2008 Eighth IEEE International Conference on Data Mining

On Locally Linear Classification by Pairwise Coupling

Feng Chen, Chang-Tien Lu, Arnold P. Boedihardjo
Virginia Polytechnic Institute and State University
7054 Haycock Road, Falls Church, VA, 22043
{chenf, ctlu, arnold.p.boedihardjo}@vt.edu

Abstract

Locally linear classification by pairwise coupling addresses a nonlinear classification problem by three basic phases: decompose the classes of complex concepts into linearly separable subclasses, learn a linear classifier for each pair, and combine pairwise classifiers into a single classifier. A number of methods have been proposed in this framework. However, these methods have two major deficiencies: 1) lack of systematic evaluation of this framework; 2) naive application of clustering algorithms to generate subclasses. This paper proves the equivalence between three popular combination schemas under general settings, defines several global criterion functions for measuring the goodness of subclasses, and presents a supervised greedy clustering algorithm to optimize the proposed criterion functions. Extensive experiments were conducted to validate the effectiveness of the proposed techniques.

1 Introduction

In recent years, there has been an emerging interest to solve a complex (nonlinear) classification problem by using locally linear classification (LLC) techniques [2–5]. The basic idea is to approximate a nonlinear decision boundary by consecutive segments, each of which is determined by a local linear classifier. Results have shown that this approach can achieve competitive generalization accuracy and higher training efficiency than other advanced approaches such as neural network [3], generalized linear discriminative analysis [4], and nonlinear support vector machines [1].

The effectiveness of LLC lies in the fact that each local classifier requires estimating a much simpler target function, thus reducing the chance of overfitting. However, as a potential disadvantage, more target functions need to be estimated with less training data. An implicit assumption of LLC is that the gain acquired by the reduced complexity is more than the loss incurred by the "reduced" training data. LLC includes three major categories: pairwise coupling based (LLC-PC) [2, 3, 5], local space based (LLC-LS) [4], and model based (LLC-MD) [1, 8]. LLC-PC decomposes the classes of complex concepts into linearly separable subclasses, then learns a linear prototype classifier for each pair of subclasses, and finally combines the pairwise prototype classifiers into a single classifier. LLC-LS divides the input space into several disjoint subspaces, and then learns a linear classifier for each subspace. LLC-MD assumes each class as a mixture of normals and learns an LDA classifier by treating each normal as a pseudo-class.

This paper focuses on LLC-PC, the **L**ocally **L**inear **C**lassification by **P**airwise **C**oupling. It is a natural generalization of the state-of-the-art multiclass classification approach by pairwise coupling [7]. Existing methods for LLC-PC apply naive clustering methods (e.g., k-means) to generate subclasses, and present different combination schemas (e.g., voting, MinMax) to integrate pairwise prototype classifiers [2, 3, 5]. Some empirical comparisons demonstrate similar classification accuracy between different combination schemas [3]. However, there is no research presented to explain this phenomenon.

We address two major issues: First, the generation of appropriate subclasses can not be optimally solved by directly applying general clustering algorithms. This is due to the main principle for solving problems using a restricted amount of information: "When solving a given problem, try to avoid solving a more general problem as an intermediate step [9]." A supervised clustering algorithm must be designed by considering the impacts of other phases. Second, there should exist some connections between different combination schemas, in order to explain the fact that they usually exhibit similar classification accuracy. As shown later, the connections lead to a new reformulation of the pairwise coupling problem as a voronoi diagram problem, thus introducing a new direction to further optimize LLC-PC.

The rest of the paper is organized as follows. Section 2 presents preliminaries of LLC-PC. Section 3 defines new criterion functions and discuses their major characteristics. Section 4 presents a greedy subclasses generation algorithm. Experiments and conclusion are discussed in sections 5 and 6, respectively.

1550-4786/08 $25.00 © 2008 IEEE
DOI 10.1109/ICDM.2008.137

2 Preliminaries

This section discusses three popular combination schemas. Suppose there are N classes $\{C_1, C_2, \ldots, C_N\}$, each class C_i $(i = 1, \ldots, N)$ is divided into N_i pseudo clusters $(C_{i1}, C_{i2}, \ldots, C_{iN_i})$, and the separating hyperplane for C_{ij} and C_{kp} is $f_{ij-kp}(\mathbf{x}) = \mathbf{w}_{ij-kp}^T\mathbf{x} + b$. Three popular combination schemas can be summarized as follows:

Voting based: The decision function for the subclass C_{ij} can be defined by $F_{ij}(\mathbf{x}) = \sum_{k\neq i,k=1}^{N}\sum_{p=1}^{N_k}(\delta(f_{ij-kp}(\mathbf{x})))$, where $\delta(z) = 1$ if $z \geq 0$, and 0 otherwise. The decision function for the class C_i can be defined by $F_i(\mathbf{x}) = max(F_{ij}(\mathbf{x})/\sum_{o=1,o\neq i}^{N} N_o)$, where $1 \leq j \leq N_i$, and the denominator is used for normalization, since the number of subclasses generated for each class may be different. The new point $\mathbf{x}$ is classified as follows: $G(\mathbf{x}) = argmax_{i=1,\ldots,N}(F_i(\mathbf{x}))$.

Probability based: The decision function $F_{ij}(\mathbf{x})$ can be defined by $F_{ij}(\mathbf{x}) = Prob(y = C_{ij}|\mathbf{x})$, where the posterior probability $Prob(y = C_{ij}|\mathbf{x})$ can be estimated from the available pairwise class probabilities $Prob_{ij-kp} = Prob(y = C_{ij}|y = C_{ij} \text{ } or \text{ } C_{kp}, \mathbf{x})$ [7]. The decision function $F_i(\mathbf{x})$ is defined by $F_i(\mathbf{x}) = max(F_{ij}(\mathbf{x}))$, where $1 \leq j \leq N_i$. The new point x is classified as follows: $G(\mathbf{x}) = argmax_{i=1,,\ldots,N}(F_i(\mathbf{x}))$.

MinMax based: The decision function $F_{ij}(\mathbf{x}) = min(f_{ij-kp}(\mathbf{x}))$, where $k \neq i$. The decision function $F_i(\mathbf{x}) = max(F_{ij}(\mathbf{x}))$, where $1 \leq j \leq N_i$. The new point x is classified as follows: $G(\mathbf{x}) = argmax_{i=1,,\ldots,N}(F_i(\mathbf{x}))$.

Theorem 2.1 (Equivalence)**.** *Given a new object* $\mathbf{x}$*, if one of the following conditions is true, then* $G(\mathbf{x})_{Voting} = G(\mathbf{x})_{MinMax} = G(\mathbf{x})_{Prob}$*:*

(1) $\exists i, j(1 \leq i \leq N, 1 \leq j \leq N_i)$, $F_{ij}(\mathbf{x})_{Voting} = \sum_{k=1,k\neq i}^{N} N_k$*;*

(2) $\exists i, j(1 \leq i \leq N, 1 \leq j \leq N_i)$, $F_{ij}(\mathbf{x})_{MinMax} > 0$*;*

(3) $\exists i, j(1 \leq i \leq N, 1 \leq j \leq N_i)$, $F_{ij}(\mathbf{x})_{Prob} > F_{kp}(\mathbf{x})_{Prob}$*, where* $k \neq i$*;*

Readers are referred to [11] for a detailed proof.

These three schemas, as well as their equivalence, are illustrated in Figure 1. There are two classes $\{C_1, C_2\}$, and their subclasses are $\{C_{11}, C_{12}\}$ and $\{C_{21}, C_{22}\}$, respectively. For each object $\mathbf{x}$ inside the region $ABCNMA$, the subclass C_{11} wins the competitions against the subclasses C_{21} and C_{22}. Then, $F_{11}(\mathbf{x})_{Voting} = 2$, and $G(\mathbf{x})_{Voting} = C_1$. Because $f_{11-22}(\mathbf{x}) > 0$ and $f_{11-21}(\mathbf{x}) > 0$, $F_{11}(\mathbf{x})_{MinMax} > 0$ and $G(\mathbf{x})_{MinMax} = C_1$. Also, because $Prob(y = C_{11}|\mathbf{x})$ is larger than $Prob(y = C_{21}|\mathbf{x})$ and $Prob(y = C_{22}|\mathbf{x})$, $G(\mathbf{x})_{Prob} = C_1$. Therefore, the three schemas are equivalent inside the region $ABCNMA$. Similarly, the equivalence is held in the regions $CDEONC$, $EFGPOE$, and $GHAMPG$. However, inside the small center region $MNOPM$, the above conditions are not satisfied and therefore the equivalence is not guaranteed.

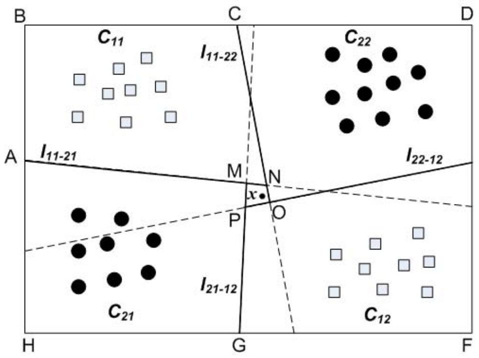


Figure 1: An example of combination schemas

Theorem 2.1 indicates that the three combination schemas are equivalent inside certain regions. As shown in Section 5.1, we empirically verified that these equivalent regions occupy in overall more than 99% of the whole input space. That means, these combination schemas are equivalent in most cases. It explains why different combination schemas usually exhibit similar accuracy.

Another observation is that, since the conflicts rarely happen in practice, we can reasonably assume that Theorem 2.1 is true for the whole space. Under this assumption, the pairwise coupling becomes equivalent to a Voronoi diagram problem [10]. Particularly, each subclass (C_{ij}) has a dominated region (Voronoi polytope), which is bounded by a subset of the related linear prototype classifiers (separating hyperplanes). If a new object $\mathbf{x}$ is within the dominated region of the sub-class C_{ij}, then it is classified to the class C_i. Thus, the pairwise coupling problem can be re-formulated as: "*Given a new object* $\mathbf{x}$*, search for a class region (Voronoi polytope), which contains the object* $\mathbf{x}$." Based on this reformulation, traditional Voronoi techniques [10] can be conveniently adapted to identify the dominated region for each subclass. The significant (necessary) and insignificant (redundant) prototype classifiers can also be easily identified. Redundant prototype classifiers refer to the prototype classifiers that do not contribute to the decision boundary of the resulting combined classifier. In addition, spatial indexing structures (e.g., R-tree) can be utilized to index the subclass regions, such that the classification time cost can be significant reduced.

3 Criterion Functions

This section addresses the criterion functions which can measure the generalization accuracy of the combined classifier, by considering a number of factors such as the division of original classes, the binary classifier model, the combination schema, and the computational cost. Many existing methods directly use general clustering criterion functions (e.g., total intra-cluster variance [1]) to measure the quality of subclasses generated. However, the subclasses that minimize total intra-cluster variance do not necessarily lead to the classifier of high generalization accuracy.

3.1 Mean Piecewise Error Function

The mean piecewise error function can be formalized as:

$$Q = \sum_{(C_{ij},C_{kp})\in \mathbf{U}} \left(P_{ij-kp} E(C_{ij}, C_{kp}) \right), \tag{1}$$

where $\mathbf{U} = \{(C_{ij}, C_{kp}) | 1 \leq i,k \leq N, i \neq k, 1 \leq j \leq N_i, 1 \leq p \leq N_k\}$, N denotes the total number of original classes, N_i denotes the number of subclasses of C_i, P_{ij-kp} denotes the prior probability of the subclass pair (C_{ij}, C_{kp}), and $E(C_{ij}, C_{kp})$ denotes the generalization error between C_{ij} and C_{kp}. The prior probabilities are used as the weights to balance the contributions of different subclasses. We set $P_{ij-kp} = P_{ij} \cdot P_{kp} / \sum_{(C_{ij},C_{kp})\in \mathbf{U}} P_{ij} \cdot P_{kp}$, where $P_{ij} = |C_{ij}|/S$, the ratio of the sample size of C_{ij} to the total sample size.

The selection of the atomic error function $E(C_{ij}, C_{kp})$ depends on the binary classifier model used for the subclasses C_{ij} and C_{kp}. We consider two popular linear classifier models, including Fisher linear discriminant analysis (LDA) and linear support vector machines (SVM). We select an identical classifier model for each pair of subclasses with default parameter settings. Depending on the specific classifier model selected, we abbreviate the related mean piecewise error (MPE) function as MPE-SVM or MPE-LDA. The whole category of MPE functions is abbreviated as MPE.

MPE-LDA selects the inverse of Fisher criterion [1], the ratio of the between-class variance to the within-class variance, as the atomic error function. It can be formalized as

$$Q = \sum (P_{ij-kp} (\mathbf{w}^{\mathrm{t}}_{ij-kp} \mathbf{S}_{W,ij-kp} \mathbf{w}_{ij-kp}) (\mathbf{w}^{\mathrm{t}}_{ij-kp} \mathbf{S}_{B,ij-kp} \mathbf{w}_{ij-kp})^{-1}) \tag{2}$$

, where $\mathbf{S}_{W,ij-kp} = \mathbf{S}_{ij} + \mathbf{S}_{kp}$ and $\mathbf{S}_{B,ij-kp} = (\mathbf{m}_{ij} - \bar{\mathbf{m}})(\mathbf{m}_{ij} - \bar{\mathbf{m}})^{\mathrm{t}} + (\mathbf{m}_{kp} - \bar{\mathbf{m}})(\mathbf{m}_{kp} - \bar{\mathbf{m}})^{\mathrm{t}}$ are the within-class scatter matrix and the between-class scatter matrix, respectively; $\mathbf{S}_{ij}$ is the within-class covariance matrix of subclass C_{ij}, $\mathbf{m}_{ij}$ is the mean vector of subclass C_{ij}, and similar definitions are used for $\mathbf{S}_{kp}$ and $\mathbf{m}_{kp}$. $\bar{\mathbf{m}} = (\mathbf{m}_{ij} + \mathbf{m}_{kp})/2$. $\mathbf{w}_{ij-kp} = \mathbf{S}^{-1}_{W,ij-kp}(\mathbf{m}_{ij} - \mathbf{m}_{kp})$. The definitions of other symbols are consistent with the related definitions for Equation (1).

MPE-SVM selects the error function of a linear SVM model, the addition of the inverse classifier margin to the empirical error, as the atomic error function. It can be formalized as follows:

$$Q = \sum \left(P_{ij-kp} \frac{1}{2} \|\mathbf{w}_{ij-kp}\|^2 \right) + C \sum \left(P_{ij-kp} \sum_{o=1}^{m_{ij-kp}} \zeta_{o,ij-kp} \right), \tag{3}$$

where $\frac{1}{2}\|\mathbf{w}_{ij-kp}\|^2$ and $\zeta_{o,ij-kp}$ refer to the inverse classifier margin and the slack variables for subclasses C_{ij} and C_{kp}, respectively; m_{ij-kp} refers to the number of slack variables, and C denotes a tradeoff parameter. For simplicity, we assume that the tradeoffs of all SVM classifiers are identical. The left part of the equation is the weighted sum of the inverse margins of pairwise SVM classifiers, which can be regarded as the approximate structure error of the combined classifier. The right part of the equation is the weighted sum of the slack variables of pairwise SVM classifiers, which can be viewed as the approximate empirical error of the combined classifier. The parameter C is used to balance the contributions of the classifier margin and the empirical error.

3.2 Major Characteristics

This subsection evaluates the correlation between the proposed criterion functions and the cluster granularity, and conducts a comparison between these criterion functions.

Theorem 3.1 (Monotonicity of MPE-SVM)**.** *Given a data set of N classes $(C_1, ..., C_N)$, suppose each class C_i has N_i subclasses, then the value of MPE-SVM can be decreased by randomly decomposing one subclass into two smaller-size subclasses.*

Theorem 3.2. *Given a data set of N classes, the values of the criterion functions MPE-SVM and MPE-LDA are minimized if the maximum number of subclasses are generated for each class.*

Theorem 3.3. *Given a data set of $N(N > 1)$ classes, if the maximum number of subclasses are generated for each class, then the resulting classifier is equivalent to a 1-nearest-neighbor classifier.*

Readers are referred to [11] for detailed proofs.

MPE-LDA vs. MPE-SVM

First, we consider the case when each class only contains one cluster (the lowest cluster granularity). In this case, these two functions degeneralize to LDA and SVM, respectively. Results have been shown that in overall SVM can achieve higher classification accuracy than LDA [1]. The possible reason is that SVM considers both empirical error and structure capacity and is based on recent advances in statistical learning theory [9]. In comparison, LDA assumes that each class is normally distributed with common covariances. This assumption is usually not held in real applications. However, LDA is much more efficient to compute and easier to understand than SVM. Particularly, LDA and SVM have the time complexities of $O(d^2n)$ and $O(d^2n^{\delta})$, respectively, where d refers to the dimension cardinality, n refers to the training sample size, and $\delta > 1$.

Second, we consider the case when some classes have more than one subclass. In this case, MPE-SVM appears more stable than MPE-LDA. As shown in Theorem 3.1, MPE-SVM has the important characteristic of monotonicity with respect to the total number of subclasses. It is also more resilient to outliers. In comparison, MPE-LDA does not have the feature of monotonicity and requires calculating the inverse of the within-class scatter matrix for each pair of subclasses. If some subclasses have singular covariance matrixes (e.g., outlier classes or the classes with correlated attributes), then the total score of MPE-LDA will be affected. The selection of MPE-LDA or MPE-SVM is not necessarily dependent on the classifier model used in the pairwise prototype classifiers. For example, in the scenario of limited computation, MPE-LDA may be used as the criterion function to guide the generation of subclasses, even though SVM is used latter to build the pairwise prototype classifiers.

Characteristics of MPE

As demonstrated in Section 5, MPE exhibits much higher accuracy than general clustering criterion functions (e.g., total intra-cluster variance). However, it still has several limitations: 1) **Dependence on Cluster Granularity**. Theorem 3.2 indicates that MPE can always get the minimal value at the highest cluster granularity. According to Theorem 3.3, the combined classifier degeneralizes to a 1-nearest-neighbor classifier. It implies the requirement of a predefined total number of subclasses to be generated. Otherwise, the criterion functions may not be useful to find meaningful subclasses. 2) **Inappropriate for a Large Number of Subclasses**. The total number of prototype classifiers is quadratically increased with the total number of subclasses. When the number is high, the differences between the error scores of prototype classifiers will be neutralized. As a result, MPE will become insensitive to different generalizations of subclasses.

Variants of MPE

To alleviate the negative impacts of the large number of prototype classifiers, we can redefine the set **U** (see equation (1)) as a small set of representative prototype classifiers. Depending on the different definitions of the representative classifiers, several variants of MPE can be derived. Due to lack of space, we only briefly present two major variants.

The first variant is called **R**efined MPE (R-MPE), which defines **U** as the set of necessary prototype classifiers. As discussed in Section 2, by assuming that Theorem 2.1 is true for the whole space, the pairwise coupling can be reformulated as a Voronoi diagram problem. Based on this reformulation, many prototype classifiers are actually redundant when the data is in a low-dimensional space (e.g., smaller than 10 dimensions). For example, suppose there are totally N subclasses in a 2-dimensional space, then the number of necessary prototype classifiers is smaller than $(3N-6)$ [10]. That means, even there are $O(N^2)$ prototype classifiers, only a linear number of classifiers contribute to the decision boundary of the resulting classifier.

Another variant is named **S**ymmetric **N**earest **N**eighbor based MPE (SNN-MPE), which defines **U** as the pairs of subclasses which are symmetric k-nearest neighbors. We use the Euclidean distance between the centers of two subclasses as the proximity metric. The subclasses of a same parent class are not considered as neighbors. The effectiveness of SNN-MPE is based on an important observation that the significant prototype classifiers are usually related to the pairs of subclasses, which are close to each other. SNN-MPE provides a parameter k to allow users to balance the tradeoff between the computational cost and the accuracy.

4 A Greedy Clustering Algorithm

To evaluate the effectiveness of the proposed criterion functions, this section presents a simple but effective supervised clustering algorithm named Greedy-MPE. It generates the subclasses in a greedy manner to minimize the criterion functions (MPE). The algorithm is described as follows:

Algorithm (Greedy-MPE). Given a data set of N classes $\{C_1,\ldots,C_N\}$ and the total number (K) of subclasses to be generated,

1. Regard each class as a single cluster (subclass).
2. From the set **U** of subclass pairs, search for a pair of subclasses (C_{ij},C_{kp}) that has the maximum weighted classification error $F(C_{ij},C_{kp})$. The maximum weighted classification error indicates that this pair of subclasses is currently most linearly inseparable and hence can be regarded as the priority candidate subclasses for further decompositions.
3. Select a subclass from C_{ij} and C_{kp}, which has the highest intra-class variance, and decompose it into two smaller-size subclasses.
4. If the total number of the subclasses generated is smaller than K, go to step 2. Otherwise, output the current subclasses and terminate the algorithm.

The set **U** of candidate subclass pairs is determined by a specific criterion function, which the algorithm greedily minimizes. For example, for MPE, **U** refers to the pairs of subclasses, which do not have the same parent class label. For SNN-MPE, **U** refers to the pairs of subclasses, which are symmetric k-nearest neighbors. $F(C_{ij},C_{kp}) = P_{ij-kp} * E(C_{ij},C_{kp})$, where P_{ij-kp} refers to the prior probability of the subclass pair (C_{ij},C_{kp}), and $E(C_{ij},C_{kp})$ refers to the classification error between C_{ij} and C_{kp}. In the step 3, traditional clustering algorithms (e.g., k-means) can be used to decompose the selected subclass into two smaller-size subclasses.

The key issue of Greedy-MPE is to select an appropriate subclass in each iteration for further splits. The current selection bias is to prefer a subclass which is not well-separatable from others and has a high intra-cluster variance. Two alternative selection biases may also be considered. The first is to prefer a subclass which has the highest aggregated classification error over the related subclass pairs: $\arg\max_{C_{ij}}(\sum_{k\neq i} P_{ij-kp}E(C_{ij},C_{kp}))$. The second is to prefer a subclass which has the maximum gain of MPE score: $\arg\max_{C_{ij}}(Q_{before\_splitting\_C_{ij}} - Q_{after\_splitting\_C_{ij}})$, where $Q_{before\_splitting\_C_{ij}}$ refers to the MPE score before splitting the subclass C_{ij}, and $Q_{after\_splitting\_C_{ij}}$ refers to the MPE score after splitting the subclass C_{ij}.

5 Experiment

This section demonstrates the equivalence between three popular combination schemas under general settings (Theorem 2.1), and compares the performances of the resulting classifiers produced by different clustering methods.

Experimental Tools. We used linear SVM as the prototype classifier and four different clustering algorithms to generate subclasses: Greedy-MPE, *k*-means, hierarchical clustering (HC), and EM clustering. The major settings were as follows: 1) Euclidean distance was used as the proximity metric, 2) the parameter "replicates" for *k*-means (number of times to repeat the clustering) was set to 10, 3) the link metric in the HC clustering algorithm was set to average link, and 4) the tradeoff parameter (C) for linear SVM was set to 100. The default combination schema was the voting-based. For *k*-means, HC, and EM, we generated the same number of subclasses for each class.

Experimental Data Sets. In our experiments, we used 22 benchmark data sets provided by UCI, STATLOG, DELVE, and LIBSVM data repositories: flare solar, thyroid, breast cancer, breast-w, pima-diabetes, heart, image, ringnorm, twonorm, waveform, german, diabetis, fourclass, svmguide1, vehicle, page-block, segment, glass, satimage, pendigits, optdigits, and letter. Among these data sets, the range of class numbers is [2, 26], and the range of dimensions is [2, 60]. Table 1 shows the detailed information of six representative data sets. We generated 100 random partitions into training and test sets (mostly 60%:40%). On each partition, we trained a classifier and then calculated its test accuracy. The mean accuracy over all partitions was reported. We considered the settings of cluster granularity (the total number of subclasses) from 1 to 40.

Table 1: Some characteristics of experimental data sets

Dataset	Source	#Objects	#features	#classes
Thyroid	UCI	140:75	3	2
Flare solar	UCI	666:400	9	2
Image	UCI	1300:1010	18	2
Glass	UCI	128:86	9	6
Ringnorm	DELVE	400:7000	20	2
Fourclass	LIBSVM	517:345	2	2

Note: The numbers before and after ":" are for training and testing, respectively.

5.1 Combination Schemas

This subsection validates the equivalence between three popular combination schemas (voting based, probability based, and MinMax) on the generalization accuracy. As discussed in Section 2, these three combinations are provably equivalent inside certain regions, which empirically constitute a majority of the input space. To evaluate the percentage of the provable equivalent area to the whole space, we used k-means to generate subclasses and calculated the rate of training and testing objects, which were within the provable equivalent area. Figure 2 shows the experimental results on the twenty-two benchmark data sets. The *X-axis* refers to the total number of subclasses generated and the *Y-axis* refers to the rate of training and testing objects which are within the provable equivalent area. In the figure, there are totally 306 sample points, and each sample point denotes the result of a data set under a specific cluster granularity. A linear regression line was generated to show the correlation between the provable equivalent rate and the cluster granularity. The results indicate that on average more than 99% of objects are within the provable equivalent area. Another observation is that the provable equivalent rate has a tendency of decreasing when the cluster granularity increases. That means, when the cluster granularity is extremely high (e.g., 200), these schemas will be significantly different. However, as shown later, the optimal number of subclasses is usually smaller than 40 in practice.

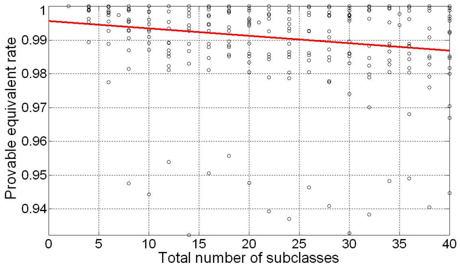
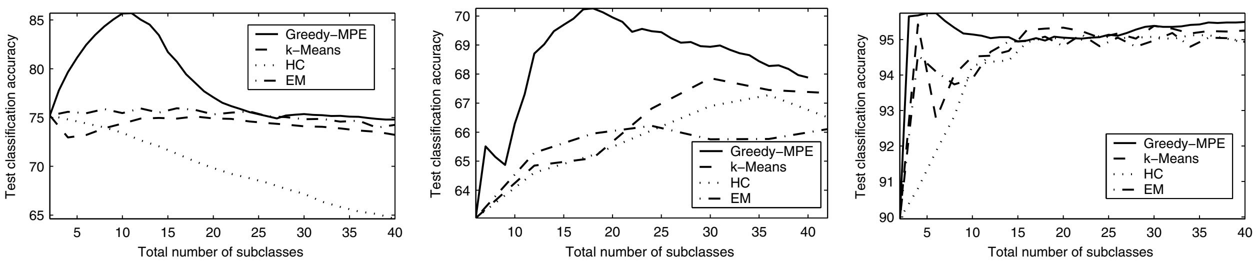


Figure 2: Provable equivalent rate

Theorem 2.1 is the sufficient but not necessary condition of the equivalence. The objects which do not satisfy Theorem 2.1 are still possibly equivalent for these combination schemas. We observed that the actual equivalent rate is much higher than the provable equivalent rate. For example, among all the tested data sets, the actual equivalent rate between the voting based and MinMax is 0.999 ± 0.002. As to the non-equivalent objects, in which the voting based and MinMax reported different results, these two schemas have the test accuracies close to a random assignment. For instance, among fourteen binary data sets, the voting based and MinMax schemas have the test accuracies of 0.52 ± 0.33 and 0.48 ± 0.33, respectively, on the non-equivalent objects.

5.2 Subclass Generation

This subsection compares the performances of the LLC-PC classifiers led by Greedy-MPE and three popular clustering algorithms, k-means, HC, and EM. Figure 3 shows par-



(a) Ringnorm (b) Glass (c) Thyroid

Figure 3: Comparison on test classification accuracy

tial results on test classification accuracy. The *X-axis* refers to the total number of subclasses, and the *Y-axis* refers to the test accuracy. The results indicate that Greedy-MPE is more accurate and stable than general clustering algorithms in most settings. For example, on the data set ringnorm, the optimal test accuracy of Greedy-MPE is 10% higher than those of the other algorithms. Note that, the optimal test accuracy refers to the highest test accuracy over all the settings. A possible explanation to this superiority is that Greedy-MPE is guided by the criterion function MPE. Because MPE is specifically designed to measure the generalization error of an LLC-PC classifier, a greedy division of the training data to minimize MPE can be regarded as a greedy division strategy to minimize the generalization error. Thus, the overall good (but not optimal) accuracy and stability are guaranteed.

In comparison, general clustering algorithms exhibit inconsistent performances on different data sets. For example, the HC clustering algorithm can achieve comparable optimal test accuracies to the others on thyroid, however, its optimal test accuracy on ringnorm is 10% less than Greedy-MPE. As shown in Figure 3, this pattern of inconsistency is also exhibited in all the settings. It is important to compare the algorithms over all the settings, since in practice it is difficult to accurately estimate the optimal number of subclasses. Other tested benchmark data exhibit similar trends. Readers are referred to [11] for the experimental results on more data sets (e.g., image, fourclass, flare solar), and the time cost comparison between different algorithms.

6 Conclusion and Future Work

This paper conducts a systematic and experimental evaluation of LLC-PC, including the equivalence between different combination schemas, the criterion functions, and the sub-class generation algorithms. In the future, we plan to conduct empirical comparisons between LLC-PC and other categories, LLC-LS and LLC-MD, and summarize the appropriate applications for each one. We will also study the theoretical connections between different categories and design a general framework for LLC.

References

[1] T. Hastie, R. Tibshirani, and J. Friedman. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. *Springer*, 2001.

[2] B. Schulmeister and F. Wysotzki. Dipol - a hybrid piecewise linear classifier. *Machine Learning and Statistics: the Interface*, 133-151, New York, John Wiley and Sons, Inc, 1997.

[3] B.L. Lu and M. Ito. Task decomposition and module combination based on class relations: a modular neural network for pattern classification. *IEEE Transaction on Neural Networks*, 10(5), 1999.

[4] T.K. Kim and J. Kittler. Locally Linear Discriminant Analysis for Multimodally Distributed Classes for Face Recognition with a Single Model Image. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 27(3):318-327, 2005.

[5] J.J Wu, H. Hui, W. Peng, and J. Chen. Local Decomposition for Rare Class Analysis. In *Proc. of the 13th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD)*, 814-823, 2007.

[6] P. Geibel, U. Brefeld, and F. Wysotzki. Perceptron and SVM learning with generalized cost models. *Journal of Intelligent Data Analysis*, 8(5):439-455, 2004.

[7] T.F. Wu, C.J. Lin, and R.C. Weng. Probability Estimates for Multi-class Classification by Pairwise Coupling. *Journal of Machine Learning Research*, 975-1005, 2004.

[8] C. Fraley, A.E. Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 611-631, 2002.

[9] V.N. Vapnik. The nature of statistical learning theory. Springer-Verlag, New York, 1995.

[10] F. Aurenhammer. Voronoi Diagrams - A Survey of a Fundamental Geometric Data Structure. *Journal of ACM Computing Surveys*, 23:345-405, 1991.

[11] F. Chen, C.T. Lu, and A.P. Boedihardjo. On Locally Linear Classification by Pairwise Coupling. *Technical Report TR-08-20*, Department of Computer Science, Virginia Tech, 2008.