Improved kNN Rule for Small Training Sets

Conference Paper · December 2014

DOI: 10.1109/ICMLA.2014.37

2 authors:

Sunsern Cheamanunkul
University of California, San Diego
5 PUBLICATIONS **14** CITATIONS

SEE PROFILE

Yoav Freund
University of California, San Diego
104 PUBLICATIONS **26,184** CITATIONS

SEE PROFILE

Improved kNN Rule for Small Training Sets

Sunsern Cheamanunkul
Department of Computer Science and Engineering
University of California, San Diego
La Jolla, California 92093-0404
Email: scheaman@eng.ucsd.edu

Yoav Freund
Department of Computer Science and Engineering
University of California, San Diego
La Jolla, California 92093-0404
Email: yfreund@eng.ucsd.edu

***Abstract*—The traditional k-NN classification rule predicts a label based on the most common label of the k nearest neighbors (the plurality rule). It is known that the plurality rule is optimal when the number of examples tends to infinity. In this paper we show that the plurality rule is sub-optimal when the number of labels is large and the number of examples is small. We propose a simple k-NN rule that takes into account the labels of all of the neighbors, rather than just the most common label. We present a number of experiments on both synthetic datasets and real-world datasets, including MNIST and SVHN. We show that our new rule can achieve lower error rates compared to the majority rule in many cases.**

I. Introduction

The k-nearest neighbors (k-NN) algorithm is one of the oldest methods of pattern recognition and machine learning. Given an unlabeled instance s, the algorithm finds the k labeled examples that are closest to s according to some measure of distance or divergence. The algorithm then counts the number of times each of the m possible label appears within this set of k elements and predicts with the label that appears the largest number of times. This is called the "plurality vote". In the binary label case ($m = 2$), the plurality vote is equal to the majority vote. In this paper our focus is on problems where $m \gg 2$.

Fix and Hodges [1] show that for sufficiently large datasets the k-NN rule achieves the Bayes error rate r^* under very mild conditions. More precisely, if n denotes the number of examples which grows to infinity $n \to \infty$, and we choose k as a function of n so that $k(n) \to \infty$ and $k(n)/n \to 0$ then the error rate of the k-NN rule approaches the Bayes optimal error rate. Furthermore, Cover and Hart [2] show that, when $k = 1$, the asymptotic error rate of the 1-NN rule is upper bounded by $r^*(2 - \frac{m}{m-1}r^*)$.

However, real-world datasets are always finite and often small. In such cases, the theoretical results give little guidance. Indeed, as we will show, there are good reasons to suggest that rules other than the plurality rule might perform significantly better. The basic intuition is that, in all cases other than the binary case, the plurality vote ignores information present in the counts of labels other than the largest label.

To motivate our approach, consider the following example from handwriting recognition using 5-NN where the distances between examples are computed using a standard elastic matching divergence. Figure 1 shows an instance of a handwritten letter "k" and its 5 nearest neighbors: two "h"s, two "m"s and a single "k". The plurality vote will go for either "h" or "m", both of which are incorrect. To avoid this mistake, we propose an alternative classification rule. The key is to bring into the model the *average neighbor distribution for each letter*. In other words, using the labeled training set, we can estimate a *model* for the distribution of labels of the k neighbors for each label. In most cases, the most common neighbor label will be the same as the label of the queried point. However, the distribution will also capture information about the other labels that tend to be in the k-NN set of a point.

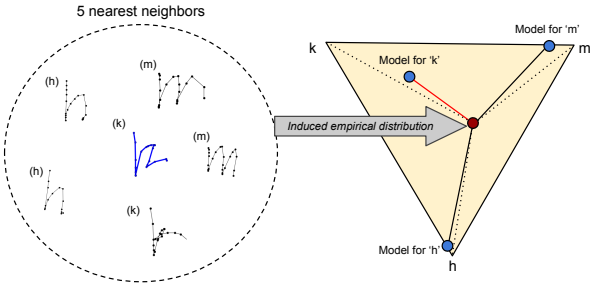


Fig. 1: Problem with the majority rule in the finite N setting. The five nearest neighbors of an example from handwriting recognition are shown on the left with their true labels in parentheses. The majority rule predicts either h or m while the true label is k. However, when the label distribution of the neighborhood is mapped onto the probability simplex shown on the right, a better classification rule is to minimize the KL-divergence (solid lines) between the empirical distribution (red dot) and the typical neighborhood distributions of letter h, m and k (blue dots).

In Figure 1, we represent the model distributions for the letters "k", "h" and "n" as points on a 2D triangular probability simplex. Note that the neighbors for "h" and "m" assign very small probabilities to having the label "k" in their neighborhood, while the letter "k" has a significant probability of having "h" or "m" in its neighborhood. On the same simplex we plot the empirical distribution $(2, 2, 1)$ (indicated by a red dot). By using the Kullback-Leibler divergence, we can recover the fact that the letter is a "k". This reversal of the classification is based on the following logic: while it is true that there are more instances of "h" and "m" in the 5-NN neighborhood, there exists one instance of "k". On the other hand, it is common for "h" and "m" to be in the neighborhood set of a "k" while it is very rare for a "k" to be in the neighborhood of an "h" or an "m". The result is that "k", even though it is in the minority, is a better explanation of the observed labels.

Our approach is related to work on learning label embeddings [3], [4]. The main difference is that our approach is far simpler, does not require any convex optimizations and can be seamlessly integrated into the k-NN framework. Another related work is [5] which introduces a bias term to the likelihood ratio testing which is justified by the difference between the estimated and the true class conditional probability.

This paper is organized as follows. In Section II, we describe the framework and the notations. In Section III, we describe our approach and justification. In Section IV, we present experiments comparing our approach with the traditional k-NN algorithm using both synthetic data and real-world data. Then, we discuss the results in Section V and conclude the paper in Section VI.

II. Background

Let $\mathcal{S} = \{(x_1, y_1,) \dots (x_N, y_N)\}$ be a set of training examples where each instance x_i comes from an example space $\mathcal{X}$ of which the distance between any two examples is measured by $d(\cdot, \cdot)$. Without loss of generality, we assume that each label y_i takes on a value from $\mathcal{Y} = \{1, 2, \dots, m\}$. To simplify the analysis, we assume that the distribution of classes is uniform and the number of examples per class is denoted by n.

Let $\mathcal{N}_k(x)$ denote the neighborhood of size k of an example $x \in \mathcal{X}$ with respect to the distance measure d. The traditional k-NN rule predicts the label of an example x with the majority of the labels in $\mathcal{N}_k(x)$. More formally, given x and $\mathcal{N}_k(x)$, we can define an empirical distribution $\widehat{\mathbf{P}}_{(x,\mathcal{S},k)}$ such that, for each $i \in \mathcal{Y}$,

$$\widehat{\mathbf{P}}_{(x,\mathcal{S},k)}(i) = \frac{\#\{\text{occurrences of label } i \text{ in } \mathcal{N}_k(x)\}}{k}$$

The k-NN rule predicts the label $\hat{y}$ such that

$$\hat{y} = \arg\max_{i \in \mathcal{Y}} \widehat{\mathbf{P}}_{(x,\mathcal{S},k)}(i)$$

For any example $x \in \mathcal{X}$, we can consider the true class distribution of x, denoted by $\mathbf{P}_{(x)}$ which is given by, for each $i \in \mathcal{Y}$,

$$\mathbf{P}_{(x)}(i) = \Pr(Y = i | X = x)$$

Under certain assumptions, it is shown in [1] that, for every class label $i \in \mathcal{Y}$,

$$\lim_{\substack{n\to\infty \\ k\to\infty \\ k/n\to 0}} \widehat{\mathbf{P}}_{(x,\mathcal{S},k)}(i) = \mathbf{P}_{(x)}(i)$$

Therefore, the majority rule is asymptotically optimal. However, in the finite sample scenario, it can be sub-optimal due to the discrepancy between the empirical distribution $\widehat{\mathbf{P}}_{(x,\mathcal{S},k)}$ and the true distribution $\mathbf{P}_{(x)}$ as demonstrated in Figure 1.

III. Minimizing KL-Divergence Rule

We propose a new k-NN rule that predicts the class label based on the entire class distribution $\widehat{\mathbf{P}}_{(x,\mathcal{S},k)}$ instead of just the mode (majority) of $\widehat{\mathbf{P}}_{(x,\mathcal{S},k)}$. We refer to this rule as the minimizing KL-divergence rule (MinKL). Given a training set $\mathcal{S}$ and the neighborhood of size k, we define, for each class j, an empirical center distribution $\widehat{\mathbf{Q}}_{(j,\mathcal{S},k)}$ as

$$\widehat{\mathbf{Q}}_{(j,\mathcal{S},k)} = \frac{\sum_{(x,j)\in\mathcal{S}_j} \widehat{\mathbf{P}}_{(x,\mathcal{S},k)}}{|\mathcal{S}_j|}$$

where $\mathcal{S}_j = \{(x, y) \in \mathcal{S} | y = j\}$ consists of all examples with class label j. To classify a new example x, the empirical class distribution $\widehat{\mathbf{P}}_{(x,\mathcal{S},k)}$ is compared to each of the center distributions $\widehat{\mathbf{Q}}_{(j,\mathcal{S},k)}$ with respect to the KL-divergence $D_{\mathrm{KL}}(\widehat{\mathbf{P}}_{(x,\mathcal{S},k)} || \widehat{\mathbf{Q}}_{(j,\mathcal{S},k)})$ and the class label that minimizes the distance is then predicted. More formally, the predicted label $\hat{y}$ is given by

$$\hat{y} = \arg\min_{j \in \mathcal{Y}} D_{\mathrm{KL}}(\widehat{\mathbf{P}}_{(x,\mathcal{S},k)} || \widehat{\mathbf{Q}}_{(j,\mathcal{S},k)})$$

where the KL-divergence between two discrete distributions p and q is defined as

$$D_{\mathrm{KL}}(p||q) = \sum_i p(i) \log \frac{p(i)}{q(i)}$$

The training algorithm and the prediction rule are summarized in Algorithm 1 and Algorithm 2 respectively.

Algorithm 1 The MinKL k-NN rule: Training

Require: Training set $\mathcal{S}$ and k
Output: The center distributions $\widehat{\mathbf{Q}}_j$ for all $j \in \mathcal{Y}$
1: $\widehat{\mathbf{Q}}_j \leftarrow \vec{0}$ for $j \in \mathcal{Y}$
2: **for** each example $(x, j) \in \mathcal{S}$ **do**
3: $\quad \widehat{\mathbf{Q}}_j \leftarrow \widehat{\mathbf{Q}}_j + \widehat{\mathbf{P}}_{(x,\mathcal{S},k)}$
4: **end for**
5: $\widehat{\mathbf{Q}}_j \leftarrow \widehat{\mathbf{Q}}_j / |\mathcal{S}_i|$ for all $j \in \mathcal{Y}$

Algorithm 2 The MinKL k-NN rule: Prediction

Require: Training set $\mathcal{S}$,
A test example x,
The center distributions $\widehat{\mathbf{Q}}_j$ for all $j \in \mathcal{Y}$
Output: Predicted label $\hat{y}$
1: $\hat{y} = \arg\min_{i \in \mathcal{Y}} D_{\mathrm{KL}}(\widehat{\mathbf{P}}_{(x,\mathcal{S},k)} || \widehat{\mathbf{Q}}_i)$

To analyze our approach in the finite sample setting, we introduce a few more notations. Let $\overrightarrow{\mathbf{P}}_{(x,k)}$ denote the expected class distribution of an example x induced by a neighborhood of size k, which is given by

$$\overrightarrow{\mathbf{P}}_{(x,k)} = \mathbf{E}_{\mathcal{S}}[\widehat{\mathbf{P}}_{(x,\mathcal{S},k)}]$$

Similarly, let $\overrightarrow{\mathbf{Q}}_{(j,k)}$ denote the expected center distribution for examples of class j defined by

$$\overrightarrow{\mathbf{Q}}_{(j,k)} = \mathbf{E}_{\mathcal{S}}[\widehat{\mathbf{Q}}_{(j,\mathcal{S},k)}]$$

Note that the expectation is taken over all possible training sets of size N.

Ideally, the empirical distribution $\widehat{\mathbf{P}}_{(x,\mathcal{S},k)}$ should be compared to the expected center distribution $\overrightarrow{\mathbf{Q}}_{(j,k)}$. However, in practice, we use $\widehat{\mathbf{Q}}_{(j,\mathcal{S},k)}$ as an estimate for $\overrightarrow{\mathbf{Q}}_{(j,k)}$. This is

reasonable because $\widehat{\mathbf{Q}}_{(j,\mathcal{S},k)}$ for each class j is estimated from a relatively large amount of examples in the training set.

To justify our approach, we begin by defining the KL-divergence between a distribution p and a set of distributions E. Then, we prove a lemma to show that an empirical distribution $\hat{p}$ induced by drawing examples from a true distribution $p \in E$ is likely to be "close" to the set E in the KL-divergence sense.

Definition 1: The *KL-divergence* between any distribution p and a set of distributions E is defined as

$$D_{\mathrm{KL}}(p||E) \doteq \arg\min_{q \in E} D_{\mathrm{KL}}(p||q)$$

Lemma 2: Let E be a set of distributions over some domain χ. For any $p \in E$, let x^n be a sample of size n drawn from p and let $\hat{p}$ be the empirical distribution induced by x^n. Then,

$$\Pr\{D_{\mathrm{KL}}(\hat{p}||E) > \epsilon\} \leq (n+1)^{|\chi|} e^{-n\epsilon}$$

Proof: Theorem 11.2.1 in [6] states that, for any distribution p and any $\epsilon > 0$,

$$\Pr\{D_{\mathrm{KL}}(\hat{p}||p) > \epsilon\} \leq (n+1)^{|\chi|} e^{-n\epsilon}$$

Since $p \in E$, by definition,

$$\Pr\{D_{\mathrm{KL}}(\hat{p}||E) > \epsilon\} \leq \Pr\{D_{\mathrm{KL}}(\hat{p}||p) > \epsilon\}$$

So we have

$$\Pr\{D_{\mathrm{KL}}(\hat{p}||E) > \epsilon\} \leq (n+1)^{|\chi|} e^{-n\epsilon}$$

■

Suppose $E_1, E_2, \ldots, E_m$ be sets of distributions such that $\bigcap E_i = \emptyset$. We assume that a sample of size n, x^n, is generated by the following process:

1) Class i^* is chosen with probability π_{i^*}.
2) A distribution p is chosen such that $\Pr\{p \in E_{i^*}\} \geq 1 - \delta$.
3) x^n is sampled from the distribution p.

Theorem 3: Let $E_1, E_2, \ldots, E_m$ be sets of distributions such that $\bigcap E_i = \emptyset$. Suppose x^n is generated according to the above process with respect to the correct class i^* and the true distribution p where $\Pr\{p \in E_{i^*}\} \geq 1 - \delta$. Let $\hat{p}$ denote the empirical distribution induced by x^n. Then,

$$\Pr\{D_{\mathrm{KL}}(\hat{p}||E_{i^*}) > min_{j \neq i^*} D_{\mathrm{KL}}(\hat{p}||E_j) \mid p \in E_{i^*}\} \leq (m-1)(n+1)^{|\chi|} e^{-n\Delta}$$

where $\Delta \doteq \min_{i,j;j \neq i} \min_{q \in \mathcal{P}} \max(D_{\mathrm{KL}}(q||E_j), D_{\mathrm{KL}}(q||E_i))$

Proof: By applying Lemma 2

$$\begin{aligned}
&\Pr\{D_{\mathrm{KL}}(\hat{p}||E_{i^*}) > min_{j;j \neq i^*} D_{\mathrm{KL}}(\hat{p}||E_j) \mid p \in E_{i^*}\} \\
&\leq \sum_{j;j \neq i^*} \Pr\{D_{\mathrm{KL}}(\hat{p}||E_{i^*}) > D_{\mathrm{KL}}(\hat{p}||E_j) \mid p \in E_{i^*}\} \\
&\leq \sum_{j;j \neq i^*} \Pr\{D_{\mathrm{KL}}(\hat{p}||E_{i^*}) > \Delta) \mid p \in E_{i^*}\} \\
&\leq (m-1)(n+1)^{|\chi|} e^{-n\Delta}
\end{aligned}$$

■

Algorithm 3 Theoretical MinKL

Require: The number of classes m, the sets of distributions $E_1, E_2, \ldots, E_m$, and the empirical distribution $\hat{p}$
Output: Predict i^*

```
for all i ∈ 1, ..., m do
    p_i ← arg min_{q∈E_i} D_KL(p̂||q)
end for
i* = arg min_i D_KL(p̂||p_i)
```

TABLE I: A summary of the datasets.

DATASET	NO. OF CLASSES	NO. OF TRAIN EX.	NO. OF TEST EX.
SYN-1	10	UP TO 1600	10000
SYN-2	64	UP TO 6400	6400
SYN-3	10	UP TO 1600	10000
URIGHT	26	9945	-
MNIST	10	60000	10000
SVHN	10	73257	26032

Using Theorem 3, we can justify Algorithm 3 under the assumptions about the data generation process. It is worth noting that Algorithm 3 is not quite the same as Algorithm 2 in that, in Algorithm 2, we only compare the KL distance from the empirical distribution $\hat{p}$ to a single center distribution $\widehat{\mathbf{Q}}_j$. However, we can show that if each E_j contains only distributions that are α-close to the center $\widehat{\mathbf{Q}}_j$, then Algorithm 2 is also justified by applying Lemma 5.

Definition 4: For any $\alpha > 0$, a distribution p is said to be α-close to another distribution q if and only if

$$(1-\alpha)q(x) \leq p(x) \ \forall x \in \chi$$

Lemma 5: If every $q \in E_j$ is α-close to $\widehat{\mathbf{Q}}_j$, then

$$D_{\mathrm{KL}}(p||E_j) \leq D_{\mathrm{KL}}(p||\widehat{\mathbf{Q}}_j) + \log\frac{1}{1-\alpha}$$

for any distribution p

Proof:

$$\begin{aligned}
D_{\mathrm{KL}}(p||E_j) &= \min_{q \in E_j} D_{\mathrm{KL}}(p||q) \\
&= \min_{q \in E_j} \sum_x p(x) \log\frac{p(x)}{q(x)} \\
&\leq \min_{q \in E_j} \sum_x p(x) \log\frac{p(x)}{(1-\alpha)\widehat{\mathbf{Q}}_j(x)} \\
&\leq \sum_x p(x) \log\frac{p(x)}{(1-\alpha)\widehat{\mathbf{Q}}_j(x)} \\
&\leq D_{\mathrm{KL}}(p||\widehat{\mathbf{Q}}_j) + \log\frac{1}{1-\alpha}
\end{aligned}$$

■

IV. EXPERIMENTS

In this section, we describe experiments we have performed with both synthetic data and real-world data. For each dataset, we compare the error rates of the k-NN with the minimizing KL-divergence rule (MinKL) to those of the k-NN with the majority rule (Majority) under various conditions. A summary of the datasets is given in Table I.

A. Synthetic data

We performed 3 experiments using synthetic data that can be described as follows. Each example x is a point inside a wrap-around d-dimensional hypercube of size b, namely $x \in [0, b-1]^d$. The instances of each class are generated by a normal distribution with mean located at each integer lattice point of the hypercube and a covariance matrix $\sigma \mathbf{I}_d$. Thus, the total number of classes is b^d. The distribution of the classes in each dataset is uniform. In Figure 2, the generating distributions of each dataset are shown in the left-most column. The Manhattan distance (L_1 norm) is used for measuring the distance between examples.

In our first experiment, we generated a dataset called SYN-1 using the following parameters: $b = 10, d = 1$ and $\sigma = 1.5$. SYN-1 was intended to mimic the situation described in Figure 1. The number of classes in SYN-1 is 10. In the second experiment, we generated another dataset called SYN-2 using the following parameters: $b = 4$, $d = 3$ and $\sigma = 0.4$. SYN-2 has a very similar structure to SYN-1 but it is more complex with the total of 64 classes. In our third experiment, we generated yet another dataset called SYN-3. Similar to SYN-1, each instance of SYN-3 is one-dimensional. However, the generating distribution for class i is a mixture of two normal distributions centered at i and $i + 3$ and the mixing coefficient is 0.8 and 0.2 respectively. SYN-3 is intended for simulating when a single center distribution may not be sufficient.

In Figure 2, we compare the error rates of MinKL and Majority using different n and k for each dataset. For each n, we ran both MinKL and Majority for k ranged. The center column of Figure 2 shows the error rates for different k when n is fixed at 20 per classes for each dataset. Then, for each n, the best error rate of both MinKL and Majority over k are shown in the right-most column of Figure 2. The error rates of both MinKL and Majority converge to the Bayes error as n increases. In SYN-1 and SYN-2, MinKL converges faster than Majority and is able to attain lower error rates especially when n is small. However, in SYN-3, MinKL has higher average error rates than Majority for when n is small.

B. uRight

The uRight dataset contains handwriting trajectories of the 26 lowercase English characters. We collected the handwriting data from 15 different users writing isolated lowercase English characters on a touch screen of a mobile phone with their fingers. Each example is a sequence of (x, y, t) where x and y are the (x, y)-coordinates and t is the timestamp of each sample point. Figure 3 shows some examples of the handwriting trajectories. There are 9945 examples in the dataset and the distribution of the class labels is fairly uniform. The similarity between two examples is measured by the dynamic time warping (DTW) distance [7].

Using $k = 5$, the average error rates of MinKL and Majority for each user are summarized in Figure 4. According to the paired t-test, the average error rate of MinKL (3.76%) is significantly smaller than the average error rate of Majority (5.86%) with p-value < 0.001. Figure 5 displays some of the examples that were misclassified by Majority but correctly classified by MinKL.

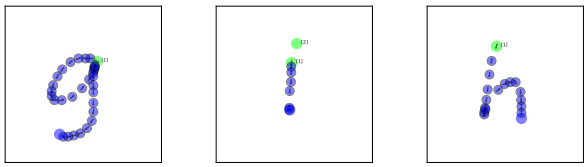


Fig. 3: Some examples from the uRight handwriting dataset.

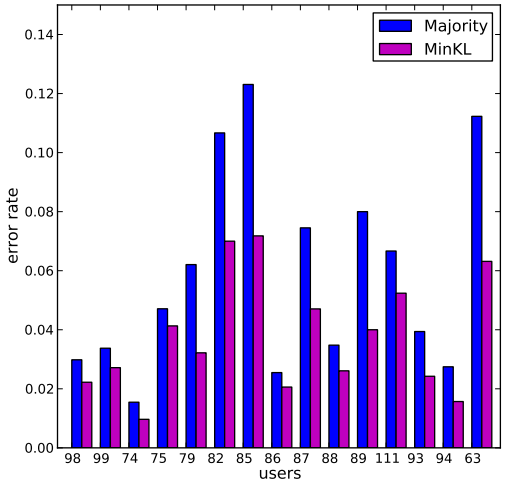


Fig. 4: The average error rates of MinKL and Majority for each user.

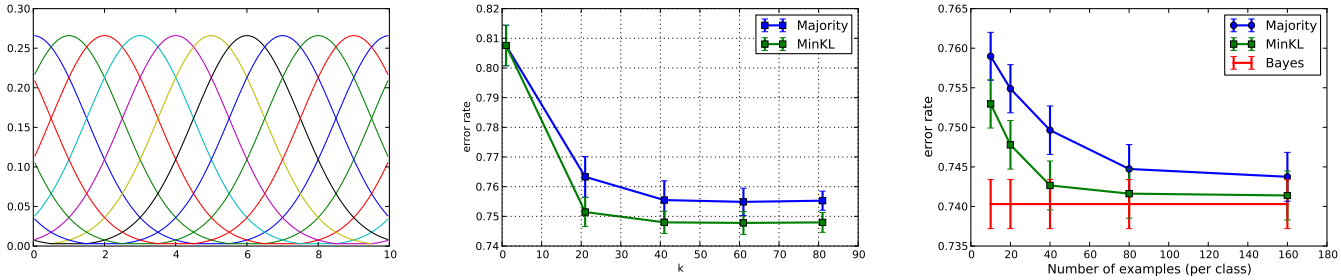
C. MNIST

The MNIST dataset [8] contains images of handwritten digits. Each example is a 28x28 grayscale image. There are 60000 training examples and 10000 test examples included in the dataset. We preprocessed the data by de-skewing and downsampling the images. After the preprocessing, we ran PCA on the training data. The feature vector of each example corresponds to the coefficients of the first 100 PCA components. The Euclidean distance is used as the similarity measure in the neighborhood calculation.

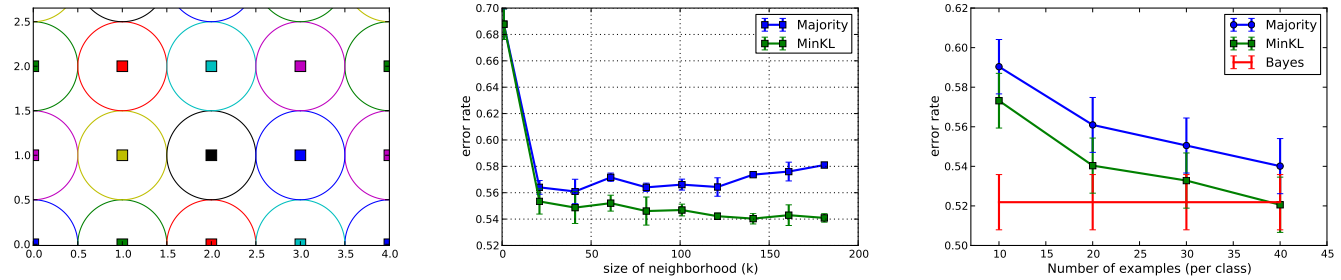
The test error rates we obtained from our experiment are comparable to what reported in [8]. The performance of both MinKL and Majority are very similar for this dataset. The lowest error rate of 1.89% for Majority and 1.90% for MinKL was obtained when $k = 5$. Figure 6 shows the test error rates of both MinKL and Majority obtained using different k.

D. SVHN

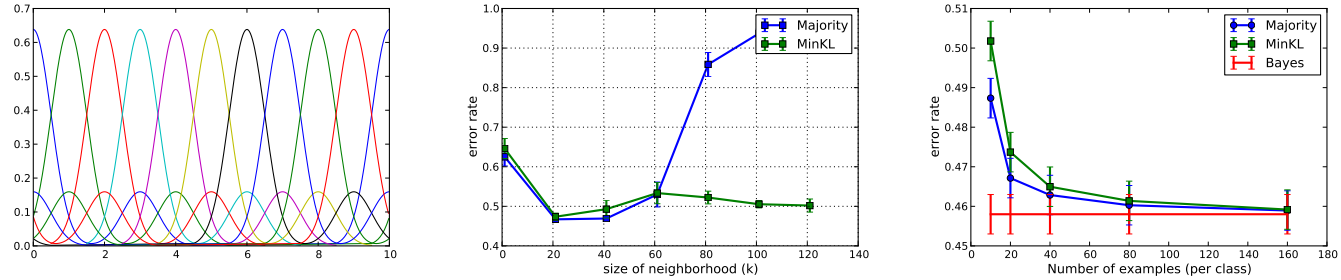
The SVHN dataset [9] contains images of digits taken from the Google street view data. It is considered a harder dataset than MNIST due a higher degree of variations. Each example in SVHN is a 32x32 RGB image. There are 73257 training examples and 26032 test examples included in the dataset. We computed, for each example, the HOG features [10] using the block size of 4x4 with 8 orientations per block. The Manhattan distance is used as the similarity measure in the neighborhood calculation.



(a) SYN-1: (left) Data models, (center) error rate vs. k while n fixed, (right) error rate vs. n while k fixed



(b) SYN-2: (left) Data models, (center) error rate vs. k while n fixed, (right) error rate vs. n while k fixed



(c) SYN-3: (left) Data models, (center) error rate vs. k while n fixed, (right) error rate vs. n while k fixed

Fig. 2: Results from the synthetic data experiment.

In [9], the test error rate for HOG features combined with an SVM is reported to be around 15%. In our experiment, the test error rates of both MinKL and Majority are between 16% to 17% with MinKL performing slightly better than Majority at every $k > 1$. Figure 6 shows the test error rates of both MinKL and Majority obtained using different k.

V. Discussion

In our experiments with SYN-1 and SYN-2, we observed that MinKL performs significantly better than Majority when n is small. This result also confirms our intuition we have on the example shown in Figure 1. Our explanation for this boost in performance is the fact that, for small n (implied a small k), the majority rule is prone to error because the prediction is based on solely the majority of the empirical class distribution $\widehat{\mathbf{P}}_{(x,\mathcal{S},k)}$ induced from a relatively small k; while the MinKL rule makes the prediction based on the entire class distribution.

According to our analysis in Section III, we show that our approach will perform optimally when the data can be roughly modeled by a single center distribution. In SYN-3, we deliberately designed the dataset so that this assumption does not hold. As expected, the error rates of our approach are inferior to those of the majority rule even when n is small. A natural workaround is to increase the richness of the center distribution model by allowing multiple centers per class.

For MNIST, the performance gap between the MinKL rule and the majority rule is very small, especially for small k. This is due to the fact that the center distributions for the MNIST dataset are very close the dirac delta function in which case the MinKL reduces to the majority rule.

Another factor that plays a role in the performance of MinKL is the distances between the center distributions. If they are far away from each other, then we expect our approach to work well. We observed that the center distributions of the uRight dataset are more spread out than those of the SVHN dataset and the MinKL rule performed better on the uRight dataset than on the SVHN dataset.

Technically, the minimizing KL-divergence rule can be applied to other classification algorithms as well. The k-NN algorithm is known to be computational expensive in classifying a

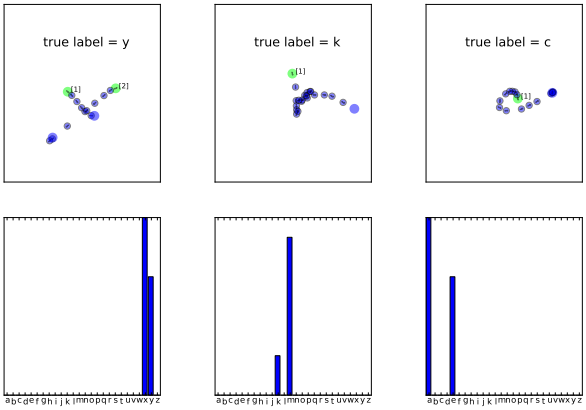
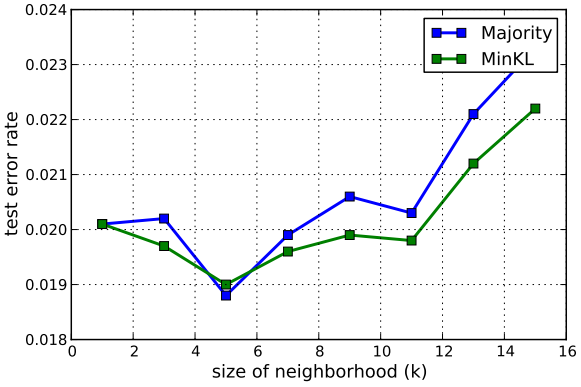


Fig. 5: Some examples misclassified by Majority but correctly classified by MinKL. Underneath each handwriting trajectory, the corresponding empirical distribution induced by its 5 neighbors is shown.

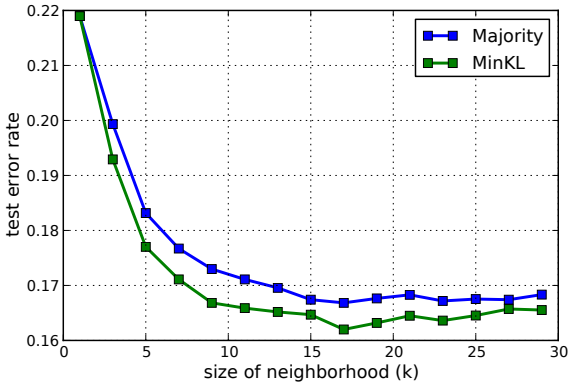
new example. In some applications, it is important to be able to classify new examples quickly. A simple modification to the k-NN algorithm that significantly reduces the classification time is to keep only a small number of representatives per class and discard the rest of the examples. This algorithm is called the k nearest-centroid algorithm (k-NC) where only the k-centroids are kept as the class representatives. In the k-NC, the class distribution $\mathbf{P}_x$ can be estimated by $\widehat{\mathbf{P}}_x(j) = \frac{e^{d(x,C(j))}}{\sum_i e^{d(x,C(i))}}$ and we can then apply the MinKL rule to the class posterior as described above.

VI. Conclusions

We proposed a simple k-NN rule that predicts based on the entire empirical class distribution rather than the majority in the neighborhood. The algorithm can be described as follows. Given a training set, we estimate the center distribution for each class. To classify a new example, we measure the KL-divergence between the empirical distribution induced by the neighborhood and the center distribution of each class. The class that minimized the KL-divergence is then predicted. In a sense, our approach is a simple method for leveraging the class information in the label space. We justified our approach in the finite sample setting under a certain assumption about the data. Finally, we show experimental results comparing the error rates of our approach to the majority rule. We found that our approach managed to outperform the majority rule in many cases.



(a) MNIST



(b) SVHN

Fig. 6: MNIST and SVHN results

References

[1] E. Fix and J. L. Hodges, "Discriminatory analysis, nonparametric discrimination," *USAF School of Aviation Medicine, Randolph Field, Texas, Project 21-49-004, Report 4*, 1951.

[2] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[3] M. Collins and N. Singh-Miller, "Learning label embeddings for nearest-neighbor multi-class classification with an application to speech recognition," *Advances in Neural Information Processing Systems*, pp. 1–9, 2009.

[4] S. Bengio, J. Weston, and D. Grangier, "Label embedding trees for large multi-class tasks," *Advances in Neural Information Processing Systems*, vol. 23, no. 1, pp. 1–10, 2010.

[5] J. Bilmes, G. Ji, and M. Meila, "Intransitive likelihood-ratio classifiers," *Advances in Neural Information Processing Systems*, pp. 0–4, 2001.

[6] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, ser. Wiley Series in Telecommunications, D. L. Schilling, Ed. Wiley, 1991, vol. 6, no. Wiley Series in Telecommunications.

[7] C. Bahlmann and H. Burkhardt, "The writer independent online handwriting recognition system frog on hand and cluster generative statistical dynamic time warping," pp. 299–310, 2004.

[8] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, 1998.

[9] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading Digits in Natural Images with Unsupervised Feature Learning," *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, pp. 1–9, 2011.

[10] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 2005.