Dissimilarity Learning for Nominal Data

Victor Cheng[a],Chun-Hung Li[b*], James T. Kwok[c†] and Chi-Kwong Li[a]

[a]Department of Information and Electronic Engineering,
Hong Kong Polytechnic University, Hong Kong

[b]Department of Computer Science,
Hong Kong Baptist Unversity, Hong Kong

[c]Department of Computer Science
Hong Kong University of Science and Technology
Hong Kong

Defining a good distance (dissimilarity) measure between patterns is of crucial importance in many classification and clustering algorithms. While a lot of work has been performed on continuous attributes, nominal attributes are more difficult to handle. A popular approach is to use the value difference metric (VDM) to define a real-valued distance measure on nominal values. However, VDM treats the attributes separately and ignores any possible interactions among attributes. In this paper, we propose the use of adaptive dissimilarity matrices for measuring the dissimilarities between nominal values. These matrices are learned via optimizing an error function on the training samples. Experimental results show that this approach leads to better classification performance. Moreover, it also allows easier interpretation of (dis)similarity between different nominal values.
keywords: nominal attributes, pattern classification, dissimilarities, distance measure, classifiers

1. Introduction

Many pattern recognition algorithms rely on the use of a pairwise similarity (e.g., inner product) or dissimilarity (e.g., distance) measure between patterns. Examples include the nearest neighbor classifiers, radial basis function networks, k-means clustering and, more recently, kernel methods [1,2]. For patterns with continuous (quantitative) attributes, a variety of distance metrics have been widely studied. For example, for two patterns $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^m$, common choices include the Euclidean distance $\sqrt{\sum_{a=1}^{m}(x_{ia} - x_{ja})^2}$ and the Minkowski distance $(\sum_{a=1}^{m}(x_{ia} - x_{ja})^q)^{\frac{1}{q}}$. More generally, to cater for the different contributions and possible correlations among attributes, one can use a generalized version of the Euclidean distance $\sqrt{(\mathbf{x}_i - \mathbf{x}_j)'\mathbf{A}^{-1}(\mathbf{x}_i - \mathbf{x}_j)}$, where $\mathbf{A}$ is a positive definite matrix.

*The support from HKBU research grant 30-01-267 and 30-02-260 is acknowledged.
†The support from HKUST research grant HKUST6195/02E and HKUST 2033/00E is acknowledged.

Various methods have been proposed to determine **A**. For example, one can set **A** to be the covariance matrix, leading to the so-called *Mahalanobis distance*. Another possibility is to maximize the ratio between intra-class variance and inter-class variance, either globally [1] or locally [3]. A survey of these feature weighting methods can be found in [4].

However, these metrics are only defined on continuous attributes. When the attributes are nominal (categorical)[3], definitions of the similarity (dissimilarity) measures become less trivial [5]. A simple but commonly used measure is the *overlap metric* [6]. Under this metric, for two possible values v_i and v_j, the distance is defined as zero when v_i, v_j are identical and one otherwise. For binary attributes, this overlap metric reduces to the so-called *Hamming distance*. A number of variants, each using a different weighting factor, have also been proposed (e.g., [7,8]). However, this overlap metric and its variants assume that all attribute values are of equal distance from each other, and thus cannot represent value pairs with differing degrees of similarities. As an example, for the attribute taste, it may be more desirable to have the value sour closer to sweet than to bitter. Hence, a real-valued distance metric is often preferred over a Boolean one.

Moreover, for nominal attributes with many possible values, a popular approach is to first transform this to a long list of binary attributes, and then apply the overlap metric (or one of its variants) to measure the distance. For example, in a movie classification system, the cast attribute will be transformed into a set of binary attributes such as "cast includes Dustin Hoffman", "cast includes Bruce Willis", "cast includes Leonardo DiCaprio", etc. The feature dimensionality may thus increase dramatically and the curse of dimensionality [9] will become an important issue.

A very popular real-valued metric for nominal attributes is the *value difference metric* (VDM) [6] (and its variants [12,13,17]). For two attribute values v_i and v_j, their distance is defined as

$$d(v_i, v_j) = \omega(v_i) \sum_{c \in C} (P(c|v_i) - P(c|v_j))^2,$$

where C is the set of all class labels, $P(c|v)$ is the conditional probability of class c given v, and $\omega(v_i) = \sqrt{\sum_{c \in C} P(c|v_i)^2}$ is a weighting factor, which attempts to give higher weight to an attribute value that is useful in class discrimination. Note that VDM is actually not a metric as the weighting factor is not symmetric. Moreover, another problem is that it implicitly assumes attribute independence. A simple example demonstrating this problem is the XOR data. VDM will then yield zero distance among all attribute values, which is clearly undesirable. Hence, its performance will deteriorate when correlations among attributes are significant. Besides, in practice, the class conditional probabilities are unknown and have to be estimated from the training data, as

$$P(c|v_i) = \frac{\text{number of training samples with attribute value } v_i \text{ and belonging to class } c}{\text{number of training samples with attribute value } v_i}.$$

[3]An attribute is *nominal* if it can take one of a finite number of possible values and, unlike *ordinal* attributes, these values bear no internal structure. An example is the attribute taste, which may take the value of salty, sweet, sour, bitter or tasteless. When a nominal attribute can only take one of two possible values, it is usually called *binary* or *dichotomous*.

This density estimation may be inaccurate, especially when the available training samples are scanty.

On the other hand, decision tree classifiers [10] can handle nominal attributes naturally, by side-stepping the issue of defining distances altogether. However, as attributes are considered only one at a time during node splitting (typically by using the information gain or gain ratio), decision trees can again perform poorly in problems with high correlations among attributes. Besides, they are not good at handling continuous attributes. Typically, these attributes have to be pre-processed by discretizing into a finite number of intervals [15], which inevitably incurs a loss of information.

In this paper, we attempt to learn the dissimilarities between the values of a nominal attribute directly. This is analogous to the works on distance metric learning for continuous attributes [3,16,18,19]. The rest of this paper is organized as follows. Section 2 describes the proposed method by introducing the notion of adaptive dissimilarity matrices. Experimental results are presented in Section 3, and the last section gives some concluding remarks.

2. Adaptive Dissimilarity Matrix

Suppose that we are given a training set $\mathcal{S} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, with input $\mathbf{x}_i = (x_{i1}, \ldots, x_{im})$ having m attributes and $y_i \in \{-1, +1\}$ is the corresponding class label. We first consider the case where all these m attributes are nominal. Assume that a particular attribute a can take values in $\mathcal{V}_a = \{v_{a1}, \ldots, v_{an_a}\}$. In the following, we attempt to define a dissimilarity measure on each of these $\mathcal{V}_a$'s. In general, a dissimilarity measure d on $\mathcal{V}$ is a real-valued function on $\mathcal{V} \times \mathcal{V}$ such that

$$0 = d(v_i, v_i) \leq d(v_i, v_j) = d(v_j, v_i) < \infty, \quad \forall v_i, v_j \in \mathcal{V}. \tag{1}$$

For attribute a, we will construct an $n_a \times n_a$ non-negative, symmetric, real-valued matrix $\mathbf{M}_a$ where its (α, β)th entry, $\mathbf{M}_{a,\alpha\beta} = \mathbf{M}_a(v_{a\alpha}, v_{a\beta})$, represents the dissimilarity between two values $v_{a\alpha}, v_{a\beta} \in \mathcal{V}_a$. Obviously, the diagonal elements $\mathbf{M}_{a,\beta\beta}$'s are zero because of (1). For a total of m attributes, we thus have a total of m such dissimilarity matrices. As will be discussed later, these will be learned based on the empirical data and so they are called *adaptive dissimilarity matrices* (or ADM's) in the sequel. Notice that, unlike the overlap metric, the distance between any two attribute values is real-valued. Hence, while the overlap metric mandates that all attribute values are equally similar (dissimilar) to each other, here, we can have a value pair (v_i, v_j) being "more similar" than another pair (v_k, v_l). Besides, this relationship is transitive, i.e., if (v_i, v_j) is more similar than (v_k, v_l) which in turn is more similar than (v_m, v_n), then (v_i, v_j) will also be more similar than (v_m, v_n). However, unlike a metric, a dissimilarity measure may not satisfy the triangle inequality. For more detailed discussions on similarity/dissimilarity measures, interested readers are referred to [5].

As the $\mathbf{M}_a$'s are non-negative matrices, we will write them as $\mathbf{M}_a = \mathbf{F}_a \odot \mathbf{F}_a$ for some real-valued matrix[4] $\mathbf{F}_a$, where $\odot$ denotes the Hadamard product[5]. For any two $\mathbf{x}_i, \mathbf{x}_j$,

[4] As $\mathbf{M}_{a,\beta\beta} = 0$, we also have $\mathbf{F}_{a,\beta\beta} = 0$ for all β.

[5] For $\mathbf{A} = [a_{ij}]$ and $\mathbf{B} = [b_{ij}]$, their Hadamard product (also known as the Schur product or elementwise product) is defined as $\mathbf{A} \odot \mathbf{B} = [a_{ij} b_{ij}]$.

the aggregate dissimilarity d^2 (which corresponds to the squared distance when $\mathbf{x}_i, \mathbf{x}_j$ are real-valued) is then defined as

$$d^2(\mathbf{x}_i, \mathbf{x}_j) = \sum_{a=1}^{m} d_a^2(\mathbf{x}_i, \mathbf{x}_j), \tag{2}$$

where

$$d_a^2(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{M}_a(x_{ia}, x_{ja}) = \mathbf{F}_a^2(x_{ia}, x_{ja}). \tag{3}$$

Our task is to learn the $\mathbf{F}_a$'s based on the training set $\mathcal{S}$.

2.1. Learning the Dissimilarities

The basic idea of our method is to first split the training set into two parts, $\mathcal{D}_1$ and $\mathcal{D}_2$. We use $\mathcal{D}_1$ to build a classifier, $f(\mathbf{x})$, and then minimize the classifier's error on $\mathcal{D}_2$ w.r.t. to the entries in $\mathbf{F}_a$'s. The error function can be any differentiable error function suitable to the problem domain. A common choice is the squared error:

$$\mathcal{E} = \frac{1}{2} \sum_{\mathbf{x}_i \in \mathcal{D}_2} (y(\mathbf{x}_i) - f(\mathbf{x}_i))^2. \tag{4}$$

As we have a total of $\sum_{a=1}^{m} \frac{1}{2} n_a(n_a - 1)$ dissimilarity values to be learned, regularization may be introduced to avoid over-fitting [1]. For example, $\mathcal{E}$ in (4) can be modified to:

$$\mathcal{E} = \frac{1}{2} \sum_{\mathbf{x}_i \in \mathcal{D}_2} (y(\mathbf{x}_i) - f(\mathbf{x}_i))^2 + \gamma \sum_{a=1}^{m} \|\mathbf{D}_a - \mathbf{F}_a\|^2, \tag{5}$$

where $\gamma > 0$ is a regularization constant, $\mathbf{D}_a$ is an $n_a \times n_a$ dissimilarity matrix corresponding to the traditional overlap metric (i.e., its diagonal elements are zero, while its off-diagonal elements are 1), and $\|\cdot\|$ denotes some appropriate matrix norm (such as the Euclidean norm). This regularizer thus favors $\mathbf{F}_a$'s being close to $\mathbf{D}_a$. In general, minimization of $\mathcal{E}$ in (4) or (5) will lead to a nonlinear optimization problem, and methods such as gradient descent can be employed. Moreover, notice that as all the $\mathbf{F}_a$'s are learned together, any possible correlation among the attributes can be taken into account during the learning process.

This proposed approach can be applied to various classifiers requiring a dissimilarity measure on the attribute values. For illustration, here we consider using the squared error function in (4) and a radial basis function (RBF) classifier of the form:

$$f(\mathbf{x}) = \sum_{\mathbf{x}_i \in \mathcal{D}_1} w(\mathbf{x}, \mathbf{x}_i) y(\mathbf{x}_i),$$

where

$$w(\mathbf{x}, \mathbf{x}_i) = \exp\left(-\frac{1}{2\sigma^2} d^2(\mathbf{x}, \mathbf{x}_i)\right) \tag{6}$$

and σ is the width of the RBF unit. Here, we use different values of σ at different $\mathbf{x}$'s, as:

$$\sigma(\mathbf{x}) = \frac{1}{n(\mathcal{D}_1)} \sum_{\mathbf{x}_i \in \mathcal{D}_1} d(\mathbf{x}, \mathbf{x}_i), \tag{7}$$

where $n(\mathcal{D}_1)$ is the number of patterns in $\mathcal{D}_1$. The class label of $\mathbf{x}$ is then given by:

$$o(\mathbf{x}) = \begin{cases} 1 & f(\mathbf{x}) > 0, \\ -1 & \text{otherwise.} \end{cases}$$

For gradient descent, we have to obtain the derivative of $\mathcal{E}$ in (4) w.r.t. $\mathbf{F}_{a,\alpha\beta}$ (where $\alpha \neq \beta$). This can be easily computed as:

$$\begin{aligned} \frac{\partial \mathcal{E}}{\partial \mathbf{F}_{a,\alpha\beta}} &= -\sum_{\mathbf{x}_i \in \mathcal{D}_2} (y(\mathbf{x}_i) - f(\mathbf{x}_i)) \frac{\partial f(\mathbf{x}_i)}{\partial \mathbf{F}_{a,\alpha\beta}} \\ &= -\sum_{\mathbf{x}_i \in \mathcal{D}_2} (y(\mathbf{x}_i) - f(\mathbf{x}_i)) \sum_{\mathbf{x}_j \in \mathcal{D}_1} y(\mathbf{x}_j) \frac{\partial w(\mathbf{x}_i, \mathbf{x}_j)}{\partial \mathbf{F}_{a,\alpha\beta}}, \end{aligned} \tag{8}$$

where

$$\begin{aligned} \frac{\partial w(\mathbf{x}_i, \mathbf{x}_j)}{\partial \mathbf{F}_{a,\alpha\beta}} &= w(\mathbf{x}_i, \mathbf{x}_j) \frac{\partial}{\partial \mathbf{F}_{a,\alpha\beta}} \left(-\frac{d^2(\mathbf{x}_i, \mathbf{x}_j)}{2\sigma^2(\mathbf{x}_i)} \right) \\ &= w(\mathbf{x}_i, \mathbf{x}_j) \left(-\frac{d(\mathbf{x}_i, \mathbf{x}_j)}{\sigma^2(\mathbf{x}_i)} \cdot \frac{\partial d(\mathbf{x}_i, \mathbf{x}_j)}{\partial \mathbf{F}_{a,\alpha\beta}} + \frac{d^2(\mathbf{x}_i, \mathbf{x}_j)}{\sigma^3(\mathbf{x}_i)} \cdot \frac{\partial \sigma(\mathbf{x}_i)}{\partial \mathbf{F}_{a,\alpha\beta}} \right), \end{aligned} \tag{9}$$

on using (6). Now, from (2) and (3), we obtain

$$\begin{aligned} \frac{\partial d(\mathbf{x}_i, \mathbf{x}_j)}{\partial \mathbf{F}_{a,\alpha\beta}} &= \frac{1}{2d(\mathbf{x}_i, \mathbf{x}_j)} \left(2\mathbf{F}_a(x_{ia}, x_{ja}) \frac{\partial \mathbf{F}_a(x_{ia}, x_{ja})}{\partial \mathbf{F}_{a,\alpha\beta}} \right) \\ &= \begin{cases} \frac{1}{d(\mathbf{x}_i, \mathbf{x}_j)} \mathbf{F}_{a,\alpha\beta} & (x_{ia} = v_{a\alpha} \text{ and } x_{ja} = v_{a\beta}) \text{ or} \\ & (x_{ia} = v_{a\beta} \text{ and } x_{ja} = v_{a\alpha}), \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{10}$$

Finally, from (7), we have

$$\frac{\partial \sigma(\mathbf{x}_i)}{\partial \mathbf{F}_{a,\alpha\beta}} = \begin{cases} \frac{1}{n(\mathcal{D}_1)} \sum_{\mathbf{x}_j \in \mathcal{D}_1} \frac{1}{d(\mathbf{x}_i, \mathbf{x}_j)} \mathbf{F}_{a,\alpha\beta} & (\mathbf{x}_{ia} = v_{a\alpha} \text{ and } x_{ja} = v_{a\beta}) \text{ or} \\ & (x_{ia} = v_{a\beta} \text{ and } x_{ja} = v_{a\alpha}), \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

Substituting (9), (10) and (11) into (8), we can thus obtain $\frac{\partial \mathcal{E}}{\partial \mathbf{F}_{a,\alpha\beta}}$ for gradient descent.

In situations where both nominal and continuous attributes exist, we simply need to modify the definition in (3) to

$$d_a^2(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} (x_{ia} - x_{ja})^2 & \text{attribute } a \text{ is continuous,} \\ \mathbf{F}_a^2(x_{ia}, x_{ja}) & \text{attribute } a \text{ is nominal.} \end{cases}$$

This approach for measuring dissimilarities between different nominal attributes can also be readily applied to other classifiers as it allows more accurate representation of the

relationships between the nominal attributes. For example, in the case of using the radial basis function as kernel in support vector machine [2], the distance between two feature vectors with nominal attributes can computed using the suggested approach instead of using the Hamming distance which ignores different possible relationships among the different nominal values. The improved distance measures can lead to more accurate measurement in the radial basis functions which can then be classified using the support vector machine.

3. Experiments

3.1. Experiments with Radial Basis Function Classifiers

In this Section, we perform experiments on five data sets (Table 1) from the UCI machine learning repository [11], using the RBF classifier (with 8 basis functions) as described in Section 2. In each problem, all patterns with missing attribute values are first removed. We then use two thirds of the training patterns as $\mathcal{D}_1$ and the remaining as $\mathcal{D}_2$. The diagonal elements of each $\mathbf{F}_a$'s are set to zero (and are not adapted), while the off-diagonal elements are initialized around one (and thus resembles the overlap metric). Minimization of the squared error in (4) will be performed by using gradient descent, which will stop when changes in the norms of all $\mathbf{F}_a$'s are small. Typically, we observe that only a small number of iterations (say, 10-30) are required. Our results are compared with the decision tree classifier C4.5 [10] and RBF classifiers using the overlap metric and the VDM. To reduce statistical variability, results reported here are based on averages over 10 random partitionings of $\mathcal{D}_1$ and $\mathcal{D}_2$.

Table 2 compares the classification performance of the methods. Our proposed method yields the lowest (or close to the lowest) error on most data sets. In particular, notice that both C4.5 and VDM perform poorly on the monks-1, in which the attributes are strongly correlated and the class labels often depend on the equality of two attributes.

Our method also allows easier interpretation of the relationship among different values of an nominal attribute. As an example, consider the mushroom data set, in which the task is to separate edible mushrooms from poisonous ones. Figure 1 shows the dissimilarity matrix for the attribute odor. A number of interesting observations can be made. For example, the odor pungent shows very large dissimilarities with the odors none, almond and anise. This corresponds well to our human perception that mushrooms with odors none, almond and anise are often non-poisonous, while pungent mushrooms are often poisonous. Similarly, the odor none is very dissimilar from odors pungent and creosote, which are often associated with poisonous mushrooms.

3.2. Experiments with Support Vector Machines

In this Section, we perform experiments on using the ADM on another type of classifier, namely the support vector machines (SVMs) [2]. We use the Gaussian kernel

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{1}{2v^2} d^2(\mathbf{x}_i, \mathbf{x}_j)),$$

where v^2 is the variance parameter (which is set to the variance of the whole data set). In general, learning can proceed by iterating SVM training and dissimilarity learning (of the $\mathbf{F}_a$'s). In this experiment, we simply perform one round of dissimilarity learning

using the Gaussian radial basis functions and then train the SVM. Table 3 compares our classification errors with that from using the overlap metric. Also shown in the table are kernel target alignments over the training sets [14]. The kernel target alignment measures the similarity between the kernel matrix and the class labels. Mathematically, for an $n \times n$ kernel matrix $\mathbf{K}$ and an n-dimensional vector of class labels $\mathbf{y} = (y_1, \ldots, y_n)'$ (n being the number of training patterns), the kernel target alignment is given by $\frac{\langle \mathbf{K}, \mathbf{y}\mathbf{y}' \rangle_F}{n\sqrt{\langle \mathbf{K}, \mathbf{K} \rangle_F}}$, where $\langle \cdot, \cdot \rangle$ denotes the Frobenius product[6]. In general, a high alignment implies good generalization performance of the resulting classifier. As can be seen from Table 3, our method again leads to smaller classification errors and higher alignments.

4. Conclusion

In this paper, we address the issue of measuring dissimilarity between two attribute values in a pattern recognition problem. We propose learning these dissimilarities directly by minimizing an error function on the training samples. Since the dissimilarities for all nominal attributes are learned together, any possible correlation among these attributes will be taken into account during the learning process. Experimental results on a number of synthetic and real-world data sets show the effectiveness of this approach.

Besides improving the classification performance on nominal data, the proposed approach allows meaningful interpretations of the relationships among the different values of a nominal attribute. These relationships inference process could also be valuable in data exploratory and data mining applications where interpretations and understanding of unknown nominal attributes are of significant importance.

REFERENCES

1. R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification.* Wiley, New York, 2nd edition, 2001.
2. V. Vapnik. *Statistical Learning Theory.* Wiley, New York, 1998.
3. T. Hastie and R. Tibshirani. Discriminant adaptive nearest neighbor classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(6):607–616, June 1996.
4. D. Wettschereck, D.W. Aha, and T. Mohri. A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. *Artificial Intelligence Review*, 11:273–314, 1997.
5. F. Esposito, D. Malerba, V. Tamma, and H.H. Bock. Classical resemblance measures. In H.-H. Bock and E. Diday, editors, *Analysis of Symbolic Data*, pages 139–152. Springer-Verlag, Berlin, 2000.
6. C. Stanfill and D. Waltz. Towards memory-based reasoning. *Communications of the ACM*, 29(12):1213–1228, November 1986.
7. J.C. Gower and P. Legendre. Metric and Euclidean properties of dissimilarity coefficients. *Journal of Classification*, 3:5–48, 1986.
8. H. Späth. *Cluster Analysis Algorithms for Data Reduction and Classification.* Ellis Horwood, Chichester, 1980.

[6]The Frobenius product between two matrices $M = [m_{ij}]$ and $N = [n_{ij}]$ is give n by $\langle M, N \rangle = \sum_{ij} m_{ij} n_{ij}$.

9. R.E. Bellman. *Adaptive Control Processes*. Princeton University Press, Princeton, NJ, 1961.
10. J.R. Quinlan. *C4.5 : Programs for Machine Learning*. Morgan Kaufmann, 1993.
11. C. Blake, E. Keogh, and C.J. Merz. UCI repository of machine learning databases, 1998. http://www.ics.uci.edu/∼mlearn/MLRepository.html University of California, Irvine, Department of Information and Computer Sciences.
12. S. Cost and S. Salzberg. A weighted nearest neighbor algorithm for learning with symbolic features. *Machine Learning*, 10:57–78, 1993.
13. R.H. Creecy, B.M. Masand, S.J. Smith, and D.L. Waltz. Trading MIPS and memory for knowledge engineering. *Communications of the ACM*, 35:48–64, 1992.
14. N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. Kandola. On kernel-target alignment. In T.G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA, 2002. MIT Press.
15. R. Kerber. ChiMerge: Discretization of numeric attributes. In *Proceedings of the International Conference on Machine Learning*, pages 123–127, 1992.
16. I.W. Tsang and J.T. Kwok. Distance metric learning with kernels. In *Proceedings of the International Conference on Artificial Neural Networks*, Istanbul, Turkey, June 2003.
17. D.R. Wilson and T.R. Martinez. Improved heterogeneous distance functions. *Journal of Artificial Intelligence Research*, 6:1–34, 1997.
18. E.P. Xing, A.Y. Ng, M.I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems 15*, Cambridge, MA, 2003. MIT Press.
19. Z. Zhang, J.T. Kwok, and D.-Y. Yeung. Parametric distance metric learning with label information. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, Acapulco, Mexico, August 2003.

Biography

Victor Cheng received the BEng in Electronic Engineering and MPhil degrees from the Hong Kong Polytechnic University in 1990 and 1993 respectively. Since 1993, he has been collaborated with the Hong Kong Polytechnic University in researches of pattern classifications. His current research interests include pattern classification, artificial neural networks, fuzzy logic and optimization.

Chun-hung Li received the BSc degree from the State University of New York at Stony and the PhD degree from the Hong Kong Polytechnic University. He is currently Assistant Professor in the Department of Computer Science in the Hong Kong Baptist University. His research interest is in the area of pattern recognition, machine learning and data mining.

James Tin-Yau Kwok received the Ph.D. degree in computer science from the Hong Kong University of Science and Technology in 1996. He then joined the Department of Computer Science, Hong Kong Baptist University as an Assistant Professor. He returned to the Hong Kong University of Science and Technology in 2000 and is now an Assistant Professor in the Department of Computer Science. His research interests include kernel methods, artificial neural networks, pattern recognition and machine learning.

Dr. C.K. Li received the BSc degree with first class honours from the University of Westminster, the MSc degree from the London University and PhD from the University of Westminster in 1976, 1978 and 1984 respectively. In 1982, he moved back to the University of Westminster as Senior Lecturer for two years before returning to Hong Kong in late 1984. In 1985 he was appointed Senior Lecturer in the Department of Electronic & Information Engineering of the Hong Kong Polytechnic University. He was subsequently promoted to Associate Professor. He has published over 150 technical papers and a number of patents. His research interests include adaptive systems, distributed/concurrent computing, signal processing and industrial electronics.

Table 1
Data sets used in the experiments.

data set	# nominal attributes	# continuous attributes	# training patterns	# testing patterns
credit	9	6	194	459
monks-1	6	0	124	432
monks-3	6	0	122	432
mushroom	22	0	282	5362
tic-tac-toe	9	0	190	768

Table 2
Classification errors on the data sets (Numbers in bold indicate the the lowest error obtained over the four methods).

data set	C4.5	RBF		
		overlap metric	VDM	ADM
credit	18.4%	16.7%	15.5%	**14.5%**
monks-1	23.4%	17.1%	16.3%	**0.0%**
monks-3	7.4%	9.3%	**2.9%**	3.1%
mushroom	**0.4%**	1.0%	0.7%	0.9%
tic-tac-toe	27.1%	17.8%	23.1%	**9.1%**

Table 3
Comparison between using the overlap metric and ADM in a SVM (Numbers in bold indicate the the better results).

data set	overlap metric		ADM	
	error	alignment	error	alignment
credit	15.0%	4.55	**14.6%**	**6.43**
monks-3	8.5%	5.12	**2.8%**	**24.36**
tic-tac-toe	23.3%	3.61	**10.2%**	**12.35**

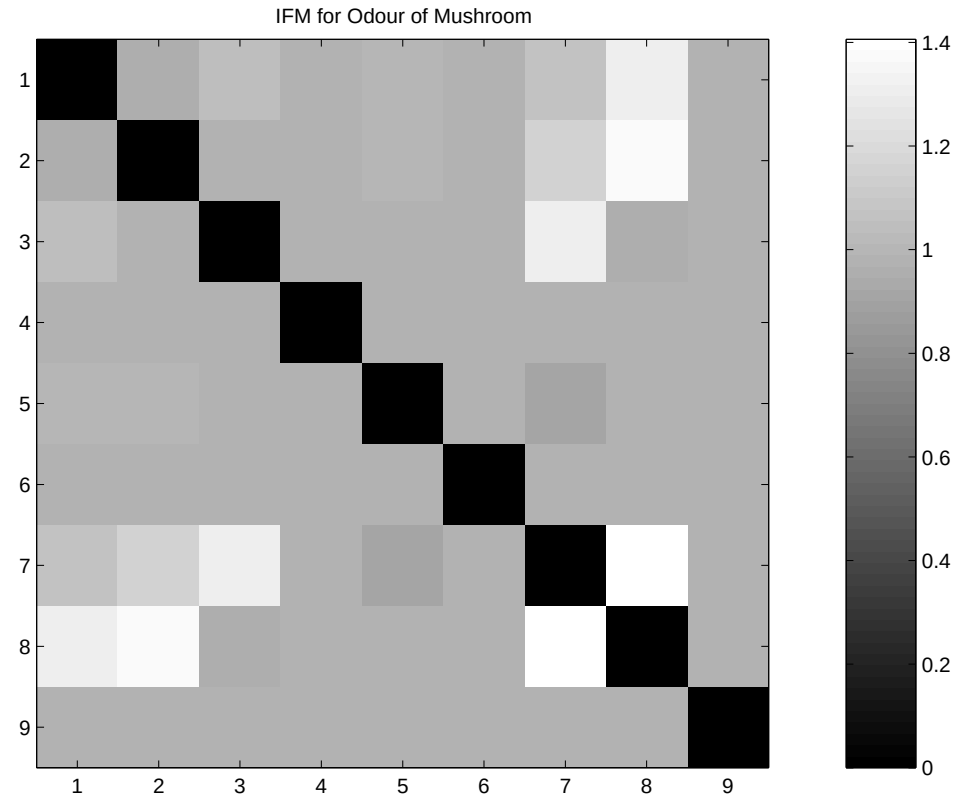

Figure 1. The dissimilarities matrix for attribute odor in the mushroom data set. Here, the indices correspond to: 1. almond, 2. anise, 3. creosote, 4. fishy, 5. foul, 6. musty, 7. none, 8. pungent, 9. spicy.