathbf{x}) \end{aligned}$$

Where, in the last step we have used our assumption implicit in Condition 3.1(1). Inside this last expectation, we have the traditional regression function $r(\mathbf{x}) = \mathbb{P}(Y = 1|X = \mathbf{x})$, and so we can estimate the expectation empirically using a voting estimator:

$$\mathbb{E}_{X|S}\mathbb{P}(Y = 1|X) \approx \frac{1}{n^{(S)}} \sum_{j=1}^{n^{(S)}} \hat{r}(\mathbf{x}_{ij})$$

□

In particular, Proposition 3.1 suggests that under Condition 3.1(1), we may ignore the source information entirely. We need only use a plug in rule estimator combined with a voting strategy to well approximate the Bayes' rule. Thus, Condition 3.1(1) is very reductive.

Though Conditions 3.1(2) and (3) above are loosely defined, they motivate our semi-supervised approach. Under condition (3), learning p_X conveys little information about the individual source distributions p_{X_S} than using only $\mathbf{X}_i$. Thus, a semi-supervised approach may not help the classification or clustering task. Under condition (2), learning the p_X may give more information about the individual source distributions $p_{X|S}$. Here, we can use the data $\mathbf{X}_i$ as labeled data combined with $\{\mathbf{x} : \mathcal{L}(\mathbf{x}) = S_k \text{ s.t. } k \neq i\}$ as unlabeled data to learn about $p_{X|S_i}$. Since, under Condition 3.1(2), there are several members of the set $\{p_{X|S_j} : j \neq i\}$ which are related to $p_{X|S_i}$, this unlabeled data may give more information about $p_{X|S_i}$. We next give a method motivated by the second condition.

3.4 Quantizing the Input Space

We now present a method for finding the relationships between data sources under Condition 3.1(2). We use a semi-supervised approach to estimate the conditional distributions $\{p_{X|S_i}\}|_{i=1}^{\mathcal{S}}$ using a quantization of the input space. We then compute the distances between these estimates, giving us an estimated pairwise distance matrix between the data sources for use in future inference.

We begin by partitioning the space $\mathcal{X}$ into K bins. Let each bin be represented by a basis function $\phi_k(\cdot) : \mathcal{X} \to \mathcal{H}$, where $\mathcal{H}$ is left arbitrary. For each bin, we define a functional $V_k(\cdot) : \mathcal{X} \to \mathbb{R}$. Therefore, we can approximate a source S_i in $\mathcal{H}$ with our bins using the general equation:

$$S_i \approx \tilde{S}_i = \frac{\sum_k \left(\int_{\mathcal{X}} V_k(\mathbf{x})\ d\mu_{S_i}(\mathbf{x})\right) \phi_k}{\sum_k \int_{\mathcal{X}} V_k(\mathbf{x})\ d\mu_{S_i}(\mathbf{x})}.$$

We then define entry k of the quantized projection of S_i, h_{S_i} as follows:

$$h_{S_i}[k] = \frac{\int_{\mathcal{X}} V_k(\mathbf{x}) d\mu_{S_i}(\mathbf{x})}{\sum_k \int_{\mathcal{X}} V_k(\mathbf{x})\ d\mu_{S_i}(\mathbf{x})}. \tag{1}$$

In data applications, we may estimate the projection empirically as follows:

$$\hat{h}_{S_i}[k] = \frac{\sum_{j=1}^{n^{(k)}} \hat{V}_k(\mathbf{x})}{\sum_l \sum_{j=1}^{n^{(l)}} \hat{V}_l(\mathbf{x})}. \tag{2}$$

Here, $\hat{V}_k$ is an empirical estimate of the bin functional V_k. We now discuss methods for estimating the partition.

3.5 Creating the Quantization

We base our semi-supervised method on a quantization of the space $\mathcal{X}$. Here, we consider two popular vector quantization (VQ) methods and define the corresponding projection functional estimators $\hat{V}_k$. VQ methods seek to represent data in terms of a common set of standard patterns called a codebook. Our idea and approach are similar.

3.5.1 Kmeans as a Space Partition Method

In our setting, we can view kmeans as a data driven partitioning method which minimizes the expected value of the distortion caused by mapping data points to their cluster centers.

We first fit kmeans to all of the data $\mathbf{X}$ and use each of the K clusters as a bin. We use the resulting K centers as estimates for our quantization basis $\{\hat{\phi}_k\}$. We define the following function as our estimate of the bin functionals:

$$\hat{V}_k(\mathbf{x}) = \begin{cases} 1: & \underset{j \in \{1,2,\ldots,K\}}{\operatorname{argmin}} (||\mathbf{x} - \hat{\phi}_j||_X) = k \\ 0: & \text{else} \end{cases}. \tag{3}$$

Here, $||\cdot||_X$ represents the norm in the space $\mathcal{X}$. Such a definition is very simple and intuitive: the empirical projection of S_i on the k^{th} component of the partition is just the proportion of points falling into the k^{th} kmeans cluster.

3.5.2 Mixture Models

Mixture models are another natural quantization technique. Mixture models represent a probability distribution as a mixture of K component distributions. To create a quantization with K bins we fit the following model:

$$p(\mathbf{x}) = \sum_{g=1}^{K} p(Z = g) p(\mathbf{x}|Z = g),$$
$$(\mathbf{x}|Z = g) \sim \mathcal{G}_g(\boldsymbol{\theta}_g).$$

Here, $\mathcal{G}_g$ is a distribution, possibly parameterized by the vector of parameters $\boldsymbol{\theta}_g$. Z is a latent class variable which stands for the mixture components. We define the collection of parameters $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K\}$. Note that the $\{\mathcal{G}_g\}$ need not be from the same family of distributions, though such an assumption often aids model fitting.

After fitting a mixture model, we obtain an estimate $\hat{\boldsymbol{\theta}}$ for $\boldsymbol{\theta}$. We let $\hat{\phi}_k = \mathcal{G}_k(\hat{\boldsymbol{\theta}}_k)$. We define our bin functions as:

$$\hat{V}_k(\mathbf{x}) = p(Z = k|\mathbf{x}, \hat{\boldsymbol{\theta}}_k). \tag{4}$$

The above definition leads to the following statistical properties.

Proposition 3.2. *Suppose $\hat{\boldsymbol{\theta}}_k$ is consistent for $\boldsymbol{\theta}_k$. Then, using the estimate for $\hat{V}_k$ defined in Equation 4, the estimator $\hat{h}_{S_i}[k]$ is a consistent estimator of $\mathbb{P}(Z = k|\mathcal{L}(\boldsymbol{x}) = S_i, \boldsymbol{\theta})$.*

Proof. For brevity of notation, we replace $\mathcal{L}(\mathbf{x}) = S_i$ with S_i. We now derive an estimate for this quantity. By the law of total probability:

$$\begin{aligned}\mathbb{P}(Z = k|S_i, \boldsymbol{\theta}) &= \int_{\mathcal{X}} \mathbb{P}(Z = k|S_i, \mathbf{x}, \boldsymbol{\theta}) p(\mathbf{x}|S_i, \boldsymbol{\theta}) d\mathbf{x} \\ &= \mathbb{E}_{\mathbf{x}|S_i, \boldsymbol{\theta}} \left(\mathbb{P}(Z = k|S_i, \mathbf{x}, \boldsymbol{\theta})\right). \end{aligned} \tag{5}$$

By the law of large numbers, and the consistency of $\hat{\boldsymbol{\theta}}_k$, a consistent estimator for the expectation in Eq. 5 is:

$$\hat{h}_{S_i}[k] = \frac{1}{n^{(i)}} \sum_{j=1}^{n^{(i)}} \mathbb{P}(Z = k|S_i, \mathbf{x}_{ij}, \hat{\boldsymbol{\theta}}_k). \tag{6}$$

□

Proposition 3.3. *Using the estimate for $\hat{V}_k$ defined in Equation 6, $\hat{h}_{S_i}[k]$ is also MLE for $\mathbb{P}(Z = k|\mathcal{L}(\boldsymbol{x}) = S_i, \boldsymbol{\theta})$*

Proof. We begin with the log-likelihood for $\mathbf{X}$ and the labels. For simplicity of notation, we drop conditioning on $\boldsymbol{\theta}$, which is fixed in all of these calculations. We also let the event z stand for $Z = z$.

$$\begin{aligned}\ell(\mathbf{X}, \mathcal{L}(\mathbf{X})) &= \sum_{l=1}^{n} \log \mathbb{P}(\mathbf{x}_l, \mathcal{L}(\mathbf{x}_l)) \\ &= \sum_{l=1}^{n} \log \sum_{z=1}^{K} \mathbb{P}(\mathbf{x}_l|\mathcal{L}(\mathbf{x}_l), z)\mathbb{P}(z|\mathcal{L}(\mathbf{x}_l))\mathbb{P}(\mathcal{L}(\mathbf{x}_l))\end{aligned}$$

Now, we wish to maximize this quantity with respect to the variable $\mathbb{P}(z|\mathcal{L}(\mathbf{x}_l))$. To do this, we use a Lagrange multiplier from the constraint $\sum_{z=1}^{K} \mathbb{P}(z|\mathcal{L}(\mathbf{x}_l)) = 1$:

$$\ell(\mathbf{X}, \mathcal{L}(\mathbf{X})) + \lambda \left(\sum_{z=1}^{K} \mathbb{P}(z|\mathcal{L}(\mathbf{x}_l)) - 1\right)$$

Without loss of generality, we take the derivative with respect to $\mathbb{P}(Z = z|\mathcal{L}(\mathbf{x}_l) = S_i)$ and set equal to zero. Using this, we can rewrite the above as follows:

$$\mathbb{P}(z|S_i) = \frac{1}{n^{(i)}} \sum_{\{\mathbf{x}_l : \mathcal{L}(\mathbf{x}_l) = S_i\}} \frac{\mathbb{P}(\mathbf{x}_l|S_i, z)\mathbb{P}(z|S_i)}{\sum_{z=1}^{K} \mathbb{P}(\mathbf{x}_l|S_i, z)\mathbb{P}(z|S_i)}.$$

Here, We write the event $\mathcal{L}(\mathbf{x}_l) = S_i$ as S_i, and $Z = z$ as z. Examining the term in the above summation, we have:

$$\frac{\mathbb{P}(\mathbf{x}|S_i, z)\mathbb{P}(z|S_i)}{\sum_{z=1}^{K} \mathbb{P}(\mathbf{x}|S_i, z)\mathbb{P}(z|S_i)} = \mathbb{P}(z|S_i, \mathbf{x}).$$

So we have that the MLE for $\mathbb{P}(z|S_i)$ is the empirical average in Eq. 4. □

3.5.3 Special Case: Gaussian Mixture Models

A popular type of mixture model is the Gaussian Mixture Models (GMMs), a related technique to kmeans. GMMs represent a density $p(\mathbf{x})$ as a mixture of K Gaussian distributions, under the following model:

$$p(\mathbf{x}) = \sum_{g=1}^{K} p(Z = g) p(\mathbf{x}|Z = g),$$
$$(\mathbf{x}|Z = g) \sim N(\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g).$$

We define $\boldsymbol{\theta} = \{\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2, \dots, \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_K\}$ as the set of all the Gaussian component parameters. After fitting a GMM, we obtain an estimate for $\boldsymbol{\theta}$, $\hat{\boldsymbol{\theta}}$. We thus define:

$$\hat{V}_k(\mathbf{x}) = p(Z = k|\mathbf{x}, \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k) \tag{7}$$

3.6 Distances Between Distributions

So far, we have outlined practical ways to obtain estimates $\hat{h}_{S_i}$ for the quantized estimates of the conditional data sources distributions. Next, we compute the pairwise distance or dissimilarity between these histograms to obtain a $\mathcal{S} \times \mathcal{S}$ matrix $\mathcal{D}$ where entry (i, j) contains the estimated distance or dissimilarity between source S_i and source S_j; $d(\hat{h}_{S_i}, \hat{h}_{S_j})$. $\mathcal{D}$ can be used to cluster or classify the data sources in a variety of ways; such as nearest neighbors, support vector machines, minimal spanning trees, or hierarchical clustering. We now give some effective ways from literature to compute the distance or dissimilarity between these estimates. [9] gives a thorough review of various distance measures. Two commonly used distance measures between a pair of histograms $\{h_1, h_2\}$ are:

- L^2 distance:

$$d(h_1, h_2) = \sqrt{\sum_k (h_1[k] - h_2[k])^2}. \tag{8}$$

- Quadratic form distance [7]:

  $$d(h_1, h_2) = \sqrt{(h_1 - h_2)^T A (h_1 - h_2)}. \tag{9}$$

  Here, A is typically a pairwise similarity matrix between all of the bins. We base the similarity between bins on similarities between the $\hat{\phi}_k$.

L^2 distance is the simplest approach, but it ignores inter-bin relationships in the space $\mathcal{X}$. This is a particular problem when points are put into bins via hard assignment. Quadratic form attempts to take relative position of bins into account. Earth mover's distance [9] is another popular measure, used commonly in image analysis. However, in our experiments, we found it did not lead to better classification results.

3.6.1 Weighting the Quantization

We now consider an extension of our method particularly adapted for the classification setting. In many applications the data from the various classes can be highly overlapping, complicating the classification task. In this case, drawing a decision boundary between classes becomes more difficult. Therefore, data points in these areas of overlap may make make classifying the related sources more difficult. Thus, in the classification setting, some areas of the marginal distribution may be more valuable for classification than others. Recall our method is based on a quantization of the marginal distribution p_X. For example, in the kmeans case, a bin containing only vectors from a single class might be considered more important than a bin which contains an equal mix of vectors from all classes.

We propose a weighting method which automatically considers the importance of bins and adapts to the distance measure we use. Specifically, we wish to find a weighting scheme which emphasizes the difference between sets from different classes as well as preserves the commonplace between sets from the same class. Note that this method now takes into account the labels of all the points in the training set, bringing the method closer to supervised learning.

Let $\mathcal{I}$ represent the collection of classes, so $\mathcal{I} = \{1, 2, ...C\}$, let $\mathcal{I}_c = \{i : S_i \in c, c \in \mathcal{I}\}$. When we have K bins, we represent the weights as a vector $\mathbf{w} = \{w_1, w_2, ..w_K\}$. For each data source S_i, we have a representation h_{s_i} of $p_{X|S_i}$ in terms of the quantization of the input space. We weight by taking the dot product $\mathbf{w} \cdot h_{S_i}$. Thus, we denote the distance between two sources S_i, S_j without weighting as $d(h_{s_i}, h_{s_j})$, and the distance with weighting as $d(\mathbf{w} \cdot h_{S_i}, \mathbf{w} \cdot h_{S_j})$. Towards our goal, we seek to minimize the difference between the average distance between data sources from the same class and the average distance between data sources from different classes. We propose the following scheme for the weights:

$$
\begin{aligned}
\hat{\mathbf{w}} &= \underset{\mathbf{w}:\mathbf{w}\in[0,K]^K,||\mathbf{w}||_1=K}{\operatorname{argmin}} \mathcal{A}_1 - \mathcal{A}_2, \\
\mathcal{A}_1 &= \frac{1}{N_1} \sum_{(i,j)\in\{\mathcal{I}_c\times\mathcal{I}_c\}} d(\mathbf{w}\cdot h_{S_i}, \mathbf{w}\cdot h_{S_j}), \\
\mathcal{A}_2 &= \frac{1}{N_2} \sum_{(i,j)\in\{\mathcal{I}_c\times\mathcal{I}'_c:c\neq c'\}} d(\mathbf{w}\cdot h_{S_i}, \mathbf{w}\cdot h_{S_j}).
\end{aligned}
$$

Where $N_1 = \sum_{c=1}^{C} \frac{|I_c|(|I_c|+1)}{2}$ is the count of pairs with the same class label and $N_2 = \frac{\mathcal{S}(\mathcal{S}-1)}{2} - N_1$ is the count of pairs with different class labels. Such a minimization scheme suffers from two fatal problems.

1. For many choices of $d(\cdot,\cdot)$, the solution tends to put all the weight on the bin in which data sources are maximally differentially present (for example, where only a single source is present) and zero weight on the others. Usually, we wish to consider a wide range of bins in our distance measure.

2. The average distance between sets may be heavily affected by outliers. For example, if the distance between a particular pair of sources is much larger than other pairs, then this weighting scheme tends to favor bins containing these sources. However, we are often very concerned with sources that are highly overlapping and/or separated by relatively small distances.

We address the first problem by adding a regularization term $||\mathbf{w}||_2^2$. To address the second issue, we normalize the distances between sources by dividing by the unweighted distance. Thus, the new optimization problem becomes:

$$
\begin{aligned}
\hat{\mathbf{w}} &= \underset{\mathbf{w}:\mathbf{w}\in[0,K]^K,||\mathbf{w}||_1=K}{\operatorname{argmin}} \mathcal{A}_3 - \mathcal{A}_4 + \lambda||\mathbf{w}||_2^2, \\
\mathcal{A}_3 &= \frac{1}{N_1} \sum_{(i,j)\in\{\mathcal{I}_c\times\mathcal{I}_c\}} \frac{d(\mathbf{w}\cdot h_{S_i}, \mathbf{w}\cdot h_{S_j})}{d(h_{s_i}, h_{s_j})}, \\
\mathcal{A}_4 &= \frac{1}{N_2} \sum_{(i,j)\in\{\mathcal{I}_c\times\mathcal{I}'_c:c\neq c'\}} \frac{d(\mathbf{w}\cdot h_{S_i}, \mathbf{w}\cdot h_{S_j})}{d(h_{s_i}, h_{s_j})}.
\end{aligned}
$$

Here, we let $\lambda > 0$ be a tuning parameter, which controls the amount of regularization (see Section 3.7).

If we use L_2 or quadratic distance for the function $d(\cdot,\cdot)$, we can write the minimization problem as a quadratic programing problem. Under this case $d(\mathbf{w}\cdot h_{s_i}, \mathbf{w}\cdot h_{s_j}) = (|h_{s_i} - h_{s_j}|\cdot\mathbf{w})^T A(|h_{s_i} - h_{s_j}|\cdot\mathbf{w})$. Here, A is a $K\times K$ matrix. For L_2 distance, $A = I$. For quadratic form distances, A may be any matrix; a common choice is a similarity matrix between the K bins. Letting

$x_{ij} = |h_{s_i} - h_{s_j}|$, we can rewrite the optimization problem as:

$$\begin{aligned} \hat{\mathbf{w}} &= \underset{\mathbf{w}:\mathbf{w}\in[0,K]^K, ||\mathbf{w}||_1=K}{\text{argmin}} \quad \mathbf{w}^T \left(\mathcal{A}_5 - \mathcal{A}_6 + \lambda I\right) \mathbf{w}, \qquad (10) \\ \mathcal{A}_5 &= \frac{1}{N_1} \sum_{(i,j)\in\{\mathcal{I}_c \times \mathcal{I}_c\}} \frac{x_{ij}x_{ij}^T \circ A}{x_{ij}^T A x_{ij}}, \\ \mathcal{A}_6 &= \frac{1}{N_2} \sum_{(i,j)\in\{\mathcal{I}_c \times \mathcal{I}'_c : c\neq c'\}} \frac{x_{ij}x_{ij}^T \circ A}{x_{ij}^T A x_{ij}}. \end{aligned}$$

Here, $A \circ B$ denotes the Hadamard product of the matrices A and B. The above problem can be solved by a variety of optimization software packages.

3.7 Choice of Tuning Parameters

Our method now involves up to two tuning parameters: the number of bins K and the regularization parameter λ. The quantization schemes we proposed also have applications in clustering, so it might seem natural to choose K using one of numerous clustering criteria. However, this often leads to a poor quantization with too few bins. The goal of our method is to give a rich representation of the marginal p_X rather than to cluster the data directly into groups. Towards this goal, we instead recommend that K is chosen via cross-validation in the classification setting. Generally, kmeans needs fewer bins than a GMM. This is because a GMM considers the relative position of bins and therefore is more stable. Kmeans is also sensitive to starting points. For our applications, we borrow the idea of "combining classifiers" by running kmeans many times and predicting based on a majority vote . This process makes kmeans less sensitive to both starting points and the choice of K.

For λ, consider Equation 10. If the matrix $H = \mathcal{A}_5 - \mathcal{A}_6 + \lambda I$ is positive-definite the objective function is convex and will have a unique global minimum. In particular, we can show that there exists λ_0, such that when $\lambda > \lambda_0$, the function is convex. Denote $H' = H - \lambda I$, with eigenvalues $\alpha_1, \alpha_2, ...\alpha_k$. Let $\alpha = \min(\alpha_1, \alpha_2, ...\alpha_n)$. If $\alpha > 0$ then $\lambda_0 = 0$. else let $\lambda_0 = |\alpha|$, then $H = H' + \lambda I$ is positive definite for any $\lambda > \lambda_0$.

Note that the term λ roughly controls how evenly the entries of $\mathbf{w}$ are distributed. When $\lambda = 0$, the $\hat{\mathbf{w}}$ tends to have be a zero vector except for one entry with weight K, as discussed above. As λ grows larger, the weights become more evenly distributed. As λ goes towards infinity, each bin has weight 1, which corresponds to the unweighted solution. We are principally concerned that $\lambda > \lambda_0$ and is not too large, rather than a particular choice of λ. Consequently, λ need not be tuned very carefully.

4 Applying our method to the dataset

Instead of following a usual pipeline method where we represent each image as a number of features, we consider computing Optimal Transportation (OT) distance between each pair of nuclei, as discribed in [12]. In this application, each nuclear structure is characterized by a set of n^2 pixel measurements. The simplest approach is to compute the Euclidean distance between vectors representing different images, but this was shown to not be able to capture nuclear mophology [12]. For this reason, Wang et al instead considered an optimal transportation (OT), i.e. Kantorovish-Wasserstein, based metric. The idea is to measure the amount of effort to "transport" the chromatin content, as discribed in the images' pixel intensity values, of one nuclear configuration to another. The distance measure takes into account the "overall" difference between nuclear mophology, but it is most sensitive to changes in the chromatin distribution. The formula of such a transportation is given by:

$$\min_{f_{i,j}} \sum_{i=1}^{N_p} \sum_{j=1}^{N_q} d(X_i, Y_j) f_{i,j} \tag{11}$$

Here, N_p and N_q are the number of pixels in each image, $d(X_i, Y_j)$ is a measure of work to move a unit of mass from location X_i in the first image to Y_j in the second, and $f_{i,j}$ represents the amount of mass moved. In the optimal transportation setting we have $d(X_i, Y_j) = ||X_i - Y_j||^2$. This can be easily formulated into a linear programming proGram and solved by a variety of software packages. Please refer to [9] for a detailed introduction of related distance measures.

The optimal transportation step results in a pair-wise $N \times N$ distance where N is the number of nuclei. We apply our method to this distance measure data set. We consider one nearest neighbor as our classifier. By considering a nearest neighbor classifier, we match each tissue sample to the "most similar tissue sample", and the label of the tissue samples should be similar. We next compare our results with an effective baseline method in which each individual nuclei is classified with SVM and the label of a sample tissue is obtained by a majority vote [12] [5] [11]. For the SVM baseline we report the classification results, where the parameters are chosen using 5-fold cross validation, and the classification risk is estimated with a test set of one tissue sample. The reported results are averaged over 30 runs, where each run consists of using each tissue sample as the test set, and the parameters are chosen via cross validation on the remaining tissue samples.

The SVM baseline ignores the tissue sample structure in the data. Rather, it considers the data point and class as independent — disregarding whether two nuclei came from the same tissue sample. Therefore, we also compare our results to another optimal transport (OT) based method. In this approach, we compute the OT between each pair of tissue samples, rather than each nuclei. We compute the OT distance between tissue samples i and j by using the matrix of OT distances between all pairs of nuclei in i and j. These distances are used as $d(X_i, Y_j)$ as in Equation 11. Since we wish to consider each nuclei as equally important, we assign a weight of $1/n_i$ to each nuclei from tissue sample i. Here, n_i denotes the number of nuclei from tissue sample i. This approach yields

Method	**Classification Rate**
SVM - RBF Kernel	96% (average over 30 runs)
Our Group Method	98% (average over 30 runs)
OT between tissue samples (Non-geometric)	95%

Table 2: A summary of classification results on the cancer imaging data on the optimal transportation distance we constructed. The SVM baseline and the OT between tissue samples approaches were, on average, able to classify all but one case (19/20) correctly. Our diffusion kmeans approach is able to more often attain perfect results.

a pairwise distance matrix between all tissue samples, which can be used for classification by a simple nearest neighbor method.

We report the results of our experiments in Table 2. Due to the potential instability of kmeans, we report the average of 30 runs of our diffusion kmeans method. The SVM baseline and the OT between tissue samples approaches were, on average, able to classify all but one case (19/20) correctly. Our diffusion kmeans approach is more often able to attain perfect results.

In the SVM voting baseline method, we implicitly assume that tissue samples with the same class label (NL, FA, FTC) have the same distribution. This might not be the case for our data, as shown in Figure 4. The Figure displays the overall distribution of each class, as well as a comparative plot for two selected samples from each class. These are plotted in the coordinates returned by applying the diffusion map. We see that the three classes have somewhat overlapping distributions, making the classification challenging. In particular, we see that support of each tissue sample is not necessarily the same as the support of the corresponding class distribution. Our approach, as opposed to the SVM baseline, takes these properties into account, leading to our superior results.

On the other hand, the OT between tissue samples method does consider the special structure of the data. As with our approach, we calculate a pairwise distance between tissue samples, rather than basing everything on distances between nuclei. However, for a given pair of tissue samples, the OT approach only considers the OT distances between nuclei within those two samples. This ignores the overall structure of the data. In contrast, the diffusion map constructs a low dimensional representation of the overall distributional structure of the data. Our geometry-based approach therefore effectively finds the structure of the dataset, giving us an additional advantage over an approach that only considers pairwise structure.

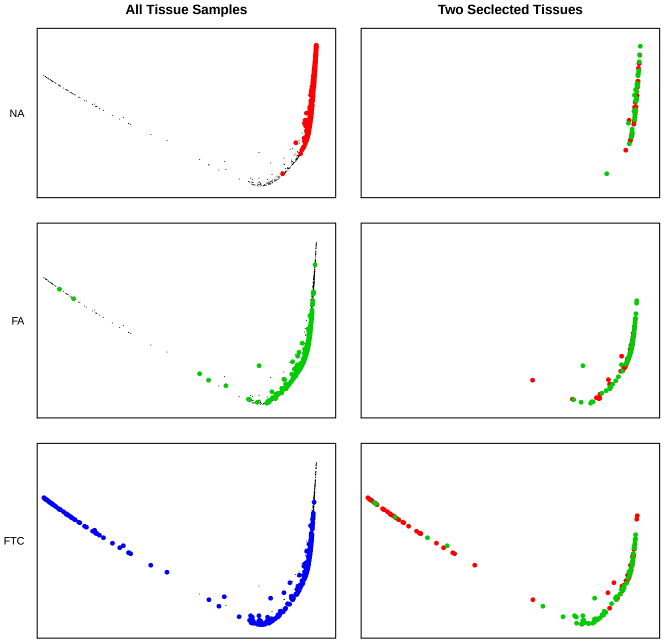

Figure 2: Diffusion map projection of the cancer pathology data. Left column: all nuclei for all tissue sample plotted by class. Right column: two selected samples from each class. We see that the classes have highly overlapping distributions, and that each tissue sample does not have the same distribution as the corresponding class.

5 Further improvement with Kernel Logistic Regression

5.1 Introduction to KLR

As shown in the previous section, a one nearest neighbour classifier produces comparable result to SVM. Although one nearest neighbor is a robust method which produces highly non-linear solutions, it is not stable. Small changes in the starting points of kmeans might dramatically affect the result. Further, the weighting scheme is not directly tied to the classification result. Rather, it is tied to a pair of heuristic quantities that we believe are related to the performance of a nearest neighbor classified.

We now consider Kernel Logistic Regression (KLR), a method which gives stable performance and allows us to directly incorporate weighting into a classification based objective function. Recall that that we obtain a distance matrix between data sources after the binning step. KLR methods built on top of the distance matrix result in non-linear and stable solution.

We now give the basics of Kernel Logistic Regression. The nonparametric model is

$$p(Y=1|X) = \frac{e^{f(x)}}{1+e^{f(x)}}.$$

Where $f \in \mathcal{H}_K$, a Reproducing Kernel Hilbert Space (RKHS). As is typical with kernel methods, we must regularize the function f to prevent overfitting. Putting this together with the log-likelihood, we obtain the following regularized risk functional:

$$J_n(f,\lambda) = \sum_{i=1}^{n} (\log(1+e^{f(x_i)}) - y_i f(x_i)) + \frac{\lambda}{2}||f||_K^2.$$

Representing this in dual form leads to the following objective function:

$$J_n(\alpha,\lambda) = \sum_{i=1}^{n} (\log(1+\exp(K_i\alpha)) - y_i K_i \alpha) + \frac{\lambda_2}{2}\alpha^T K \alpha. \tag{12}$$

Here $\alpha \in \mathbb{R}^n$ is a vector of coefficients that represents the function f on the observed data. The Gram matrix K is a matrix of inner products between all pairs of training data sources. We can use our pairwise distance between sources along with a Gaussian radial basis kernel to obtain the Gram matrix. Explicitly, for some fixed $\sigma > 0$, entry i, j of the Gram matrix is: $K_{ij} = e^{D_{ij}/\sigma}$, where D_{ij} is the distance between source i and j. We can use Newton's method to compute α iteratively.

5.2 Incorporating Weighting in KLR

We now demonstrate a method for incorporating a bin weighting scheme to KLR. Recall that $K_{ij} = e^{D_{ij}/\sigma}$. In other words, K is a function of D, the distance between different data sources. Additionally, if we allow for bin weights, $D_{ij}^2 = ((h_i - h_j) \circ \mathbf{w})A((h_i - h_j) \circ \mathbf{w})^T$, so D is in turn a function of $\mathbf{w}$, the weight of bins. Therefore, we can write K as $K(\mathbf{w})$. We then can rewrite the objective function in terms of both α and $\mathbf{w}$. This optimization problem can be formulated as:

$$\min_{\mathbf{w},\alpha} \mathcal{J}(\mathbf{w}, \alpha)$$
$$\text{Subject to: } ||\mathbf{w}||_2 < q; \mathbf{w} > 0. \tag{13}$$

We require the regularity condition $||\mathbf{w}||_2 < q$ so that the weights do not overfit and explode. The second constraint comes from the belief that each entry of the weights should be positive, otherwise the weights have no interpretable meaning.

The optimization problem in Equation 13 can be solved via gradient descent. Note that we may tackle the ℓ_2 constraint on $\mathbf{w}$ by writing the Lagrangian form: $J = \mathcal{J} + \lambda_2||\mathbf{w}||_2$. The second constraint is handled using barrier methods. We now give the gradient equations for $\mathbf{w}$ and α:

$$\frac{\partial}{\partial w_p} J = \sum_{i=1}^{n} \left[\frac{\exp(\sum_j K_{ij}\alpha_j)}{1 + \exp(\sum_j K_{ij}\alpha_j)} \sum_{j=1}^{n} \left(\alpha_j \frac{\partial}{\partial w_p} K_{ij} \right) + y_i \sum_{j=1}^{n} \left(\alpha_j \frac{\partial}{\partial w_p} K_{ij} \right) \right] + \frac{\lambda_2}{2} \alpha^T \frac{\partial}{\partial w_p} K\alpha \tag{14}$$

$$\frac{\partial}{\partial \alpha} J = K^T \left(\frac{\exp(K\alpha)}{1 + \exp(K\alpha)} - \mathbf{y} \right) + \lambda K\alpha \tag{15}$$

It remains to state the partial derivative $\frac{\partial K_{ij}}{\partial w_p}$. The kernel is a function of the pairwise distance between data sources. The weighting will change the pairwise distance. Note that, by the chain rule:

$$\frac{\partial K_{ij}}{\partial w_p} = \frac{\partial K_{ij}}{\partial D_{ij}^2} \frac{\partial D_{ij}^2}{\partial w_p}$$

Therefore, we must figure out the partial derivative $\frac{\partial D_{ij}^2}{\partial w_p}$. Recall that D_{ij} is the distance between ith and jth source.

$$D_{ij}^2 = (h_{ij} \circ \mathbf{w})A(h_{ij} \circ \mathbf{w})^T$$

where $h_{ij} = h_i - h_j$, which is a $1 \times K$ vector, where K is the number of clusters. A is a $K \times K$ matrix, which represents the similarity between each cluster. $\circ$ denotes the Hadamard, or entry wise, product. We can rewrite $D_{ij} = \sum_{p,q} H_{ij,p} A_{pq} h_{ij,q} w_p w_q$. Therefore,

$$\frac{\partial}{\partial w_p} D_{ij^2} = \sum_{q, q \neq p} h_{ij,p} A_{pq} h_{ij,q} w_q + 2h_{ij,p}^2 A_{pp} W_p.$$

And

$$\frac{\partial}{\partial w_p} K_{ij} = \frac{\partial K_{ij}}{\partial D_{ij}} \frac{\partial D_{ij}}{\partial w_p} \tag{16}$$

$$= -\frac{D_{ij}}{\sigma} \exp(-\frac{D_{ij}^2}{2\sigma})(\sum_{q,q\neq p} h_{ij,p} A_{pq} h_{ij,q} w_q + 2h_{ij,p}^2 A_{pp} W_p). \tag{17}$$

Based on these equations, we are able to iteratively solve $\mathbf{w}$ and α jointly using gradient descent:

$$\mathbf{w} = \mathbf{w} - \lambda \frac{\partial J}{\partial w},$$

$$\alpha = \alpha - \lambda \frac{\partial J}{\partial \alpha}.$$

5.3 Prelimary Results

We now compare the result of KLR with and without weighting. In order to rule out the effect of the binning step, we compare the result of KLR+weighting and KLR under the same binning scheme. We further compare across different values of K, the number of bins. We repeat the procedure 250 times for different starting points for each K. Note that our problem involves three classes, so we classify each tissue sample in two steps. In the first step we classify a tissue sample as normal or abnormal. In other words, in the first step, we consider FA and FTC as the same class. If a tissue sample is labeled as abnormal, we next determine what type of caner it is in the second step (FA or FTC). We therefore attack the 3-class classification problem as a pair of two class problems.

As with nearest neighbors, the KLR classifiers are able to achieve perfect or near perfect classification. It is not possible to compare the performance only based on classification rate. However, for KLR, the method returns a probability of assignment rather than a class label. Therefore, we consider Brier score, which measures the average squared deviation between prediction and the true label. The score is defined as $\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2$ and the smaller the score is, the better. This score roughly measures the confidence of the classification. We report the Brier scores for both steps.

Note that for $K = 8$, we are able to obtain perfect classification results for both KLR and weighted KLR in nearly every binning simulation. The result of the analysis is shown in Figure 3 and Figure 4. As we can see, for all choices of K that we considered, weighting improves the result dramatically. In fact, for any particular run, the weighted version always has a lower Brier score. We also observe that, as K increases, the Brier score gets worse for both methods. We can roughly conclude that a good choice of K is around 8 for this problem.

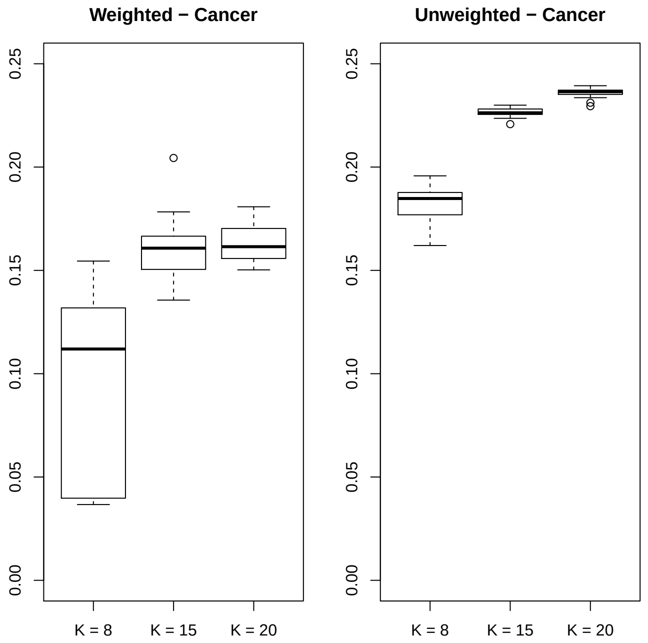

Figure 3: The Brier score for weighted and unweighted KLR in labeling whether a tissue sample is cancerous or not. As we can see, for all K's, weighing dramatically improves the performance. The box plot is obtained over 250 runs.

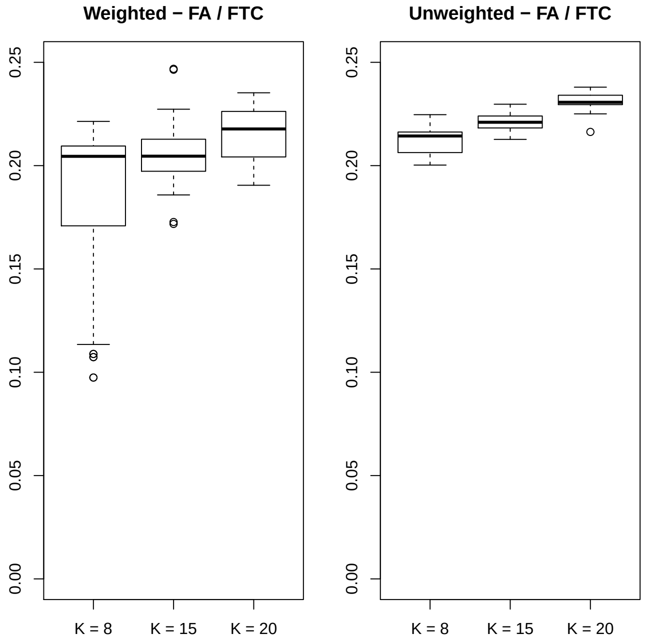

Figure 4: The Brier score for weighted and unweighted KLR in labeling whether a cancer tissue is FA or FTC. As we can see, for all K's, weighing improves the performance, even though the improvement is not as big as the in the previous plot. The boxplot is obtained over 250 runs.

6 Summary and future work

In this paper we focuses on a pathology dataset with special structure. In this application, instead of labeling each data point, we need to label a collection of data points. We name these collection of data points data sources. We suggested a practical way to create the quantization, and compare quantized distributions. We considered different classifiers on the resulting distance matrix. The technique has room for improvement with additional schemes based on different quantizations, weighting approaches, or distance measures. In our empirical studies, we used very straightforward approaches in these aspects. Our quantization approach shows good promise in our data applications, achieving comparable or superior results to state-of-the-art methods. When comparing two data sources, competing methods only use data from the two sources of interest. Our method benefits greatly from using all of the data to make such a comparison, a claim we have backed empirically.

In the future, we hope to prove such a claim theoretically. We would also like to consider different weighting schemes – including different optimization functions as well as different penalties. We would like to understand the impact of weighting on the hilbert kernel space. Also, the Brier score will help us on choosing the optimal K for our binning strategy. Note that the choice of K is important for the choice of support so it is a question of interest. Also note that we consider ℓ_2 penalty here – we will consider ℓ_1 penalty for a sparse solution in the future. In our context, a sparse solution is of great interest – some bins might simply be dropped.

References

[1] O. Abulafia and D. M. Sherer. Automated cervical cytology: meta-analysis of the performance of the PAPNET system. *Obstet. Gynecol. Surv.*, 54:253–264, 1999.

[2] Shameek Biswas, Laura B. Scheinfeldt, and Joshua M. Akey. Genome-wide insights into the patterns and determinants of fine-scale population structure in humans. *The American Journal of Human Genetics*, 84(5):641–650, 2009.

[3] A Frasoldati, M Flora, M Pesenti, A Caroggio, and R Valcavil. Computer-assisted cell morphometry and ploidy analysis in the assessment of thyroid follicular neoplasms. *Thyroid*, 11(10):941–946, Oct 2001.

[4] Joshua K Hartshorne. Visual working memory capacity and proactive interference. *PLoS ONE*, 3(7):e2716, 2008.

[5] Po-Whei Huang and Cheng-Hsiung Lee. Automatic classification for pathological prostate images based on fractal analysis. *IEEE Trans Med Imaging*, 28(7):1037–50, Jul 2009.

[6] M Ikeguchi, N Sato, Y Hirooka, and N Kaibara. Computerized nuclear morphometry of hepatocellular carcinoma and its relation to proliferative activity. *J Surg Oncol*, 68(4):225–230, Aug 1998.

[7] Wayne Niblack, Ron Barber, William Equitz, Myron Flickner, Eduardo H. Glasman, Dragutin Petkovic, Peter Yanker, Christos Faloutsos, and Gabriel Taubin. The QBIC project: Querying images by content, using color, texture, and shape. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 173–187, 1993.

[8] R. R. Pereira, P. M. Azevedo Marques, M. O. Honda, S.K. Kinoshita, R. Engelmann, C. Muramatsu, and K. Doi. Usefulness of texture analysis for computerized classification of breast lesions on mammograms. *Journal of Digital Imaging*, 20(3):248–255, Sep 2007.

[9] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40:2000, 2000.

[10] S. S. Singh, D. Kim, and J. L. Mohler. Java web start based software for automated quantitative nuclear analysis of prostate cancer and benign prostate hyperplasia. *Biomedical Engineering Online*, 4:1:31, 2005.

[11] Muhammad Atif Tahir and Ahmed Bouridane. Novel round-robin tabu search algorithm for prostate cancer classification and diagnosis using multispectral imagery. *IEEE Trans Inf Technol Biomed*, 10(4):782–793, Oct 2006.

[12] Wei Wang, John A. Ozolek, and Gustavo K. Rohde. Detection and classification of thyroid follicular lesions based on nuclear structure from histopathology images. *Cytometry Part A*, 77:1552–4922, 2010.