The Nearest Feature Midpoint - A Novel Approach for Pattern Classification

Zonglin Zhou[1]*and Chee Keong Kwoh[2]
[1]Department of Computer Science
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
zlzhou@cs.ust.hk
[2]Bioinformatics Research Centre
School of Computer Engineering, Nanyang Technological University
Nanyang Avenue, Singapore 639798
asckkwoh@ntu.edu.sg

Abstract

In this paper, we propose a method, called the nearest feature midpoint (NFM), for pattern classification. Any pair of feature points of the same class is generalized by the feature midpoint (FM) between them. Hence the representational capacity of available prototypes can be expanded. The classification is determined by the nearest distance from the query feature point to each FM. This paper compares the NFM classifier against the nearest feature line (NFL) classifier, which has reported successes in various applications. In the NFL, any pair of feature points of the same class is generalized by the feature line (FL) passing through them, and the classification is evaluated on the nearest distance from the query feature point to each FL. The NFM can be considered to be the refinement of the NFL.

A theoretical proof is provided in this paper to show that for the n-dimensional Gaussian distribution, the classification based on the NFM distance metric will achieve the least error probability as compared to those based on any other points on the feature lines. Furthermore, a theoretical investigation is provided that under certain assumption the NFL is approximately equivalent to the NFM when the dimension of the feature space is high. The experimental evaluations on both simulated and real-life benchmark data concur with all the theoretical investigations, as well as indicate that the NFM is effective for the classification of the data with a Gaussian distribution or with a distribution that can be reasonably approximated by a Gaussian.

Keywords: pattern classification, nearest feature midpoint (NFM), nearest feature line (NFL), nearest neighbor (NN) classification, $k-$nearest neighbor ($k-$NN) classification.

*Corresponding author

1 Introduction

In the context of pattern recognition, the performance of a classification approach relies critically on the distance metric employed over the input feature space. various distance metrics have been used for pattern classification: Euclidean distance, Cosine distance, Hamming distance, and so on, as well as their variations in locally adaptive fashion [10, 14, 6]. However, they all have distinction between the query and an individual prototype (feature point). In classification, a class is considered as a collection of isolated points in the feature space, and there is no class membership concept for the prototypes. This type of classification can be referred collectively as the nearest-neighbor (NN) classification [4, 3, 8, 13, 12, 14]. However, in many cases, multiple prototypes are available within a class. Such a characteristic can be utilized to improve the classification performance but has been ignored by the NN type of methods [18].

1.1 Related Work

In [19, 17, 18], the method of the nearest feature line (NFL) is proposed for pattern classification to circumvent the above mentioned limitations of the NN. The basic assumption made in the NFL is that at least two prototype feature points are available for each class, which is usually satisfied. In the NFL, a feature subspace is constructed for each class from straight lines (*feature lines*) passing through each pair of the prototypes (*feature points*) belonging to that class. The prototypes are generalized by the feature lines. A feature line (FL) covers more space than the two feature points alone and virtually provides an infinite number of feature points of the class that the two prototypes belong to. The representational capacity of available prototypes is thus expanded. A FL provides information about linear variants of the two prototypes. The NFL distance metric is defined as the minimum Euclidean distance between the query and the feature lines. The rationale of the NFL can be justified intuitively as follows [18]: An image or sound, for example, corresponds to a point (vector) in a feature space. When one prototype image or sound changes continuously to another prototype in some way, it draws a trajectory linking the corresponding feature points in the feature space. All such trajectories in the same class constitute a subspace representing that class. A similar image or sound should be close to the subspace though may not be so to the original prototypes. In the NN, such dynamic information is not represented.

1.2 Our Work

In this paper, we present a refined method of the NFL, called the nearest feature midpoint (NFM), for pattern classification. We followed the same methodology of the NFL and will compare the performance of the NFM against the NFL. The basic assumption made in the NFM is same as in the NFL, that is at least two prototype feature points are available for each class. In the NFM, each feature subspace is constructed for each class from respective midpoints (*feature midpoints*) between each pair of the prototypes belonging to that class. In addition, the NFM also makes use of the available information about classes contained in the multiple prototypes of each class. The within-class prototypes are generalized by the feature midpoints to represent variants of that class, and the generalized ability of the classifier is thus improved. The NFM distance metric is defined as the minimum Euclidean distance between the query and the feature midpoints. In this paper, we first provide the theoretical proof that for $n-$dimensional Gaussian distribution and under some reasonable assumptions, the classification based on the NFM distance metric will achieve the least error probability relative to those based on any other points on the

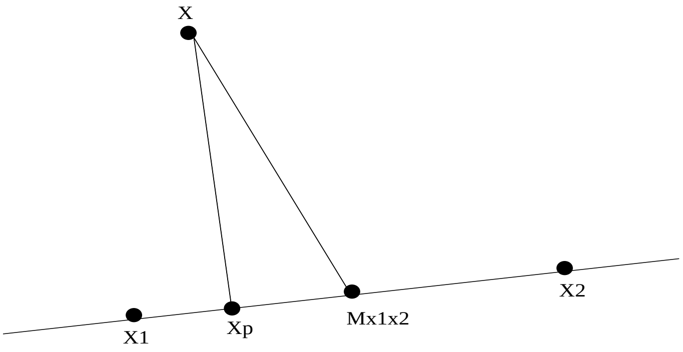

Figure 1. Generalizing two prototype feature points x_1 and x_2 by the feature line $\overline{\mathrm{x}_1\mathrm{x}_2}$, and the feature midpoint $\mathrm{m}_{\mathrm{x}_1\mathrm{x}_2}$, respectively. The feature point x of a query is projected onto the line as point x_p.

feature lines. Furthermore, we will prove that, under the assumption that the components of the query and two prototypes are independent and identically distributed (*i.i.d.* for short), the projection point of the query on the feature line passing through the two prototypes will converge in probability to the feature midpoint of the two prototypes when the dimension of the feature space is high. The NFL is thus approximately equivalent to the NFM in the case. But it will be pointed out that the computational complexity of the NFM is significantly less than the NFL. In the experiment section, we will show from empirical evidences that all theoretical claims developed in this paper are demonstrated.

1.3 Organization of Paper

The rest of this paper is organized as follows. In the next section we give a brief review of the NFL classifier and formally define the NFM classification. The detailed theoretical analysis of the NFM is given in section 3. Section 4 reports empirical results on both simulated and real-life benchmark data. Conclusions and remarks about future directions are provided in the final section.

2 Pattern Classification Using NFL and NFM

In the NFL, the straight line passing through x_1 and x_2 of the same class, denoted $\overline{\mathrm{x}_1\mathrm{x}_2}$, is called a *feature line* (FL) of that class. The feature point x of a query (test) sample is projected onto an FL as point x_p (Fig. 1). The FL distance between x and $\overline{\mathrm{x}_1\mathrm{x}_2}$ is defined as

$$d(\mathrm{x}, \overline{\mathrm{x}_1\mathrm{x}_2}) = ||\mathrm{x} - \mathrm{x}_p||, \tag{1}$$

where $|| \cdot ||$ is some norm.

The projection point can be computed as $\mathrm{x}_p = \mathrm{x}_1 + \mu(\mathrm{x}_2 - \mathrm{x}_1)$, where $\mu \in \mathcal{R}$, called the position parameter, can be calculated from x, x_1, and x_2 as follows:

$$\mu = \frac{(\mathrm{x} - \mathrm{x}_1) \cdot (\mathrm{x}_2 - \mathrm{x}_1)}{(\mathrm{x}_2 - \mathrm{x}_1) \cdot (\mathrm{x}_2 - \mathrm{x}_1)}, \tag{2}$$

where "$\cdot$" stands for dot product. The parameter μ describes the position of x_p relative to x_1 and x_2. Assuming that there are N_c, > 1, prototype feature points available for class c, a number of $K_c =$

$\frac{N_c(N_c-1)}{2}$ lines can be constructed to represent the class. The total number of feature lines for a number of M classes is $N_{total} = \sum_{c=1}^{M} K_c$. The NFL classification is done by using the minimum distance between the feature point of the query and the N_{total} feature lines.

In the NFM proposed here, the midpoint between $\mathbf{x}_1$ and $\mathbf{x}_2$ of the same class is called a *feature midpoint* (FM) of that class, and is denoted $\mathbf{m}_{\mathbf{x}_1\mathbf{x}_2}$. Any point on the feature line $\overline{\mathbf{x}_1\mathbf{x}_2}$ can be expressed as $\mathbf{x}_1 + \lambda(\mathbf{x}_2 - \mathbf{x}_1)$, where $-\infty < \lambda < \infty$. When $\lambda = \frac{1}{2}$, $\mathbf{m}_{\mathbf{x}_1\mathbf{x}_2} = \frac{1}{2}(\mathbf{x}_1 + \mathbf{x}_2)$ is the FM. The FM distance between the feature point $\mathbf{x}$ of a query and $\mathbf{m}_{\mathbf{x}_1\mathbf{x}_2}$ is defined as

$$d(\mathbf{x}, \mathbf{m}_{\mathbf{x}_1\mathbf{x}_2}) = ||\mathbf{x} - \mathbf{m}_{\mathbf{x}_1\mathbf{x}_2}||, \tag{3}$$

where $||\cdot||$ is the same norm as in Eq. (1).

If there are $N_c, > 1$, prototype feature points available for class c, a number of $K_c = \frac{N_c(N_c-1)}{2}$ feature midpoints can then be constructed to represent the class. The total number of feature midpoints for a number of M classes is $N_{total} = \sum_{c=1}^{M} K_c$, which amounts to the same number of feature lines of the M classes.

The NFM classification is done by evaluating the minimum distance between the feature point of the query and the N_{total} feature midpoints. Mathematically, let $\mathbf{x}_i^c$ and $\mathbf{x}_j^c$ be two distinct prototype feature points belonging to class c. The FM distance between $\mathbf{x}$ of the query and each pair of prototypes $\mathbf{x}_i^c$ and $\mathbf{x}_j^c$, $i \neq j$, is calculated for each class c. This yields a number of N_{total} distances. The distances are sorted in ascending order, each being associated with a class identifier, and two prototypes. The *NFM distance* is the first rank FM distance:

$$d(\mathbf{x}, \mathbf{m}_{\mathbf{x}_{i^*}^{c^*}\mathbf{x}_{j^*}^{c^*}}) = \min_{1 \le c \le M} \min_{1 \le i < j \le N_c} d(\mathbf{x}, \mathbf{m}_{\mathbf{x}_i^c\mathbf{x}_j^c}). \tag{4}$$

The first rank gives the NFM classification of the best matched class c^* and the two best matched prototypes i^* and j^* of the class.

3 Theoretical Analysis

In this section, we will investigate in theory the NFM method as well as the relationship between the NFL and NFM. The comparison of computational complexities of the NFL and NFM will be made as well.

3.1 The Theoretical Justification of the NFM Method

Denote the mean vector of class ℓ by $\overline{\mathbf{x}}_\ell$ and covariance matrix by Σ_ℓ. Let $\mathbf{x}_1$ and $\mathbf{x}_2$ be any two prototypes of class ℓ. $E(\mathbf{x}_1) = E(\mathbf{x}_2) = \overline{\mathbf{x}}_\ell$, $Cov(\mathbf{x}_1) = Cov(\mathbf{x}_2) = \Sigma_\ell$. Assume that all the points in each class are independent of each other, then

$$E(\mathbf{x}_1 + \lambda(\mathbf{x}_2 - \mathbf{x}_1)) = \overline{\mathbf{x}}_\ell, \qquad Cov(\mathbf{x}_1 + \lambda(\mathbf{x}_2 - \mathbf{x}_1)) = ((1-\lambda)^2 + \lambda^2)\Sigma_\ell, \tag{5}$$

$$\min_{-\infty < \lambda < \infty} ((1-\lambda)^2 + \lambda^2) \;=\; ((1-\lambda)^2 + \lambda^2)|_{\lambda = \frac{1}{2}} = \frac{1}{2}. \tag{6}$$

The following lemma elucidates the assumptions under which the nearest-neighbor (NN) classifier is equivalent to *Bayes* classifier.

Lemma 3.1 *Suppose there are L classes $\omega_1, \ldots, \omega_L$. The likelihood distribution is*

$$p(\mathbf{x} \mid \omega_\ell) = \frac{1}{Z_\ell} \exp\{-\frac{1}{2} d(\mathbf{x} \mid \omega_\ell)\}, \tag{7}$$

where

$$d(\mathbf{x} \mid \omega_\ell) = (\mathbf{x} - \overline{\mathbf{x}}_\ell)^T \Sigma_\ell^{-1} (\mathbf{x} - \overline{\mathbf{x}}_\ell), \tag{8}$$

is the distance between $\mathbf{x}$ and $\overline{\mathbf{x}}_\ell$ given Σ_ℓ, and $Z_\ell = (2\pi)^{n/2} |\Sigma_\ell|^{1/2}, \quad \ell \in \{1, ..., L\}$, n is the dimension of $\mathbf{x}$.

The NN classifier is equivalent to Bayes classifier, if the following assumptions hold:

(i) All classes are equally probable, $P_\ell = P(\omega_\ell) = \frac{1}{L}$,

(ii) $\frac{\Delta \exp\{-\frac{1}{2} d(\mathbf{x} \mid \omega_\ell)\}}{\Delta d(\mathbf{x} \mid \omega_\ell)} \gg \frac{\Delta Z_\ell}{\Delta \Sigma_\ell}$ for all $\ell \in \{1, ..., L\}$, especially, $\Sigma_1 = \Sigma_2 = ... = \Sigma_L$.

Proof: The posterior distribution is

$$p(\omega_\ell \mid \mathbf{x}) = \frac{P_\ell \, p(\mathbf{x} \mid \omega_\ell)}{p(\mathbf{x})}, \tag{9}$$

where $p(\mathbf{x}) = \sum_{\ell=1}^{L} P_\ell p(\mathbf{x} \mid \omega_\ell)$. In terms of Bayes classification rule, $\mathbf{x}$ is classified as ω_i if $p(\omega_i \mid \mathbf{x}) = \max_{1 \le \ell \le L} p(\omega_\ell \mid \mathbf{x})$, which is equivalent to $p(\mathbf{x} \mid \omega_i) = \max_{1 \le \ell \le L} p(\mathbf{x} \mid \omega_\ell)$ in terms of Eq. (9) and assumption (i). According to assumption (ii), the change of $d(\mathbf{x} \mid \omega_\ell)$ relatively dominates that of $p(\mathbf{x} \mid \omega_\ell)$ from the change of Z_ℓ, i.e., from the change of Σ_ℓ. Thereby, Z_ℓ in Eq. (7) assumes to be almost unchanged for all $\ell \in \{1, ..., L\}$. As such, $p(\mathbf{x} \mid \omega_i) = \max_{1 \le \ell \le L} p(\mathbf{x} \mid \omega_\ell)$ if and only if $d(\mathbf{x} \mid \omega_i) = \min_{1 \le \ell \le L} d(\mathbf{x} \mid \omega_\ell)$, which is just the NN classification rule. Thus, the NN classifier is equivalent to the Bayes classifier under the assumptions. ■

The assumption (ii) of lemma 3.1 is especially true in many real cases where the prototypes from different classes are subjected to the same noise processes, hence we can set $\Sigma_1 = \Sigma_2 = ... = \Sigma_L$ [5].

Since the approximate expression of the error probability of Bayes classifier is available in [11], derivation of lemma 3.1 will have the same result for the NN.

To illustrate further, the error probability of Bayes classifier and hence that of the NN is examined in the two-class case for simplicity, where the two classes are equally possible, $P_1 = P_2 = P$, i.e., no priors are available. Under the assumptions of lemma 3.1, the error probability of Bayes classifier [11], thus that of the NN, is

$$\begin{aligned} \varepsilon &= P_1 \varepsilon_1 + P_2 \varepsilon_2 && (10) \\ &= P \int_0^\infty p_h(h|\omega_1) dh + P \int_{-\infty}^0 p_h(h|\omega_2) dh, && (11) \end{aligned}$$

where

$$\begin{aligned} h(\mathbf{x}) &= -\ln p(\mathbf{x} \mid \omega_1) + \ln p(\mathbf{x} \mid \omega_2) && (12) \\ &= \frac{1}{2}(\mathbf{x} - \overline{\mathbf{x}}_1)^T \Sigma_1^{-1} (\mathbf{x} - \overline{\mathbf{x}}_1) - \frac{1}{2}(\mathbf{x} - \overline{\mathbf{x}}_2)^T \Sigma_2^{-1} (\mathbf{x} - \overline{\mathbf{x}}_2) + \frac{1}{2} \ln \frac{|\Sigma_1|}{|\Sigma_2|}, && (13) \end{aligned}$$

is called the discriminant function. ε_1 and ε_2 can be calculated as follows.

When $\Sigma_1 = \Sigma_2 = \Sigma$, $h(\mathbf{x}) = (\overline{\mathbf{x}}_2 - \overline{\mathbf{x}}_1)^T \Sigma^{-1}\mathbf{x} + \frac{1}{2}(\overline{\mathbf{x}}_1^T \Sigma^{-1}\overline{\mathbf{x}}_1 - \overline{\mathbf{x}}_2^T \Sigma^{-1}\overline{\mathbf{x}}_2)$. In this case, $h(\mathbf{x})$ is also a Gaussian random variable. The mean of $h(\mathbf{x})$ can be calculated as:

$$E\{h(\mathbf{x})|\omega_\ell\} = (\overline{\mathbf{x}}_2 - \overline{\mathbf{x}}_1)^T \Sigma^{-1} E\{\mathbf{x}|\omega_\ell\} + \frac{1}{2}(\overline{\mathbf{x}}_1^T \Sigma^{-1}\overline{\mathbf{x}}_1 - \overline{\mathbf{x}}_2^T \Sigma^{-1}\overline{\mathbf{x}}_2) \tag{14}$$

where $E\{\mathbf{x}|\omega_\ell\} = \overline{\mathbf{x}}_\ell$, $\ell = 1, 2$.

Letting, $\eta = \frac{1}{2}(\overline{\mathbf{x}}_2 - \overline{\mathbf{x}}_1)^T \Sigma^{-1}(\overline{\mathbf{x}}_2 - \overline{\mathbf{x}}_1)$, we have $E\{h(\mathbf{x})|\omega_1\} = -\eta$, and $E\{h(\mathbf{x})|\omega_2\} = +\eta$. The variance of $h(\mathbf{x})$ is

$$Var\{h(\mathbf{x})|\omega_\ell\} = E\{[h(\mathbf{x}) - E\{h(\mathbf{x})|\omega_\ell\}]^2|\omega_\ell\} = (\overline{\mathbf{x}}_2 - \overline{\mathbf{x}}_1)^T \Sigma^{-1}(\overline{\mathbf{x}}_2 - \overline{\mathbf{x}}_1) = 2\eta. \tag{15}$$

Now,

$$\begin{aligned} \varepsilon_1 &= \int_0^\infty p_h(h|\omega_1)dh = \int_{\eta/\sigma}^\infty \frac{1}{\sqrt{2\pi}} e^{-\zeta^2/2} d\zeta = 1 - \Phi(\frac{\eta}{\sigma}), & (16) \\ \varepsilon_2 &= \int_{-\infty}^0 p_h(h|\omega_2)dh = \int_{-\infty}^{-\eta/\sigma} \frac{1}{\sqrt{2\pi}} e^{-\zeta^2/2} d\zeta = \Phi(-\frac{\eta}{\sigma}), & (17) \end{aligned}$$

where $\Phi(\xi) = \int_{-\infty}^{\xi} \frac{1}{\sqrt{2\pi}} e^{-\zeta^2/2} d\zeta$ is the normal error function, and $\sigma^2 = Var\{h(\mathbf{x})|\omega_1\} = Var\{h(\mathbf{x})|\omega_2\} = 2\eta$.

Therefore,

$$\varepsilon = P(1 - \Phi(\frac{\eta}{\sigma})) + P\Phi(-\frac{\eta}{\sigma}). \tag{18}$$

Derivation of Eq. (18) is needed for the proof of the theorem below. The following theorem will theoretically justify the correctness of the NFM method. In the theorem below, the classification based on an arbitrary point $\mathbf{x}_\lambda$ on feature lines means that the position parameter λ is fixed for all the feature lines. The classification is done using the nearest distance from the query to each $\mathbf{x}_\lambda$.

Theorem 3.1 *For the n-dimensional Gaussian distribution in Eq. (7), if the two assumptions in lemma 3.1 are satisfied and $\Sigma_1 = \Sigma_2 = \Sigma$, then the classification based on the nearest feature midpoint will achieve the least error probability as compared to those based on any other point $\mathbf{x}_\lambda$, $\lambda \neq \frac{1}{2}$, on the feature lines.*

Proof: Assume that $\mathbf{x}_1, \mathbf{x}_2$ are two prototypes selected arbitrarily from same class ℓ. Let $\mathbf{x}_\lambda$ be any point on the feature line $\overline{\mathbf{x}_1\mathbf{x}_2}$, $\mathbf{x}_\lambda = \mathbf{x}_1 + \lambda(\mathbf{x}_2 - \mathbf{x}_1)$, $-\infty < \lambda < \infty$. From Eq. (5), $\mathbf{x}_\lambda$ follows a Gaussian distribution $N(\overline{\mathbf{x}}_\ell, ((1-\lambda)^2 + \lambda^2)\Sigma_\ell)$. In addition, from Eq. (5) and Eq. (14),

$$\begin{aligned} E\{h(\mathbf{x}_\lambda)|\omega_1\} &= -\eta, & (19) \\ E\{h(\mathbf{x}_\lambda)|\omega_2\} &= \eta, & (20) \\ Var\{h(\mathbf{x}_\lambda)|\omega_\ell\} &= E[\{h(\mathbf{x}_\lambda) - E\{h(\mathbf{x}_\lambda)|\omega_\ell\}\}^2|\omega_\ell] & \\ &= (\overline{\mathbf{x}}_2 - \overline{\mathbf{x}}_1)^T \Sigma^{-1} D\{\mathbf{x}_\lambda|\omega_\ell\}\Sigma^{-1}(\overline{\mathbf{x}}_2 - \overline{\mathbf{x}}_1) & \\ &= ((1-\lambda)^2 + \lambda^2)\sigma^2, \quad \ell = 1, 2. & (21) \end{aligned}$$

From Eq. (18), the error probability of the classification based on the $\mathbf{x}_\lambda$ will be

$$\begin{aligned}\varepsilon^{(\lambda)} &= P(1-\Phi(-E\{h(\mathbf{x}_\lambda)|\omega_1\}/Var\{h(\mathbf{x}_\lambda)|\omega_1\})) + P\Phi(-E\{h(\mathbf{x}_\lambda)|\omega_2\}/Var\{h(\mathbf{x}_\lambda)|\omega_2\}) \\ &= P(1-\Phi(\frac{\eta}{(\sqrt{(1-\lambda)^2+\lambda^2})\sigma})) + P\Phi(-\frac{\eta}{(\sqrt{(1-\lambda)^2+\lambda^2})\sigma}). \qquad (22)\end{aligned}$$

Thus, from Eq. (6),

$$\begin{aligned}\min_{\mathbf{x}_\lambda \in \overline{\mathbf{x}_1\mathbf{x}_2}}\{\varepsilon^{(\lambda)}\} &= (P(1-\Phi(\frac{\eta}{(\sqrt{(1-\lambda)^2+\lambda^2})\sigma})) + P\Phi(-\frac{\eta}{(\sqrt{(1-\lambda)^2+\lambda^2})\sigma}))|_{\lambda=\frac{1}{2}} \\ &= P(1-\Phi(\frac{\sqrt{2}\eta}{\sigma})) + P\Phi(-\frac{\sqrt{2}\eta}{\sigma}). \qquad (23)\end{aligned}$$

■

In the case $\Sigma_1 \neq \Sigma_2$, no optimal solution, in theory, is available since Σ_1, Σ_2 are class-dependent and the discriminant function $h(\mathbf{x})$ in Eq. (13) is quadratic. The common practice is to convert this case into the class-independent case, i.e., to approximate Σ, by letting $\Sigma_1 = \Sigma_2 = \Sigma$, $\Sigma = P_1\Sigma_1 + P_2\Sigma_2 = \frac{1}{2}\Sigma_1 + \frac{1}{2}\Sigma_2$. That is to obtain the suboptimal solution [7, 21]. This is so called the *shrinkage* technique. In many real-life problems, it's often assumed that the prototypes from different classes are subjected to the same noise processes, hence we can assume $\Sigma_1 = \Sigma_2$ [5]. The empirical results on both simulated and real-life benchmark data in next section will demonstrate the finding in theorem 3.1.

3.2 The Gaussian Approximation

If the likelihood distribution $p(\mathbf{x} \mid \omega_\ell)$ in Eq. (7) for some class ω_ℓ is not Gaussian, it can often be approximated as a Gaussian distribution based on the central limit theorem. As the number of samples in the class ω_ℓ increases, this Gaussian approximation is expected to become increasingly accurate. Furthermore, it is common to use gradient-based methods to find the maximum of $\ln p(\mathbf{x} \mid \omega_\ell)$, which defines the most probable value $\mathbf{x}_{MP(\ell)}$ for the variable $\mathbf{x}$. Using a second degree Taylor polynomial of $\ln p(\mathbf{x} \mid \omega_\ell)$ about $\mathbf{x}_{MP(\ell)}$ to approximate $\ln p(\mathbf{x} \mid \omega_\ell)$, we obtain

$$\ln p(\mathbf{x} \mid \omega_\ell) \approx \ln p(\mathbf{x}_{MP(\ell)} \mid \omega_\ell) + -\frac{1}{2}(\mathbf{x} - \mathbf{x}_{MP(\ell)})^T \mathbf{A}(\mathbf{x} - \mathbf{x}_{MP(\ell)}), \qquad (24)$$

where $\mathbf{A} = -\nabla^2 \ln p(\mathbf{x} \mid \omega_\ell)|_{\mathbf{x}=\mathbf{x}_{MP(\ell)}}$, the negative Hessian of $\ln p(\mathbf{x} \mid \omega_\ell)$ evaluated at $\mathbf{x}_{MP(\ell)}$. Exponentiating both sides in the above equation, we obtain

$$p(\mathbf{x} \mid \omega_\ell) \approx p(\mathbf{x}_{MP(\ell)} \mid \omega_\ell) \exp\{-\frac{1}{2}(\mathbf{x} - \mathbf{x}_{MP(\ell)})^T \mathbf{A}(\mathbf{x} - \mathbf{x}_{MP(\ell)})\}. \qquad (25)$$

On the accuracy of Gaussian approximation, in general it provides adequate approximation for the problems in which the likelihood distributions of modest dimensionality are not grossly non-Gaussian. It is hard to be specific, roughly the sample size of one class should be not less than $5d$, with d being the dimension of parameters of the likelihood distribution of the class. Those data with sample size greater than $20d$ are large enough for Gaussian approximation to work well in most cases, provided that a reasonably good parameterization is used [15]. For more general statistical models with multiple maxima in their likelihood distributions, we can still expect the likelihood distributions to be dominated by locally Gaussian peaks in terms of the central limit theorem.

3.3 The comparison of the NFM against the NFL

In the NFM, the midpoint position of each pair of prototypes in each class is fixed. Likewise in theorem 3.1, the position parameter λ in the classification based on an arbitrary point $\mathbf{x}_\lambda$ is fixed for all the feature lines. In the NFL, the parameter μ in Eq. (2) describing the position of the projection point $\mathbf{x}_p$ varies for different feature lines. Consequently, theorem 3.1 can not ensure that the error probability of the NFM is less than the NFL. The theorem below addresses the relationship between the NFL and NFM.

Theorem 3.2 *Let* $\mathbf{x} = (\mathbf{x}^{(1)}, ..., \mathbf{x}^{(n)})^T$, $\mathbf{x}_1 = (\mathbf{x}_1^{(1)}, ..., \mathbf{x}_1^{(n)})^T$, $\mathbf{x}_2 = (\mathbf{x}_2^{(1)}, ..., \mathbf{x}_2^{(n)})^T$ *be n-dimensional random vectors, and* $\mathbf{x}_1$, $\mathbf{x}_2$ *belong to the same class and have a common distribution,*

$$\mathbf{x}_p = \mathbf{x}_1 + \frac{(\mathbf{x} - \mathbf{x}_1) \cdot (\mathbf{x}_2 - \mathbf{x}_1)}{(\mathbf{x}_2 - \mathbf{x}_1) \cdot (\mathbf{x}_2 - \mathbf{x}_1)} (\mathbf{x}_2 - \mathbf{x}_1), \tag{26}$$

where "$\cdot$" stands for dot product. If the components of $\mathbf{x}$, $\mathbf{x}_1$ *and* $\mathbf{x}_2$ *are i.i.d., then* $\mathbf{x}_p \xrightarrow{P} \frac{1}{2}(\mathbf{x}_1 + \mathbf{x}_2)$ *when* $n \to \infty$. *It means that* $\mathbf{x}_p$ *converges, in probability sense, to* $\frac{1}{2}(\mathbf{x}_1 + \mathbf{x}_2)$ *when n goes to infinity.*

Proof: From Eq. (26), we have, for $\forall m = 1, ..., n$,

$$\begin{aligned} \mathbf{x}_p^{(m)} &= \mathbf{x}_1^{(m)} + \frac{(\mathbf{x} - \mathbf{x}_1) \cdot (\mathbf{x}_2 - \mathbf{x}_1)}{(\mathbf{x}_2 - \mathbf{x}_1) \cdot (\mathbf{x}_2 - \mathbf{x}_1)} \\ &\quad \cdot (\mathbf{x}_2^{(m)} - \mathbf{x}_1^{(m)}), \end{aligned} \tag{27}$$

therefore,

$$\begin{aligned} \mathbf{x}_p^{(m)} &= \mathbf{x}_1^{(m)} + \frac{\sum_{k=1}^n (\mathbf{x}^{(k)} - \mathbf{x}_1^{(k)})(\mathbf{x}_2^{(k)} - \mathbf{x}_1^{(k)})}{\sum_{k=1}^n (\mathbf{x}_2^{(k)} - \mathbf{x}_1^{(k)})(\mathbf{x}_2^{(k)} - \mathbf{x}_1^{(k)})} \\ &\quad \cdot (\mathbf{x}_2^{(m)} - \mathbf{x}_1^{(m)}). \end{aligned} \tag{28}$$

According to the assumption made in the theorem, $(\mathbf{x}^{(1)} - \mathbf{x}_1^{(1)})(\mathbf{x}_2^{(1)} - \mathbf{x}_1^{(1)}), ..., (\mathbf{x}^{(n)} - \mathbf{x}_1^{(n)})(\mathbf{x}_2^{(n)} - \mathbf{x}_1^{(n)})$ are i.i.d., and $(\mathbf{x}_2^{(1)} - \mathbf{x}_1^{(1)})(\mathbf{x}_2^{(1)} - \mathbf{x}_1^{(1)}), ..., (\mathbf{x}_2^{(n)} - \mathbf{x}_1^{(n)})(\mathbf{x}_2^{(n)} - \mathbf{x}_1^{(n)})$ are i.i.d. as well.
$\forall k, k = 0, ..., n$, suppose $Var(\mathbf{x}_1^{(k)}) = Var(\mathbf{x}_2^{(k)}) = \sigma_1^2$, then,

$$\begin{aligned} E[(\mathbf{x}^{(k)} - \mathbf{x}_1^{(k)})(\mathbf{x}_2^{(k)} - \mathbf{x}_1^{(k)})] &= \sigma_1^2, \\ E[(\mathbf{x}_2^{(k)} - \mathbf{x}_1^{(k)})(\mathbf{x}_2^{(k)} - \mathbf{x}_1^{(k)})] &= 2\sigma_1^2. \end{aligned}$$

In the law of large numbers, we have,

$$\begin{aligned} &\frac{1}{n}\sum_{k=1}^n (\mathbf{x}^{(k)} - \mathbf{x}_1^{(k)})(\mathbf{x}_2^{(k)} - \mathbf{x}_1^{(k)}) \xrightarrow{P} \sigma_1^2, \quad n \to \infty, \\ &\frac{1}{n}\sum_{k=1}^n (\mathbf{x}_2^{(k)} - \mathbf{x}_1^{(k)})(\mathbf{x}_2^{(k)} - \mathbf{x}_1^{(k)}) \xrightarrow{P} 2\sigma_1^2, \quad n \to \infty. \end{aligned}$$

From Eq. (28),

$$\mathbf{x}_p^{(m)} \xrightarrow{P} \mathbf{x}_1^{(m)} + \frac{1}{2}(\mathbf{x}_2^{(m)} - \mathbf{x}_1^{(m)}), \quad n \to \infty,$$

that is,

$$\mathbf{x}_p^{(m)} \xrightarrow{P} \frac{1}{2}(\mathbf{x}_1^{(m)} + \mathbf{x}_2^{(m)}), \quad n \to \infty,$$

equivalently,

$$\mathbf{x}_p \xrightarrow{P} \frac{1}{2}(\mathbf{x}_1 + \mathbf{x}_2), \quad n \to \infty. \quad \blacksquare$$

In theorem 3.2, $\mathbf{x}_1$ and $\mathbf{x}_2$ belong to the same class, so they should follow a common distribution. The assumption that the components of $\mathbf{x}$, $\mathbf{x}_1$ and $\mathbf{x}_2$ are i.i.d. seems too demanding in practice. It is only expected to hold approximately for many real life applications. When n is much larger than the number of all the prototype feature points available, however, the classification rules assuming that the feature dimensions are independent often perform better than the rules assuming dependent feature dimensions [1]. This phenomenon has been reported recently for texture classification [16], and for microarray data [9]. In this paper, derivation of the theoretical relationship between the NFM and the NFL in the above theorem is based on the *i.i.d.* assumption. The dimension of the feature spaces of many real life applications is high. Thus, the performance of the NFL classifier is approximately equivalent to the NFM in such cases according to theorem 3.2.

Complexity wise, since the position parameter μ in Eq. (2) for the NFL depends on the query and each pair of prototypes in each class, it needs to be calculated for each query and each pair of prototypes. In the NFM, however, the midpoint position of each pair of prototypes in each class is fixed, and there is no need to perform the computation of the position parameter. Thus, the computational complexity of the NFM is significantly less than the NFL. In view of this and theorem 3.2, the NFM is a good alternative to the NFL in high dimensional feature spaces. In fact, the NFM consistently outperforms the NFL in the following experiments on both simulated and real-life benchmark data and various applications tested.

4 Numerical Experiments

In this section we report the experimental examination of the NFM method in comparison with some state of the art classification methods such as NN, $k-$NN, and the NFL. The experiments are conducted on a simulated data set and 15 real-life benchmark data sets from the UCI machine learning repository [2]. They are all evaluated by using the *leave-one-out* test: when a sample is used as the query, it is not used as a prototype, i.e., it is removed from the prototype set. The experiments provide an illustration of the findings given in theorem 3.1 and theorem 3.2.

4.1 Simulated Data

In the simulated data set, sixteen classes are assumed, of which the samples are randomly generated from Gaussian distributions. Let ξ and ζ denote two uniform random variables at $[0, 1]$. They determine, respectively, the means and variances of the Gaussian distributions as follows: ξ varies with the change of both the component of a random sample and the membership of the class that the sample belongs to, but does not vary for the same component of different random samples in the same class. The change of ζ only depends on the change of the membership of the class. Consequently the covariance matrix of each random sample of each class is a diagonal matrix. All the diagonal components of every covariance matrix are randomly generated by ζ. The components of each random sample of each class are mutually independent. The random samples of different classes have a different covariance matrix each. Cases with different dimensions of data (dim) and different numbers of prototypes per class (N_c) are examined in Table 1 and Table 2. It is noted that the same random data set is applied to all classifiers. It produces very similar results that multiple runs of randomization and each same set applies to all the classifiers.

Table 1. Accuracy(%) of the NFM in comparison to those based on other points on the feature lines for the simulated data

Dim	$\mathbf{N_c}$	$\lambda = \mathbf{-1}$	$\lambda = \mathbf{-0.5}$	$\lambda = \mathbf{0}$	$\lambda = \mathbf{0.25}$	$\lambda = \mathbf{0.5}$	$\lambda = \mathbf{0.75}$	$\lambda = \mathbf{1}$	$\lambda = \mathbf{1.5}$	$\lambda = \mathbf{2}$
8	6	19.79	31.25	42.71	40.62	**47.92**	40.62	42.71	31.25	19.79
16	6	28.12	35.42	46.88	54.17	**63.54**	54.17	46.88	35.42	28.12
	12	22.40	31.77	50.00	63.02	**64.58**	63.02	50.00	31.77	22.40
32	6	17.71	41.67	55.21	66.67	**68.75**	66.67	55.21	41.67	17.71
	12	24.48	35.42	55.73	73.44	**74.48**	73.44	55.73	35.42	24.48
	24	23.18	40.10	63.54	78.65	**83.07**	78.65	63.54	40.10	23.18
64	6	20.83	34.38	50.00	65.62	**71.88**	65.62	50.00	34.38	20.83
	12	18.75	30.21	55.21	78.65	**83.85**	78.65	55.21	30.21	18.75
	24	20.83	35.68	61.72	85.42	**91.67**	85.42	61.72	35.68	20.83
	48	28.52	45.70	73.70	90.89	**94.53**	90.89	73.70	45.70	28.52
128	6	23.96	37.50	61.46	83.33	**87.50**	83.33	61.46	37.50	23.96
	12	25.52	36.98	56.77	80.21	**89.06**	80.21	56.77	36.98	25.52
	24	15.36	25.00	50.00	86.20	**93.49**	86.20	50.00	25.00	15.36
	48	22.01	33.59	73.44	95.70	**98.70**	95.70	73.44	33.59	22.01
	96	19.01	36.00	72.01	95.44	**98.18**	95.44	72.01	36.00	19.01

Table 1 displays the accuracy of classifications using the NFM ($\lambda = 0.5$) and those based on other points in the feature lines. The NFM yields consistently higher accuracy rates of classifications than all the others.

Table 2. Accuracy(%) of the NFM in comparison to the NFL, NN, and k-NN for the simulated data

Dim	$\mathbf{N_c}$	**NFM**	**NFL**	**NN**	**5-NN**	**10-NN**	**15-NN**
8	6	**47.92**	36.46	36.46	37.50	31.25	31.25
16	6	**63.54**	50.00	50.00	42.71	40.62	20.83
	12	**64.58**	59.90	52.60	47.40	43.23	39.06
32	6	68.75	**69.79**	56.25	51.04	42.71	31.25
	12	**74.48**	72.40	57.29	50.52	45.31	42.71
	24	**83.07**	79.43	62.24	59.38	53.91	52.34
64	6	**71.88**	71.88	51.04	41.67	35.42	20.83
	12	**83.85**	81.77	54.17	48.96	42.71	38.54
	24	**91.67**	90.36	64.84	58.59	57.81	55.47
	48	**94.53**	93.62	74.74	73.70	73.31	72.53
128	6	87.50	**88.54**	60.42	47.92	50.00	42.71
	12	**89.06**	89.06	55.21	50.00	48.44	48.44
	24	**93.49**	93.49	48.70	41.67	40.62	38.02
	48	**98.70**	98.57	74.09	70.18	68.23	66.54
	96	**98.18**	98.11	72.53	66.67	63.48	61.26

Table 2 shows the accuracy of classifications using the NFM, NFL, NN and $k-$NN methods. The NFM also yields consistently higher accuracy rates of classifications than the NFL, NN and $k-$NN.

For $k-$NN, values of k equal to 5, 10 and 15 are tested. As proved in theorem 3.2, under the *i.i.d.* assumption, the NFL is approximately equivalent to the NFM when the dimension of the feature space is high. In Table 2, for example, when the number of prototypes is 6, the difference between accuracy rates of NFM and NFL decreases, although not monotonically, from 11.46% to 1.04% when the dimension increases from 8 to 128. In view of theorem 3.2, our observation is emphasized on the cases when the number of prototypes in each class is finite and the dimension of feature space is relatively high, and therefore the Dim in Table 1 and Table 2 is set to be large. In addition, since the simulated data is generated from a standard Gaussian distribution, the N_c in Table 1 and Table 2 can be set small. In Gaussian case, the NFM is effective with small sample data sets

4.2 Benchmark Data

In what follows, we present the experimental examination of the performance of the NFL classifier on 15 real-life benchmark data sets from the UCI machine learning repository. The experimental results are reported in Table 3, Table 4, and Table 5.

Table 3. Accuracy(%) of the NFM in comparison to those based on other points on the feature lines for 15 benchmark data sets

Data sets	**classes, dim, inst**	$\boldsymbol{\lambda = -1}$	$\boldsymbol{\lambda = -0.5}$	$\boldsymbol{\lambda = 0}$	$\boldsymbol{\lambda = 0.25}$	$\boldsymbol{\lambda = 0.5}$	$\boldsymbol{\lambda = 0.75}$	$\boldsymbol{\lambda = 1}$	$\boldsymbol{\lambda = 1.5}$	$\boldsymbol{\lambda = 2}$
balance	3, 4, 625	52.64	77.12	62.72	90.72	**92.32**	90.72	62.72	77.12	52.64
image	7, 19, 2310	91.73	94.81	96.97	**97.27**	96.36	97.27	96.97	94.81	91.73
ionosphere	2, 34, 351	83.19	86.32	90.88	**93.16**	92.02	93.16	90.88	86.32	83.19
iris	3, 4, 150	83.33	90.00	95.33	**96.00**	95.33	96.00	95.33	90.00	83.33
lenses	3, 4, 24	62.50	70.83	37.50	79.17	**79.17**	79.17	37.50	70.83	62.50
liver	2, 6, 345	54.78	56.23	60.87	63.77	**66.38**	63.77	60.87	56.23	54.78
newthyroid	3, 5, 215	84.65	89.30	97.21	**97.21**	95.81	97.21	97.21	89.30	84.65
pendigits	10, 16, 10992	98.22	99.22	99.45	**99.66**	99.60	99.66	99.45	99.22	98.22
pima	2, 8, 768	65.23	65.62	69.27	69.66	**72.66**	69.66	69.27	65.62	65.23
sonar	2, 60, 208	81.25	85.10	85.10	86.06	**88.94**	86.06	85.10	85.10	81.25
soybean	4, 35, 47	100.00	100.00	100.00	100.00	**100.00**	100.00	100.00	100.00	100.00
spect	2, 22, 267	75.66	78.28	70.41	77.90	**78.65**	77.90	70.41	78.28	75.66
waveform	3, 21, 5000	64.78	69.38	76.78	82.32	**83.42**	82.30	76.74	69.36	64.78
yeast	10, 8, 1484	39.22	45.62	51.28	52.83	**54.18**	52.76	51.08	45.55	39.22
zoo	7, 16, 101	94.06	96.04	96.04	96.04	**96.04**	96.04	96.04	96.04	94.06

Table 3 displays the accuracy of classifications on the 15 data sets using the NFM ($\lambda = 0.5$) and those based on other points in the feature lines. The NFM yields consistently higher accuracies of classifications than all the others.

Table 4 shows the accuracy of classifications on the 15 data sets using the NFM, NFL, NN, and $k-$NN methods. The NFM also yields consistently higher accuracies of classifications than the NFL, NN, and $k-$NN. As in the above simulated data, values of k equal to 5, 10 and 15 are tested for $k-$NN.

The empirical results on the 15 data sets as well further support theorem 3.2, i.e., under the *i.i.d.* assumption the NFL is approximately equivalent to the NFM when the dimension of the feature space is high. For instance, it can be found from Table 4 that in higher dimensional data sets such as soybean and sonar, the difference between accuracy rates of NFM and NFL is very small. In lower dimensional data sets such as balance, iris, and lenses, the difference is relatively large. This fact shows as well that the *i.i.d.* assumption made in theorem 3.2 approximates well for many real-life data sets of high

Table 4. Accuracy(%) of the NFM in comparison to the NFL, NN, and k-NN for 15 benchmark data sets

Data sets	classes, dim, inst	NFM	NFL	NN	5-NN	10-NN	15-NN
balance	3, 4, 625	**92.32**	86.24	62.72	42.08	33.28	27.04
image	7, 19, 2310	**96.36**	96.32	96.06	94.16	92.73	91.21
ionosphere	2, 34, 351	**92.02**	88.60	86.61	84.62	83.76	83.19
iris	3, 4, 150	95.33	88.67	96.00	96.67	**97.33**	96.67
lenses	3, 4, 24	**79.17**	50.00	37.50	16.67	16.67	16.67
liver	2, 6, 345	66.38	59.42	61.45	66.67	67.25	**68.70**
newthyroid	3, 5, 215	**95.81**	90.23	94.88	93.49	89.77	88.37
pendigits	10, 16, 10992	**99.60**	99.45	99.37	99.29	99.05	98.81
pima	2, 8, 768	72.66	68.36	67.97	71.48	73.83	**74.09**
sonar	2, 60, 208	**88.94**	87.50	82.69	82.69	68.75	66.83
soybean	4, 35, 47	**100.00**	100.00	97.87	97.87	76.60	57.45
spect	2, 22, 267	**78.65**	70.01	70.01	46.44	24.72	23.22
waveform	3, 21, 5000	83.42	81.98	78.04	81.86	83.52	**84.48**
yeast	10, 8, 1484	54.18	47.71	49.80	54.58	57.61	**58.56**
zoo	7, 16, 101	96.04	96.04	**98.02**	80.20	45.54	40.59

dimension. The NFM and NFL may be both well suited to the data sets with small number of feature points since they expand the volume of each class in a quadratic way. The error probability of $k-$NN approaches theoretically to that of Bayes classifier when the number of prototypes goes to infinity with k fixed. However, the performance of $k-$NN will decrease when k increases on the data sets with small number of prototypes.

In the simulated data with the Gaussian distribution, the NFM consistently outperforms almost all the other classifiers considered in this paper. Only the classifier with $\lambda = 0.25$ and the NFL come close to it. The more λ is deviating from 0.5, the worse the corresponding classifier performs. As well $k-$NN gives the worse performance with the increasing value of k. It conforms theorem 3.1. In the real-life data, the NFM achieves the best accuracy for 10 of the 15 data sets, followed closely by the classifier with $\lambda = 0.25$, which is very close to the NFM. For the remaining 5 data sets, the classifier with $\lambda = 0.25$ achieves the best performance. This observation indicates that most real-life data sets do not follow Gaussian distributions. They may be approximately Gaussian to certain degrees. We deduce that the better approximation to Gaussian, the better the NFM performs than the other classifiers. Therefore, it remains an interesting question to derive the relationship in terms of the superiority in performance of the NFM, over the other classifiers, on a data set with respect to the degree of its approximation to a Gaussian distribution.

The computation time of the NFM, NFL, NN, and $k-$NN on the 15 benchmark data sets is provided in Table 5. The experiment is executed on the HP AlphaServer SC45 with 44 nodes, each node comprising of four 1GHz Alpha processors with 4GB memory. From Table 5, the NFM can save about 40-50% CPU time compared to the NFL for the 15 benchmark data sets. But it still consumes much more CPU time than the NN and $k-$NN. The NFM takes 408517 seconds to complete the computation on the super computer for the pendigits data set, but achieves the comparable performance of classification with the NN and $k-$NN. The NFM may be not well suited to this kind of data sets with very large number of samples in contrast to the classical NN algorithm. The NFM is very effective as compared to the classical NN algorithm for a data set with small number of samples if its likelihood distribution is a standard or very close to Gaussian.

Table 5. The CPU time (seconds) of the NFM, NFL, NN, and k-NN running on 15 benchmark data sets

Data sets	classes, dim, inst	NFM	NFL	NN	5-NN	10-NN	15-NN
balance	3, 4, 625	84.32	149.57	0.28	1.41	2.86	4.41
image	7, 19, 2310	6719.13	11491.77	19.14	95.32	190.39	285.18
ionosphere	2, 34, 351	161.52	294.43	0.79	4.01	8.03	11.98
iris	3, 4, 150	0.68	1.43	0.01	0.08	0.16	0.24
lenses	3, 4, 24	0.0041	0.0071	0.0003	0.0018	0.0028	0.0041
liver	2, 6, 345	25.94	47.24	0.13	0.68	1.36	2.04
newthyroid	3, 5, 215	5.44	9.51	0.03	0.23	0.44	0.68
pendigits	10, 16, 10992	408517	791779	367	1827	3639	5454
pima	2, 8, 768	369.88	692.53	0.91	4.46	8.89	13.34
sonar	2, 60, 208	53.71	105.36	0.44	2.53	5.01	7.51
soybean	4, 35, 47	0.16	0.26	0.01	0.04	0.09	0.14
spect	2, 22, 267	57.01	101.16	0.23	1.18	2.38	3.56
waveform	3, 21, 5000	188136	359839	104	516	1032	1546
yeast	10, 8, 1484	1158.78	1971.82	3.03	15.16	30.23	45.08
zoo	7, 16, 101	0.63	0.99	0.01	0.11	0.21	0.33

This paper aims to compare the NFM against the NFL in terms of both accuracy and efficiency. The NFM and NFL are both the NN like algorithms, thus we only compare the NFM with the NFL, NN, and $k-$NN, and don't compare it with other classification methods such as C4.5, NB, SVM, etc.

5 Conclusion and Future Directions

In this paper, a pattern classification method named NFM is proposed, as well as a detailed theoretical analysis is conducted and provides insights on why and when the NFM works. A theoretical proof shows that for $n-$dimensional Gaussian distribution, under some reasonable assumptions, the classification using the NFM metric will achieve the least error statistically than those based on any other points on the feature lines. Furthermore, it has been theoretically proved that under the *i.i.d.* assumption, the performance of the NFL is approaching to the NFM when the dimension of the feature space is high. However, the computational complexity of the NFM is significantly less than the NFL. Therefore, the NFM is a good alternative to the NFL in high dimensional feature spaces. This is desirable for the analysis of many real-life data such as microarray data in bioinformatics, where the number of features characterizing some data is in the thousands or tens of thousands. Moreover, the NFM expands, as the NFL does, the representational capacity of a finite number of available feature points, which provides a way to address the *curse of dataset sparsity*.

The experimental evaluation on both simulated and real-life data further shows that the NFM can yield considerably higher accuracies than the classifications based on other points on the feature lines, as well as the NFL, NN, and $k-$NN. This is because the NFM takes advantage of the correlations between prototypes within a class, whereas NN and $k-$NN do not. These forms of the correlations may be such as local linearity. As shown in [20], the global nonlinear structure in some real-life data can be recovered from locally linear fits. The NFM is a linear pattern classification method. Compared to nonlinear models, a linear model is rather robust against over-fitting and is able to provide cost-effective solutions. Overall, the NFM is highly recommended as an alternative to the NFL, NN, and $k-$NN for pattern classification in the case where a data set follows a Gaussian distribution or can be approximated by a Gaussian distribution. The NFM is very effective for a data set with the small number of samples if its likelihood distribution is a standard or very close to Gaussian.

The NFM and NFL are both the NN like algorithms. The NN algorithms are targeted for arbitrary distributions in a nonparametric way. In this paper, we recommend the NFM to be employed in a Gaussian or an approximate Gaussian case only to address the potential use of it. This doesn't mean that the likelihood distribution of a data set must be known and be Gaussian in use of the NFM. For example, the empirical evaluation of the locally adaptive NN [10, 14] is likewise made at several Gaussian cases.

This work can be extended in various directions. Although the NFM metric expands the representative capacity of a finite number of prototypes and thus improves the performance of the NN, it still suffers from bias in high dimensions. It can potentially improve the performance of the NFM further in some cases to develop a locally adaptive NFM. It can be made by estimating feature relevance locally at the query point, and by computing feature midpoints that are elongated along less relevant feature dimensions and constricted along most influential ones. In addition, the NFM is still more expensive computationally than the NN classification. It is under our investigation to develop a novel strategy to reduce significantly the computational complexity of the NFM.

Acknowledgment

We thank the anonymous reviewers for their helpful comments. The work in this paper was mainly done when the first author was with Bioinformatics Research Centre, School of Computer Engineering, Nanyang Technological University.

References

[1] P. J. Bickel and E. Levina. "Some theory for fisher's linear discriminant function, "naive bayes", and some alternatives when there are many more variables than observations". Technical Report #404, University of Michigan, Dept of Statistics, 2003.

[2] C. Blake and C. Merz. UCI repository of machine learning databases. University of California, Irvine, Dept. of Information and Computer Sciences, `http://www.ics.uci.edu/~mlearn/MLRepository.html`, 1998.

[3] T. M. Cover. "Estimation by the nearest neighbor rule". *IEEE Transactions on Information Theory*, 14:50–55, 1968.

[4] T. M. Cover and P. E. Hart. "Nearest neighbor pattern classification". *IEEE Transactions on Information Theory*, 13:21–27, 1967.

[5] S. Dasgupta. *Learning Probability Distribution*. PhD thesis, University of California at Berkeley, 2000.

[6] C. Domeniconi, J. Peng, and D. Gunopulos. "Locally adaptive metric nearest-neighbor classification". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1281–1285, September 2002.

[7] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*. New York: Wiley, 2 edition, 2001.

[8] S. A. Dudani. "The distance-weighted $k-$nearest-neighbor rule". *IEEE Transactions on Systems, Man and Cybernetics*, 6(4):325–327, 1976.

[9] S. Dudoit, J. Fridlyand, and T. P. Speed. "Comparison of discrimination methods for the classification of tumors using gene expression data". *Journal of the American Statistical Association*, 97(457):77–87, 2002.

[10] J. H. Friedman. "Flexible metric nearest neighbor classification". Technical report, Dept. of Statistics , Stanford University, 1994.

[11] K. Fukunaga. *Introduction to statistical pattern recognition*. Academic Press, Boston, 2 edition, 1990.

[12] K. Fukunaga and D. M. Hummels. "Bayes error estimation using parzen and $k-$nn procedures". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):634–643, 1987.

[13] K. Fukunaga and D. M. Hummels. “Bias of nearest neighbor error estimates”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1):103–112, 1987.

[14] T. Hastie and R. Tibshirani. “Discriminant adaptive nearest neighbor classification”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(6):607–616, June 1996.

[15] R. Kass and A. Raftery. “Bayes factors”. *Journal of the American Statistical Association*, 90:773–795, 1995.

[16] E. Levina. *Statistical Issues in Texture Analysis*. PhD thesis, University of California at Berkeley, 2002.

[17] S. Z. Li. “Content-based classification and retrieval of audio using the nearest feature line method”. *IEEE Transactions on Speech and Audio Processing*, 8(5):619–625, September 2000.

[18] S. Z. Li, K. L. Chan, and C. L. Wang. “Performance evaluation of the nearest feature line method in image retrieval”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1335–1339, November 2000.

[19] S. Z. Li and J. Lu. “Face recognition using the nearest feature line method”. *IEEE Transactions on Neural Networks*, 10(2):439–443, March 1999.

[20] S. T. Roweis and L. K. Saul. “Nonlinear dimensionality reduction by locally linear embedding”. *Science*, 290(5500):2323–2326, 2000.

[21] J. Schurmann. *“Pattern classification: a unified view of statistical and neural approaches”*. New York: Wiley, 1996.

Dr. Zonglin Zhou received his PhD in Probability and Statistics from Beijing Normal University in 1994. Since then, he worked as a regular researcher at the Institute of Computing Technology, Chinese Academy of Sciences for more than one year, and spent another about nine years as postdoctoral research associate and research fellow at University of Maryland Baltimore County, National University of Singapore, and Nanyang Technological University, respectively before he currently works as postdoctoral fellow in Department of Computer Science, Hong Kong University of Science and Technology. His research interests mainly involve machine learning and its application in bioinformatics

Dr. Chee Keong Kwoh is the Programme Director, M Sc (Bioinformatics) and Associate Professor, in the Division of Computing Systems and Division of Bioengineering, Nanyang Technological University. His research interests include: Apply statistical learning theory in bioinformatics research; Probabilistic inference; Learning from data and various methodologies in objective learning; Numerical expert systems, particularly in the area of probabilistic reasoning; Biomedical engineering research and applications.