Improving the kNN rule for finite training sets

Abstract

Traditionally, the k-NN classification rule predicts a label based on the majority of the labels in the neighborhood. While it can be shown that the majority rule is optimal aymptotically, there is no such guarantee for finite training sets. We propose a simple k-NN rule that incorporates non-majority classes into the prediction. 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.

1. Introduction

We consider the k-nearest neighbor (k-NN) classification rule for multiclass classification problems where the number of classes $m > 2$. Given a set of training examples, the k-NN rule predicts the label of a new example with the majority of the class labels among its k nearest neighbors. Fix and Hodges (Fix & Hodges, 1951) show that, asymptotically, the k-NN rule achieves the Bayes error rate r^* by choosing a large enough k but small compared to the number of examples n. Nonetheless, when n is small, there is no guarantee of how well the k-NN algorithm will perform.

Additionaly, Cover and Hart (Cover & Hart, 1967) show that, when $k = 1$, the aymptotic error rate of the 1-NN rule is upperbounded by $r^*(2 - \frac{m}{m-1} r^*)$. This result suggests that at least a half of the class information is contained in the nearest neighbor. When the nearest neighbor does not have all of the class information, it is possible that the missing class information could be extracted from other examples in the neighborhood.

In this paper, we propose a modification to the k-NN rule which potentially leverages additional information from the non-majority class labels in the neighborhood to improve the classification accuracy. Our approach makes a prediction based on the entire distribution of the class labels in the neighborhood instead of just the majority. While the majority rule works well in most cases, it completely ignores the information from the minority classes which, in some cases, can contain crucial information.

To motivate our approach, consider the following example. Suppose there are 3 classes of examples where each class is generated according to each of the one-dimensional normal distributions depicted in Figure 1. Even though the example x in Figure 1 is of class A but it is likely that the majority of class labels in the neighborhood of x is class B. In such case, the majority rule will predict class B. However, if we consider the rest of the examples in the neighborhood of x and we rarely observe examples from class C, then a better prediction would have been class A.

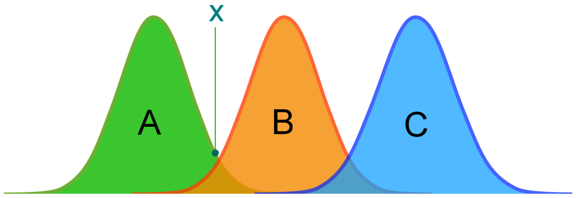


Figure 1. A toy example showing that the majarity rule (ignoring information about class C in the neighborhood) can be suboptimal. Based on the majority rule, the example x of class A can be misclassified as class B. However, the rare occurences of examples of class C in the neighborhood of x can provide useful information regarding the true label of x.

Our approach is related to work on learning label embeddings (Collins & Singh-Miller, 2009; Bengio et al., 2010). The main difference is that our approach is far simpler, does not require any convex optimizations and can be seemlessly integrated into the k-NN framework. Another related work is (Bilmes et al., 2001) 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 2, we describe the framework and the notations. In Section 3, we describe our approach and justification. In Section 4, 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 5 and conclude the paper in Section 6.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

2. 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

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

3. 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 $D_{\text{KL}}(\widehat{\mathbf{P}}_{(x,\mathcal{S},k)} || \widehat{\mathbf{Q}}_{(j,\mathcal{S},k)})$ and the class label that minimizes the distance is then predicted. More formally, the predicted label $\hat{y}$ is given by

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

where the KL-divergence between two discrete distributions $\mathbf{p}$ and $\mathbf{q}$ is defined as

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

A summary of the algorithm is given in Algorithm 1 and Algorithm 2.

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 $j \in \mathcal{Y}$
$\widehat{\mathbf{Q}}_j \leftarrow \vec{0}$ for $j \in \mathcal{Y}$
for each example $(x, j) \in \mathcal{S}$ do
    $\widehat{\mathbf{Q}}_j \leftarrow \widehat{\mathbf{Q}}_j + \widehat{\mathbf{P}}_{(x,\mathcal{S},k)}$
end for
$\widehat{\mathbf{Q}}_j \leftarrow \widehat{\mathbf{Q}}_j / |\mathcal{S}_i|$ 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 $\hat{y}$
$\hat{y} = \arg\min_{i \in \mathcal{Y}} D_{\text{KL}}(\widehat{\mathbf{P}}_{(x,\mathcal{S},k)} || \widehat{\mathbf{Q}}_i)$
```

3.1. Analysis

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 (Fix & Hodges, 1951) 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 suboptimal due to the discrepancy between the empirical distribution $\widehat{\mathbf{P}}_{(x,\mathcal{S},k)}$ and the true distribution $\mathbf{P}_{(x)}$ as demonstrated by the toy example in Figure 1.

To analyze our approach in the finite sample setup, 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.

For any training set $\mathcal{S}$ and for any k, there exists some $\delta_k > 0$ such that

$$D_{\mathrm{KL}}(\overrightarrow{\mathbf{P}}_{(x,k)}||\widehat{\mathbf{Q}}_{(j,\mathcal{S},k)}) < \delta_k \ \forall \ (x,j) \in \mathcal{S}$$

When $\delta_k = 0$, our algorithm can be justified by a direct application of Lemma 1 and Collorary 2.

Let $y^k = [y_1, \ldots, y_k]$ denote a sample of size k drawn IID from a fixed distribution over Y and let $\mathbf{P}_{y^k}$ denote the empirical distribution induced by the sample y^k. Specifically,

$$\mathbf{P}_{y^k}(i) = \frac{\#\{\text{occurrences of } i \text{ in } y^k\}}{k}$$

Lemma 1. *For any distribution $\mathbf{Q}$ and for any sample y^k (not neccessarily drawn from $\mathbf{Q}$), the likelihood of y^k drawn from $\mathbf{Q}$ is given by*

$$\mathbf{Q}(y^k) = 2^{-k(H(\mathbf{P}_{y^k})+D_{\mathrm{KL}}\mathbf{P}_{y^k}||\mathbf{Q}))}$$

Proof.

$$\begin{aligned}
\mathbf{Q}(y^k) &= \prod_{l=1}^{k} \mathbf{Q}(y_l) \\
&= \prod_{j\in\mathcal{Y}} Q(j)^{n\mathbf{P}_{y^k}(j)} \\
&= \prod_{j\in\mathcal{Y}} 2^{n\mathbf{P}_{y^k}(j)\log \mathbf{Q}(j)} \\
&= \prod_{j\in\mathcal{Y}} 2^{n(\mathbf{P}_{y^k}(j)\log \mathbf{Q}(j) - \mathbf{P}_{y^k}(j)\log \mathbf{P}_{y^k}(j) + \mathbf{P}_{y^k}(j)\log \mathbf{P}_{y^k}(j))} \\
&= 2^{k\sum_{j\in\mathcal{Y}}(-\mathbf{P}_{y^k}(j)\log\frac{\mathbf{P}_{y^k}(j)}{\mathbf{Q}(j)} + \mathbf{P}_{y^k}(j)\log \mathbf{P}_{y^k}(j))} \\
&= 2^{k(-D_{\mathrm{KL}}(\mathbf{P}_{y^k}||\mathbf{Q}) - H(\mathbf{P}_{y^k}))}
\end{aligned}$$

□

Collorary 2. *Given a set of distributions $\mathcal{Q} = \{\mathbf{Q}_1, \mathbf{Q}_2, \ldots, \mathbf{Q}_m\}$ and a sample y^k drawn from any distribution, the likelihood of y^k is maximized under $\mathbf{Q}_{i^*}$ if and only if the KL-divergence from $\mathbf{P}_{y^k}$ to $\mathbf{Q}_{i^*}$ is minimized.*

$$i^* = \arg\max_{i\in\mathcal{Y}} \log \mathbf{Q}_i(y^k) = \arg\min_{i\in\mathcal{Y}} D_{\mathrm{KL}}(\mathbf{P}_{y^k}||\mathbf{Q}_i)$$

Proof. Applying Lemma 1, we have

$$\begin{aligned}
\mathbf{Q}_i(y^k) &= 2^{-n(H(\mathbf{P}_{y^k})+D_{\mathrm{KL}}\mathbf{P}_{y^k}||\mathbf{Q}_i))} \\
\log \mathbf{Q}_(y^k) &= -n(H(\mathbf{P}_{y^k}) + D_{\mathrm{KL}}(\mathbf{P}_{y^k}||\mathbf{Q}_i)) \\
\arg\max_{i\in\mathcal{Y}} \log \mathbf{Q}(y^k) &= \arg\max_{i\in\mathcal{Y}} -nD_{\mathrm{KL}}(\mathbf{P}_{y^k}||\mathbf{Q}_i)) \\
&= \arg\min_{i\in\mathcal{Y}} D_{\mathrm{KL}}(\mathbf{P}_{y^k}||\mathbf{Q}_i))
\end{aligned}$$

□

However, when $\delta_k > 0$, we need to enforce a stronger assumption about the data in the training set in order to justify our algorithm. Suppose j^* is the true class for an example x. Intuitively, our approach will be justified if the expected center distributions $\overrightarrow{\mathbf{Q}}_{(j,k)}$ for incorrect classes are far enough from the empirical distribution $\widehat{\mathbf{P}}_{(x,\mathcal{S},k)}$. Specifically, suppose

$$D_{\mathrm{KL}}(\widehat{\mathbf{P}}_{(x,\mathcal{S},k)}||\overrightarrow{\mathbf{P}}_{(x,k)}) \leq D_{\mathrm{KL}}(\widehat{\mathbf{P}}_{(x,\mathcal{S},k)}||\overrightarrow{\mathbf{Q}}_{(j,k)})+\delta_k \ \forall j \neq j^*$$

We can then justify our approach using information geometry. For any $\delta_k > 0$, we have

$$D_{\mathrm{KL}}(\widehat{\mathbf{P}}_{(x,\mathcal{S},k)}||\overrightarrow{\mathbf{P}}_{(x,k)}) \leq D_{\mathrm{KL}}(\widehat{\mathbf{P}}_{(x,\mathcal{S},k)}||\widehat{\mathbf{Q}}_{(j^*,\mathcal{S},k)}) + \delta_k$$

Hence, it follows that

$$D_{\mathrm{KL}}(\widehat{\mathbf{P}}_{(x,\mathcal{S},k)}||\widehat{\mathbf{Q}}_{(j^*,\mathcal{S},k)}) \leq D_{\mathrm{KL}}(\overrightarrow{\mathbf{P}}_{(x,k)}||\widehat{\mathbf{Q}}_{(j,\mathcal{S},k)}) \ \forall j \neq j^*$$

It is worth noting that our algorithm will reduce to the majority rule when the prototypical distribution $\widehat{\mathbf{Q}}_{(j,\mathcal{S},k)}$ is defined as

$$\widehat{\mathbf{Q}}_{(j,\mathcal{S},k)}(i) = \begin{cases} 1-\epsilon & \text{if } i = j \\ \epsilon & \text{otherwise} \end{cases}$$

where ϵ is a small constant. In this case, the non-majority examples do not contribution any information to the final prediction. Hence, the prediction will be made based solely on the majority.

4. 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 1.

Table 1. 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

4.1. 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, or $x \in [0, b-1]^d$. 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 $\delta_k > 0$.

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.

4.2. 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 wraping (DTW) distance (Bahlmann & Burkhardt, 2004).

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.

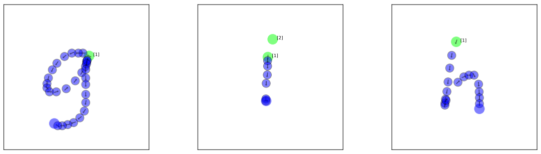


Figure 3. Some examples from the uRight handwriting dataset.

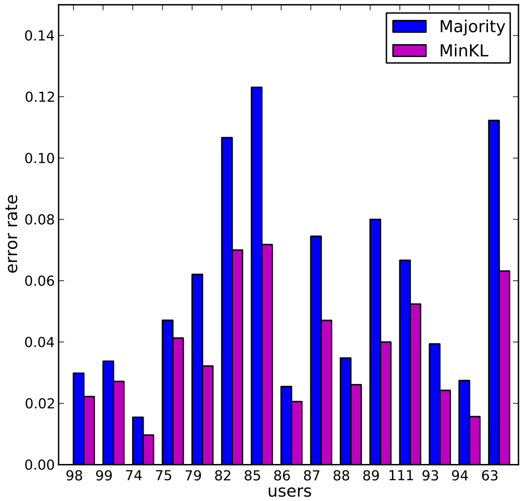
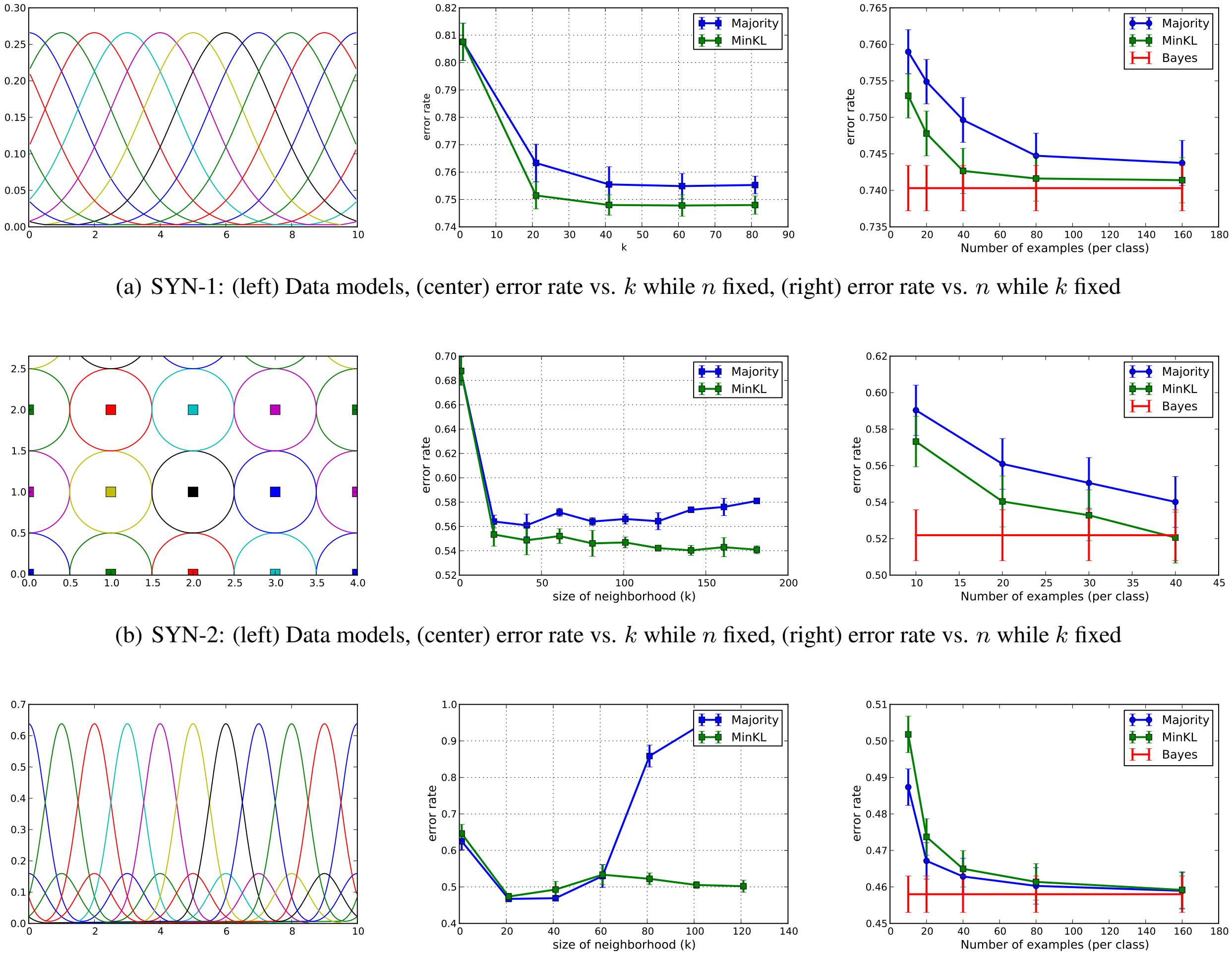


Figure 4. The average error rates of MinKL and Majority for each user.



(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

Figure 2. Results from the synthetic data experiment.

4.3. MNIST

The MNIST dataset (Lecun et al., 1998) 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 prepropressed 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 (Lecun et al., 1998). 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 7 shows the test error rates of both MinKL and Majority obtained using different k.

4.4. SVHN

The SVHN dataset (Netzer et al., 2011) 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 (Dalal & Triggs, 2005) using the block size of 4x4 with 8 orientations per block. The Manhattan distance is used as the similarity measure in the neighborhood calculation.

In (Netzer et al., 2011), the test error rate for HOG features combined with an SVM is reported to be around 15%.

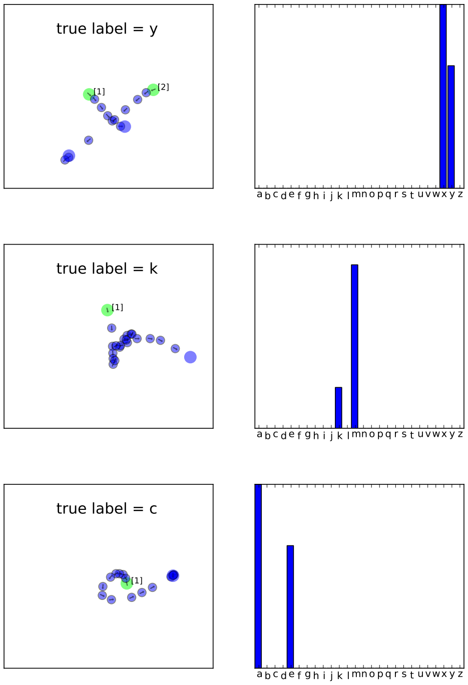
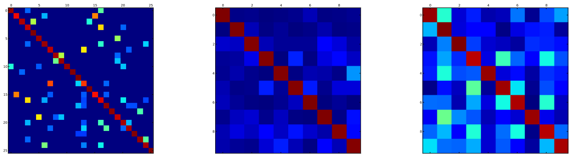


Figure 5. Some examples misclassified by Majority but correctly classified by MinKL. Each handwriting trajectory is shown on the left and the corresponding empirical distribution induced by its 5 neighbors is shown on the right.

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 7 shows the test error rates of both MinKL and Majority obtained using different k.

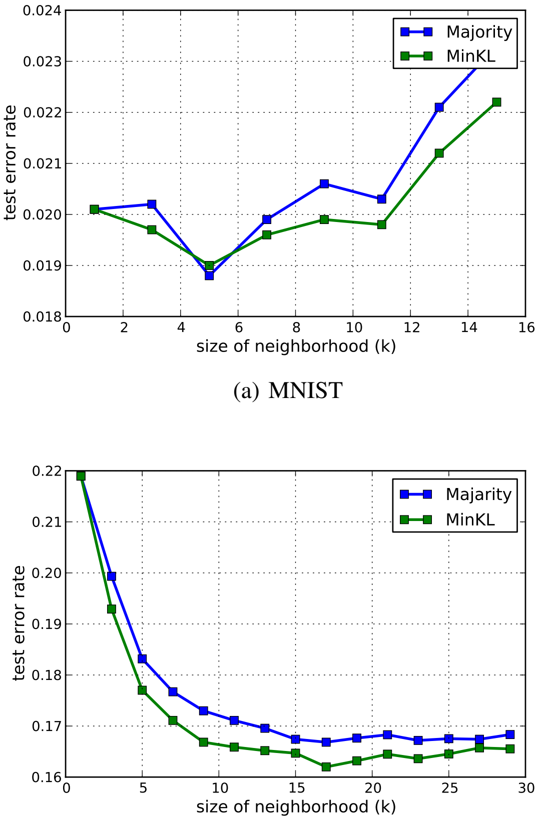
5. Discussion

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 toy example 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.



(a) uRight (b) MNIST (c) SVHN

Figure 6. A visualization of the empirical center distributions. Each row in each matrix corresponds to each class.



(a) MNIST

(b) SVHN

Figure 7. MNIST and SVHN results

From our analysis in Section 3, we know that our approach relies on an assumption that δ_k is relatively small. In **SYN-3**, we deliberately designed the dataset such that its δ_k is large.

In a sense, our approach naturally incoporates the infomation from the label space into the classification. The label space information is encoded in the form of the center distribution $\widehat{\mathbf{Q}}_{(j,\mathcal{S},k)}$. The performance of our approach depends on how well the center distribution $\widehat{\mathbf{Q}}_{(j,\mathcal{S},k)}$ can model the examples of the class.

neighborhood distributions of different classes are.

If the prototypical distributions are similar to each other, then our approach will not perform well. A workaround is to maintain multiple prototypical distributions per class.

As n increases, the performance gap between the two rules decreases and the error rate of both rules converge to the Bayes error.

We suspect that our approach will be more beneficial to problems with a large number of classes and the confusions between classes are non-uniform.

Our approach does not have a consistency gaurantee. It is possible that our approach will be sub-optimal when the number of training examples goes to infinity because the prototypical distribution model is incorrect or insuffcient. We do not worry about this problem that much since we see our approach being used in a small sample scenario.

In practice, other divergences might work better than the KL-divergence. The KL-divergence is considered a special case of a more general divergence function called Alpha-divergence (Cichocki & Amari, 2010), which is given by

$$D_A^{(\alpha)}(p||q) = \frac{1}{\alpha(\alpha-1)}\left(\sum_1^n p_i^\alpha q_i^{1-\alpha} - 1\right),\ \alpha \in \mathrm{R}\,\{0,1\}$$

The KL-divergence can be expressed as $D_{KL}(p||q) = \lim_{\alpha\to 1} D_A^{(\alpha)}(p||q)$. For the uRight dataset, we were able to obtain even lower error rate by using Alpha-divergence with $\alpha = 2$.

Our approach can be applied to other classification algorithms as well. The k-NN algorithm is very computational expensive in classifying a 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 significanly reduces the classication 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 estimating by

$$\mathbf{P}_x(j) = \frac{e^{d(x,C(j))}}{\sum_i e^{d(x,C(i))}}$$

We can apply MinKL to the class posterior computed this way.

6. Conclusions

We suggest a simple modification to the k-NN algorithm.

References

Bahlmann, C and Burkhardt, H. The writer independent online handwriting recognition system frog on hand and cluster generative statistical dynamic time warping, 2004. ISSN 01628828.

Bengio, S., Weston, J., and Grangier, D. Label embedding trees for large multi-class tasks. *Advances in Neural Information Processing Systems*, 23(1):1–10, 2010.

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

Cichocki, A. and Amari, S. Families of Alpha- Beta- and Gamma- Divergences: Flexible and Robust Measures of Similarities. *Entropy*, 12(6):1532–1568, June 2010. ISSN 1099-4300. doi: 10.3390/e12061532.

Collins, M. and Singh-Miller, N. 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.

Cover, T. M. and Hart, P. E. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967. ISSN 0018-9448. doi: 10.1109/TIT.1967.1053964.

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

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

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 1998. ISSN 0018-9219. doi: 10.1109/5.726791.

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