Group-Induced Vector Spaces

Manuele Bicego[1,⋆], Elżbieta Pękalska[2], and Robert P.W. Duin[3]

[1] DEIR, University of Sassari, Italy
bicego@uniss.it
[2] School of Computer Science, University of Manchester, UK
[3] Delft University of Technology, The Netherlands

Abstract. The strength of classifier combination lies either in a suitable averaging over multiple experts/sources or in a beneficial integration of complementary approaches. In this paper we focus on the latter and propose the use of group-induced vector spaces (GIVSs) as a way to combine unsupervised learning with classification. In such an integrated approach, the data is first modelled by a number of groups, found by a clustering procedure. Then, a proximity function is used to measure the (dis)similarity of an object to each group. A GIVS is defined by mapping an object to a vector of proximity scores, computed with respect to the given groups. In this study, we focus on a particular aspect of using GIVSs in a mode of building a trained combiner, namely the integration of generative and discriminative methods. First, in the generative step, we model the groups by simple generative models, building the GIVS space. The classification problem is then mapped in the resulting vector space, where a discriminative classifier is trained. Our experiments show that the integrated approach leads to comparable or better results than the generative methods in the original feature spaces.

1 Introduction

Practice in Multiple Classifier Systems as well as life experience show that a proper integration of complementary expertise leads to a better understanding of the problem and, usually, to better solutions. In this paper, we combine the complementary views of unsupervised and supervised learning. One possible approach is to discover the data structure and to apply different classifiers (or their combinations) depending on the position of objects in a vector space or groups they belongs to. Given a set of classifiers, this can be realized in local neighborhoods, e.g. by a dynamic classifier selection, as discussed in [15,2].

Here, we propose a simpler strategy that builds a group-induced vector space (GIVS) from the information of group structure. The main idea is to create such a representation space for the addressed problem such that it is successfully employed by discriminative approaches also for very small sample size problems or for non-vectorial data. Therefore, we characterize the problem in terms of (overlapping) groups determined by a clustering procedure. In principle, groups

[⋆] Corresponding author: via Torre Tonda, 34 - 07100 Sassari, Italy. Tel: +39 79 2017321 - Fax: +39 079 2017312.

M. Haindl, J. Kittler, and F. Roli (Eds.): MCS 2007, LNCS 4472, pp. 190–199, 2007.

can also be obtained by using label information. Nevertheless, the use of labels should be avoided, since this prevents the risk of overtraining (re-using the same information). Additionally, groups may also have different scales (both large and small groups are permitted) or be detected on different levels of a hierarchical clustering. Given such (possibly multiple-view) groups, a proximity function is needed that measures similarity of an object to each group. This is defined in agreement with the underlying grouping criterion or the property of the clustering technique, such as the Euclidean distance to the class centre if the K-means clustering is used or log-likelihood in case of the EM-clustering. In a GIVS, an object is mapped to a proximity vector, such that each proximity score reflects a similarity of an object to a group. The construction of this vector space is a fusion of weak proximity scores which encode in multiple views the grouping tendencies in the data. A statistical classifier trained in GIVS combines the weak clustering evidences towards a good solution. Note, however, that such a classifier should be simple in order to avoid overtraining, as the final result is a trained combiner [4]. A similar idea was also used for image classification in [9].

In general, our approach bears some resemblance to mixtures of local models. This includes local PCA models utilising either 'hard' [8] or 'soft' [6] assignments in in the partitioning phase, or probabilistic models based on local probabilistic PCA [14] or mixture of Gaussians [11]. All but first techniques couple both the partitioning and local model building into a EM approach. As a result, the model parameters and the mixing weights are optimized simultaneously. There are two main differences between such mixtures of local models and our approach. First, we derive a sequentially trained combiner which optimizes both unsupervised and supervised stages separetely. Secondly, the models are flexible: both local and global, possibly weak and overlapping and they may be derived by any clustering procedure, including these without the probabilistic character.

Another related approach is a network of locally tuned RBF units proposed in [12]. It first uses an unsupervised learning, such as K-means to determine cluster centers. These are then taken as RBF centers (of a hidden layer), whose widths are estimated by some nearest-neighbor heuristics. The output layer is a weighted linear combination of the RBFs. In the supervised setting, it is optimized by a gradient descent method. While it seems ad-hoc, its good performance may now be better understood in the light of our proposal, as explained below.

In this paper, we focus on a particular aspect of the proposed approach related to the integration of generative and discriminative methods, two complementary learning paradigms [7,13]. Generative methods model class probability density functions, while discriminative methods directly define the class boundaries. Generative techniques better characterize data, while discriminative techniques usually lead to a high performance. The combination of their strengths by the use of GIVSs seems to be a way for improvement. Here we study to what extent the simple generative modeling of groups in case of vectorial problems is beneficial for building GIVSs and training discriminative classifiers there.

The paper is organised as follows. Section 2 describes the proposed methodology, while Sections 3 and 4 explain the experimental set-up and analyze the results. The findings are summarized in Section 5.

2 Proposed Methodology

This section describes our integrated framework for combing grouping evidence with supervised learning. The starting point is a C-class classification problem, defined by a training set $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ with the associated labels $\{y_1, y_2, ..., y_N\}$, and a test set $\mathcal{Z} = \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_M\}$. The group-induced vector space (GIVS) classification methodology is defined via the following steps:

1. **Grouping**: choose or detect groups inside the training set. This can be achieved either by employing the label information (e.g. groups are chosen as the original classes) or not. Although the latter choice prevents possible overtraining from the repetitive usage of labels, both *Supervised grouping* and *Unsupervised grouping* have been used. In the latter case, the groups are determined by a clustering technique and are assumed to represent natural clusters inside the training set. Any clustering methodology can be used here, such as the simple K-means or the complex mode-seeking. Note that clusters may overlap, which means that examples belong to multiple clusters. In addition, we also define the *Fused Unsupervised Grouping* strategy, which collects sets of groups obtained by the Unsupervised Grouping for a growing number of clusters from 2 to F. The training set is therefore used multiple times, each time to find a particular number of groups.

   In general, the result of a supervised or unsupervised grouping process is a group structure $\mathcal{G}$ describing the training set with K groups, $G_1, G_2, \ldots, G_K$. Of course, $K = C$ in Supervised grouping, while $K = 2 + \ldots + (F-1) + F = \frac{(2+F)(F-1)}{2}$, in Fused Unsupervised Grouping.
2. **Group characterization:** in this step, a set of generative one-class models (such as Gaussian probability densities) are built based on the group structure $\mathcal{G}$ in order to model or describe the elements inside the groups. It is important to emphasize that each model is trained following a one-class paradigm, i.e. without any knowledge of the remaining training examples. As a result, a set of models $\{M_k\}$ describes the group structure $\mathcal{G}$. In our experimental study, we used very simple models, namely Gaussian probability density models with diagonal or spherical covariance matrices.
3. **Building Group-Induced Vector Spaces:** in this step the GIVS is constructed by representing each object by its distance or similarity to each group G_k. Formally, each object $\mathbf{x}_i$ is mapped to the Group-Induced Vector Space by the following function:

   $$givs_K(\mathbf{x}_i) \colon \mathbf{x}_i \longrightarrow [f(\mathbf{x}_i, M_1),\ f(\mathbf{x}_i, M_2),\ \ldots,\ f(\mathbf{x}_i, M_K)]^T \ , \qquad (1)$$

   where $f(\mathbf{x}_i, M_k)$ is a function measuring the relation between the vector $\mathbf{x}_i$ and the model M_k of the group G_k. For instance, this is the probability that $\mathbf{x}_i$ belongs to the model. In our experiments, we either used the Euclidean distance between $\mathbf{x}_i$ and the mean of G_k in case M_k is a spherical Gaussian model or the log-likelihood when M_k is the a diagonal Gaussian model. The training set $\mathcal{X}$ and the test set $\mathcal{Z}$ are then mapped to this new space with the $givs_K(\cdot)$ function. Depending on the grouping used, the resulting spaces are called *Supervised GIVS*, *Unsupervised GIVS* or *Fused Unsupervised GIVS*, while their dimensions equal to C, K and $\frac{(2+F)(F-1)}{2}$, correspondingly.

4. **Classification in the GIVS**: the classification problem is solved in the new feature space, in which the training set is

$$\mathcal{GIVS}(\mathcal{X}) = \{givs_K(\mathbf{x}_1), givs_K(\mathbf{x}_2), ..., givs_K(\mathbf{x}_N)\}$$

with the associated labels $\{y_1, y_2, ..., y_N\}$ and the test set is

$$\mathcal{GIVS}(\mathcal{Z}) = \{givs_K(\mathbf{z}_1), givs_K(\mathbf{z}_2), ..., givs_K(\mathbf{z}_M)\}.$$

Any vector-based classification strategy can be used in the GIVS, such as the KNN (k-nearest neighbor) or SVM (support vector machine).

An important feature of our approach is its applicability to problems in which a direct feature space cannot easily be extracted. Examples include problems dealing with sequences, strings, structures or graphs, i.e. problems in which a vector space is not directly obtainable, and discriminative approaches are not easily employable. In such cases, the usual option is to apply generative approaches. The generative models make use of the specific properties of the non-vectorial representations, but they loose at the same time as the discriminative approaches typically have a higher discrimination power. In this sense, the strategy proposed here is a method of combining generative and discriminative strategies, a very challenging research task [7,10]. Generative models are used to characterize groups, while the classification is performed in the corresponding GIVS by discriminative techniques.

3 Experimental Evaluation

This section presents our results. They are obtained by testing different variants of the combined generative-discriminative approach applied to several classification problems. In particular, the general scheme outlined in Section 2 has been instantiated by the following choices:

1. **Grouping:** in the supervised case, groups are defined by the given classes, hence their cardinality equals C, the number of classes. In the unsupervised cases (standard and fused), a traditional Gaussian Mixture Model (GMM) for clustering is adopted assuming diagonal covariance matrices. The number of clusters K varies from 2 to 15. Also F varies from 2 to 15, leading to fused vector spaces of the dimension in the range of 2 to $119 = 2 + \ldots + 14 + 15$.
2. **Group characterization:** two simple models are applied here: a Gaussian model with a diagonal covariance matrix and a spherical Gaussian model.
3. **Building Group-Induced Vector Spaces:** the proximity measure $f(\mathbf{x}_i, M_k)$ is defined differently for the two models:

   (a) Diagonal Covariance Gaussian:

$$f(\mathbf{x}_i, M_k) = \log \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \qquad (2)$$

i.e. the log-likelihood expressing the confidence that $\mathbf{x}_i$ belongs to the Gaussian model M_k defined by the mean vector $\boldsymbol{\mu}_k$ and the covariance matrix $\boldsymbol{\Sigma}_k$. In general, it is the negative square Mahalanobis distance, which now simplifies to the negative normalized (per feature) square Euclidean distance due to a diagonal covariance matrix. This choice is motivated by the traditional use of log-likelihood models in the literature. The logarithmic transformation is usually applied to probability estimates. It often simplifies the corresponding expression (when based on the exponent function) and, more importantly, emphasizes the differences in small probabilities, leading to better numerical accuracies.

(b) Spherical Gaussian:

$$f(\mathbf{x}_i, M_k) = ||\mathbf{x}_i - \boldsymbol{\mu}_k||_2, \tag{3}$$

i.e. the Euclidean distance of $\mathbf{x}_i$ to the mean of the Gaussian group M_k, which is a natural choice for spherical clusters.

4. **Classification in the GIVS**: here we choose two simple discriminative classifiers, the K-nearest Neighbor (KNN), with K optimized by the leave-one-out error on the training set, and the Logistic Linear classifier (LogLC) [3]. The idea is that we can reduce the complexity of the classifier while increasing the discrimination power and the complexity of the vector space.

Different versions of the proposed combining scheme are tested on several well-known data sets from the UCI Repository [5]. These are: *Banana*, *Ecoli*, *Liver*, *Diabetes*, *Breast* (Wisconsin Breast Cancer), *Glass*, *Wine* and *Ionosphere* data. The classification accuracy is computed by using the hold-out technique [3]. Here, the data set is randomly split into two equal and non-overlapping parts, one used for training and the other for testing. The training set is first normalized (to a unit variance) and then the KNN and LogLC are trained in supervised and unsupervised GIVSs. The classifiers are then tested on the normalized test set. This process is repeated 20 times and the results are averaged out. These average performances are shown in Table 1, for which the average standard deviations are less than 0.8%. This suggests that the proposed scheme is robust against data partitioning and initialisations of GMM. Concerning the (Fused) Unsupervised GIVS, only the best results in the testing set over the different values of K and F are shown. A brief discussion on how to choose these values is presented in the next section.

We compare our combined scheme to the corresponding generative classification methods; see Table 2. In case of the diagonal covariance Gaussian model, the standard maximum-a-posterior (MAP) approach was used, while for the spherical Gaussian model, the minimum distance approach was used. Since we deal with vectorial data sets, we also compare our approach to some standard discriminative classifiers trained in the original features spaces; see Table 2.

4 Analysis of the Results

Several observations can be made while studying the results from Tables 1 and 2:

Table 1. Average classification accuracies of the proposed generative-discriminative integration schemes on different data sets

Group Model	Diagonal Gaussian		Spherical Gaussian	
Measure	Log PDF		Euclidean distance	
Classifier	KNN	LogLC	KNN	LogLC
Banana, 2 classes (150-150), 2 features				
Supervised GIVS	86.07%	79.30%	91.92%	92.57%
Unsupervised GIVS	97.58%	84.10%	98.07%	97.12%
Fused Unsup. GIVS	97.72%	84.10%	98.08%	95.43%
Ecoli, 3 classes (143-77-52), 5 features				
Supervised GIVS	92.34%	93.36%	92.45%	92.55%
Unsupervised GIVS	90.15%	92.55%	92.81%	93.36%
Fused Unsup. GIVS	89.71%	91.02%	92.96%	92.37%
Liver, 2 classes (145-200), 6 features				
Supervised GIVS	55.78%	59.08%	56.79%	59.42%
Unsupervised GIVS	58.82%	70.12%	57.86%	65.03%
Fused Unsup. GIVS	59.51%	70.12%	57.17%	66.27%
Diabetes, 2 classes (500-268), 8 features				
Supervised GIVS	74.49%	75.59%	64.83%	67.98%
Unsupervised GIVS	67.94%	77.37%	73.16%	76.91%
Fused Unsup. GIVS	66.28%	77.43%	72.46%	76.64%
Breast, 2 classes (444-239), 9 features				
Supervised GIVS	95.42%	96.52%	70.13%	69.37%
Unsupervised GIVS	95.38%	96.52%	96.45%	96.71%
Fused Unsup. GIVS	95.28%	96.52%	96.48%	96.58%
Glass, 4 classes (70-76-17-51), 9 features				
Supervised GIVS	61.81%	61.62%	62.64%	62.64%
Unsupervised GIVS	60.42%	62.13%	68.98%	65.14%
Fused Unsup. GIVS	60.69%	61.11%	68.52%	64.91%
Wine, 3 classes (59-71-48), 13 features				
Supervised GIVS	92.33%	93.50%	70.00%	46.61%
Unsupervised GIVS	90.94%	92.78%	94.83%	95.39%
Fused Unsup. GIVS	90.83%	94.11%	94.94%	94.72%
Ionosphere, 2 classes (225-126), 32 features				
Supervised GIVS	88.81%	89.77%	38.78%	87.41%
Unsupervised GIVS	87.93%	90.43%	92.39%	92.64%
Fused Unsup. GIVS	87.53%	91.11%	92.67%	90.77%

1. Classifiers trained in the GIVS perform almost always evidently better than the corresponding generative approaches. There are two exceptions, the *Ecoli* and the *Wine* data, where there is no significant improvement. This can however be easily explained by the fact that in the original feature vector spaces the classes are well described by normal distributions. The original models are therefore well suited, hence well performing. The other examples indicate that our integrated method is able to recover from situations in which generative models are improper either due to wrong assumptions (such as independently distributed features or Gaussian models for non-Gaussian

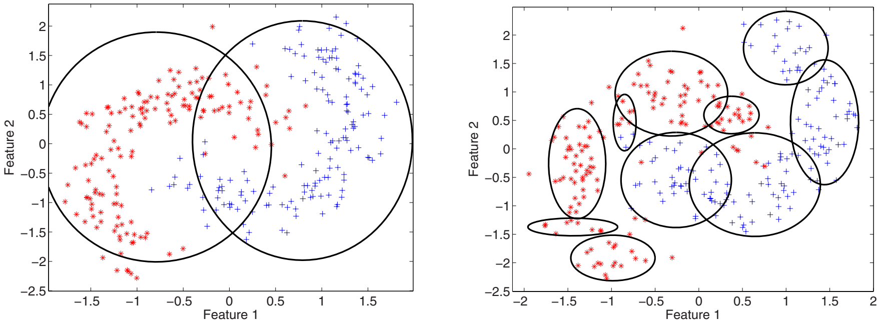
Table 2. Average classification accuracies of the standard discriminative and generative methods on different data sets

	Generative methods		*Discriminative methods*		
Data	Diag-Gauss (MAP)	Sph-Gauss (Min-dist)	KNN	LogLC	SVM
Banana	80.02%	92.05%	97.53%	85.68%	97.99%
Ecoli	92.45%	92.04%	94.22%	94.33%	94.99%
Liver	53.41%	59.28%	61.25%	68.02%	64.65%
Diabetes	75.22%	67.71%	74.19%	76.55%	77.01%
Breast	95.89%	60.37%	96.54%	96.52%	96.86%
Glass	47.41%	47.27%	69.06%	63.11%	67.36%
Wine	94.83%	95.67%	95.40%	96.93%	97.65%
Ionosphere	80.00%	89.01%	85.31%	75.97%	93.22%

classes), or due to estimation errors (e.g. for an unfavourable sample size or feature size). Our results show that in spite of a wrong model, discriminative classifiers built in the group-induced spaces lead to good results. In brief, our sequential generative-discriminative combination, being a trained combining classification scheme, can recover from initially unsuitable models.

2. When comparing the proposed integrated scheme to the discriminative approaches in the original feature spaces we can observe that they give almost comparable results, except for the *Liver* and *Ionosphere* data (except SVM). In these problems, the corresponding GIVSs are highly discriminative; the classifiers trained there outperform the classifiers in the original feature spaces. The *Liver* problem is very challenging and it seems that by using the clustering mechanism, the method is able to capture important groups in the original space to build a discriminative GIVS. With respect to the *Ionosphere* set, we should mention that this is a high-dimensional problem (32 dimensions), in which discriminative approaches could suffer from the curse of dimensionality (actually SVM, which is less sensitive to this problem, performs well on these data). By using the GIVS, we can reduce the dimension to a moderate size, significantly improving the results.
3. By analysing the GIVS approach in depth, we can observe that the Unsupervised GIVS almost always leads to better (or at least equal) results than the Supervised GIVS. If the classes cannot be characterized by normal distributions and we fit each class with a single Gaussian, then the resulting model is very poor. However, natural clusters can be discovered if we fit several ($> C$) Gaussian models to the complete data, neglecting the label information. The more-complex geometry of the classes can be revealed in this way. This fact is illustrated in Fig. 1. Different groups are shown in subplot (b). Some of them span both classes and capture the real geometry of the problem.

Concerning the Fused Unsupervised approach, there is no substantial improvement over the simple Unsupervised GIVS scheme. The logical explanation is that the dimension of the Fused GIVS is very high and the classifiers trained there suffer from the curse of dimensionality. Surely, a more clever fusion strategy is necessary, which is currently under investigation.



(a) Supervised GIVS. (b) Unsupervised GIVS with $K = 10$.

Fig. 1. Groups determined for the Banana data

4. With respect to the clustering technique, GMM seems to be a reasonable choice, since we use Gaussian models. Moreover, typically this technique is quite accurate and permits overlapping clusters. Some experiments with K-means and other methodologies were also performed, but the obtained results were comparable.
5. The two group models (Diagonal Covariance Gaussian and Spherical Gaussian) show different behaviours depending on the data characteristics. In general, the former leads to better results in low-dimensional feature spaces, while it is outperformed by the latter in high-dimensional spaces. See Table 1 to compare the results of the high-dimensional *Wine* and *Ionosphere* data with respect to the other ones. The Diagonal Covariance Gaussian models describe the groups in a more flexible way than the spherical Gaussian models are capable to, but they need sufficient data to determine their $2d$ parameters in the d-dimensional space. When the dimension of the feature space grows, simpler models are preferred to avoid bad estimates.

Finally, we emphasize that the used models should relatively be simple to prevent overtraining. In the current set-up we use the same training data twice: to build the GIVS and to train the classifier. So, we can only benefit from the sequential integration if the models are weak (such that we do not adjust to the data noise) and the final classifier is simple. The use of a complex model in the first stage can lead to overfitting of the complete classification strategy. To justify this in practice, we performed the same experiments with a model based on Parzen windows, hardly obtaining any improvements over the generative approach.

Number of groups for Unsupervised GIVS. To apply the Unsupervised (Fused) GIVS, we have to a priori set K (or F), the number of groups. Different values of K were evaluated in our experiments; we only present the overall best results. Since K is a free parameter, it should be chosen based on the training set only. Our experiments suggest, however, that the choice of perfect K is not crucial, providing that K is *sufficiently* large. In Fig. 2 we plot the average classification accuracy reached in the Unsupervised GIVS as a function of K (only the best classifier in the GIVS space is considered). The results are shown for the

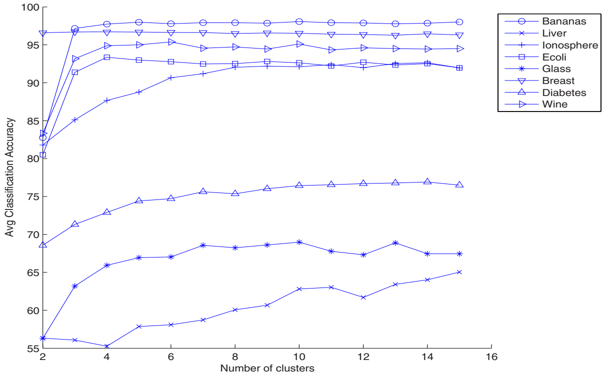


Fig. 2. Classification accuracy of the best discriminative method in the Unsupervised GIVS based on the spherical Gaussian model as a function of the number of clusters. The results are shown for several data sets.

non-Gaussian data sets and the spherical Gaussian model. We can observe that in almost all cases the performance increases with the growing number of clusters up to a certain number after which there is no further improvement. The value of K should also not be too large in order to prevent a decreasing performance due to the high dimension. Nevertheless, this peaking behavior was not present in the range we examined. Moreover, we could observe that for difficult problems (such as *Liver*, *Ionosphere* and *Diabetes*) the performance increase is slow, asking for a large number of clusters in order to reach a satisfactory accuracy. A possible solution to the direct computation of K is to determine the best value using the leave-one-out error on the training set (as typically done in several other classification contexts). Another approach is to link this value to the dimension of the problem, in terms of the cardinality of objects and classes, and the number of features.

5 Summary

In this paper, we propose a general strategy to integrate the strengths of unsupervised learning, which encodes data structure, and supervised learning. This is realized via group-induced vector spaces in which statistical classifiers are trained. In our experiments we deal with simple vectorial data and focus on combination of generative and discriminative approaches. Generative techniques (here, simple Gaussian models) are used to describe the data structure, while discriminative techniques (here, the KNN and logistic classifier) combine weak grouping evidences in a classification setting.

We find out that such an integrated generative-discriminative approach outperforms the generative techniques and leads to better results than the discriminative techniques in high-dimensional spaces (*Ionosphere* data) or in the case of highly-overlapping problems (*Liver* data). The stability of our scheme relies on the power of combining weak models: multiple (overlapping) clusters cover the data and their evidence is accumulated by a simple final combiner. In such a case, the discussed method will be robust against structure shifts in future data.

It is important to emphasize that the discussed approach is more general than the discriminative methods, applicable in vector spaces only. Now, we can also deal with non-vectorial structures, for which typically only generative models are fitted, as discriminative techniques are lacking. Since many powerful descriptive models are available (such as hidden Markov models), the advantage of the proposed integration lies in its wide applicability to almost any vectorial and non-vectorial classification problem. Future work will include the study of non-vectorial structures.

Finally, we also note that another possible employment of our approach is in the semi-supervised classification context [1]. Additional unlabeled data could efficiently be exploited in order to create an accurate and representative feature space, where a discriminant classifier may be trained using labels.

References

1. I. Cohen, F. Cozman, N. Sebe, M. Cirelo, and T. Huang. Semisupervised learning of classifiers: Theory, algorithms, and their application to human-computer interaction. *IEEE Trans. Pattern Analysis Machine Intell.*, 26(12):1553–1567, 2004.
2. L. Didaci, G. Giacinto, F. Roli, and G.L. Marcialis. A study on the performances of dynamic classifier selection based on local accuracy estimation. *Pattern Recognition*, 38(11):2188–2191, 2005.
3. R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley., 2001.
4. R.P.W. Duin. The combining classifier: To train or not to train? In *International Conference on Pattern Recognition*, volume II, pages 765–770, Canada, 2002.
5. S. Hettich, C.L. Blake, and C.J. Merz. UCI repository of ML databases, 1998. `http://www.ics.uci.edu/~mlearn/MLRepository.html`.
6. G. Hinton, M. Revow, and P. Dayan. Recognizing handwritten digits using mixtures of linear models. In *Neural Inf. Proc. Syst.*, pages 1015–1022, 1995.
7. T.S. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In *Neural Inf. Proc. Syst.*, 1999.
8. M. Kambhatla and T.K. Leen. Dimension reduction by local principal component analysis. *Neural Computation*, 9:443–482, 1997.
9. C. Lai, D.M.J. Tax, R.P.W. Duin, E. Pękalska, and P. Paclík. A study on combining image representations for image classification and retrieval. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(5):867–890, 2004.
10. M. Layton and M. Gales. Augmented statistical models: Exploiting generative models in discriminative classifiers. In *Neural Inf. Proc. Syst.*, 2005.
11. G. McLachlan and D. Peel. *Finite Mixture Models*. John Wiley & Sons, Inc., 2000.
12. J.E. Moody and C. Darken. Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1:281–294, 1989.
13. A. Ng and M. Jordan. On discriminative vs generative classifiers: A comparison of logistic regression and naive Bayes. In *Neural Inf. Proc. Syst.*, 2002.
14. M.E. Tipping and C. Bishop. Mixtures of probabilistic principal component analyzers. *Neural Computation*, 11:443–482, 1999.
15. K. Woods, W.P. Kegelmeyer, and K. Bower. Combination of multiple classifiers using local accuracy estimates. *IEEE Trans. on Pattern Recognition and Machine Intelligence*, 19(4):405–409, 1997.