Ensembles of Classifiers based on Dimensionality Reduction

Alon Schclar[a*] Lior Rokach[b] Amir Amit[c]

[a]School of Computer Science, Academic College of Tel Aviv-Yaffo
P.O.B 8401, Tel Aviv 61083, Israel
alonschc@mta.ac.il

[b]Department of Information Systems Engineering, Ben-Gurion University of the Negev
P.O.B 653, Beer-Sheva 84105, Israel
liorrk@bgu.ac.il

[c]The Efi Arazi School of Computer Science, Interdisciplinary Center (IDC) Herzliya
P.O.B 167, Herzliya 46150, Israel
ilhostc@ilhost.com

Abstract

We present a novel approach for the construction of ensemble classifiers based on dimensionality reduction. Dimensionality reduction methods represent datasets using a small number of attributes while preserving the information conveyed by the original dataset. The ensemble members are trained based on dimension-reduced versions of the training set. These versions are obtained by applying dimensionality reduction to the original training set using different values of the input parameters. This construction meets both the diversity and accuracy criteria which are required to construct an ensemble classifier where the former criterion is obtained by the various input parameter values and the latter is achieved due to the

*Corresponding author. Tel: +972-3-6803408; Fax: +972-3-6803342

decorrelation and noise reduction properties of dimensionality reduction. In order to classify a test sample, it is first embedded into the dimension reduced space of each individual classifier by using an out-of-sample extension algorithm. Each classifier is then applied to the embedded sample and the classification is obtained via a voting scheme. We present three variations of the proposed approach based on the Random Projections, the Diffusion Maps and the Random Subspaces dimensionality reduction algorithms. We also present a multi-strategy ensemble which combines AdaBoost and Diffusion Maps. A comparison is made with the Bagging, AdaBoost, Rotation Forest ensemble classifiers and also with the base classifier which does not incorporate dimensionality reduction. Our experiments used seventeen benchmark datasets from the UCI repository. The results obtained by the proposed algorithms were superior in many cases to other algorithms.

Keywords – Ensembles of classifiers; Dimensionality reduction; Out-of-sample extension; Random projections; Diffusion maps, Nyström extension

1 Introduction

Classifiers are predictive models which label data based on a training dataset T whose labels are known *a-priory*. A classifier is constructed by applying an induction algorithm, or inducer, to T - a process that is commonly known as *training*. Classifiers differ by the induction algorithms and training sets that are used for their construction. Common induction algorithms include nearest neighbors (NN), decision trees (CART [9], C4.5 [44]), Support Vector Machines (SVM) [61] and Artificial Neural Networks - to name a few. Since every inducer has its advantages and weaknesses, methodologies have been developed to enhance their performance. Ensemble classifiers are one of the most common ways to achieve that.

The need for dimensionality reduction techniques emerged in order to alleviate the so called *curse of dimensionality* [28]. In many cases, a high-dimensional dataset lies approximately on a low-dimensional manifold in the ambient space. Dimensionality reduction methods *embed* datasets into a low-dimensional space while preserving as much of the information conveyed by the dataset. The low-dimensional representation is referred to as the *embedding* of the dataset. Since the information is inherent in the geometrical structure of the dataset (e.g. clusters), a good embedding distorts the structure as little as possible while representing the dataset using a number of features that is substantially smaller than the dimension of the original ambient space. Furthermore, an effective dimensionality reduction algorithm also removes noisy features and inter-feature

correlations. Due to its properties, dimensionality reduction is a common step in many machine learning applications in fields such as signal processing [51, 2, 3] and image processing [37].

1.1 Ensembles of Classifiers

Ensembles of classifiers [32] mimic the human nature to seek advice from several people before making a decision where the underlying assumption is that combining the opinions will produce a decision that is better than each individual opinion. Several classifiers (ensemble *members*) are constructed and their outputs are combined - usually by voting or an averaged weighting scheme - to yield the final classification [43, 41]. In order for this approach to be effective, two criteria must be met: *accuracy* and *diversity* [32]. Accuracy requires each individual classifier to be as accurate as possible i.e. individually minimize the generalization error. Diversity requires to minimize the correlation among the generalization errors of the classifiers. These criteria are contradictory since optimal accuracy achieves a minimum and unique error which contradicts the requirement of diversity. Complete diversity, on the other hand, corresponds to random classification which usually achieves the worst accuracy. Consequently, individual classifiers that produce results which are moderately better than random classification are suitable as ensemble members. In [39], "kappa-error" diagrams are introduced to show the effect of diversity at the expense of reduced individual accuracy.

In this paper we focus on ensemble classifiers that use a single induction algorithm, for example the nearest neighbor inducer. This ensemble construction approach achieves its diversity by manipulating the training set. A well known way to achieve diversity is by bootstrap aggregation (*Bagging*) [8]. Several training sets are constructed by applying bootstrap sampling (each sample may be drawn more than once) to the original training set. Each training set is used to construct a different classifier where the repetitions fortify different training instances. This method is simple yet effective and has been successfully applied to a variety of problems such as spam detection [64], analysis of gene expressions [60] and image retrieval [58].

The award winning Adaptive Boosting (*AdaBoost*) [20] algorithm and its subsequent versions e.g. [18] and [56] provide a different approach for the construction of ensemble classifiers based on a single induction algorithm. This approach iteratively assigns weights to each training sample where the weights of the samples that are misclassified are increased according to a global error coefficient. The final classification combines the logarithm of the weights to yield the ensemble's classification.

Rotation Forest [45] is one of the current state-of-the-art ensemble classifiers. This method constructs different versions of the training set by employing the following steps: First, the feature set is divided into disjoint sets on which the original training set is projected. Next, a random sample of classes is eliminated and a bootstrap sample is selected from every projection result. Principal Component Analysis [27] (see Section 1.2) is then used to rotate each obtained subsample. Finally, the principal components are rearranged to form the dataset that is used to train a single ensemble member. The first two steps provide the required diversity of the constructed ensemble.

Multi-strategy ensemble classifiers [47] aim at combining the advantages of several ensemble algorithms while alleviating their disadvantages. This is achieved by applying an ensemble algorithm to the results produced by another ensemble algorithm. Examples of this approach include multi-training SVM (MTSVM) [35], MultiBoosting [62] and its extension using stochastic attribute selection [63].

Successful applications of the ensemble methodology can be found in many fields, for example, recommender systems [53], finance [34], manufacturing [46] and medicine [38].

1.2 Dimensionality reduction

The theoretical foundations for dimensionality reduction were established by Johnson and Lindenstrauss [29] who proved its feasibility. Specifically, they showed that N points in an N dimensional space can almost always be projected onto a space of dimension $C \log N$ with control over the ratio of distances and the error (distortion). Bourgain [7] showed that any metric space with N points can be embedded by a bi-Lipschitz map into an Euclidean space of $\log N$ dimension with a bi-Lipschitz constant of $\log N$. Various randomized versions of these theorems were successfully applied to protein mapping [36], reconstruction of frequency sparse signals [10, 16], textual and visual information retrieval [6] and clustering [19].

The dimensionality reduction problem can be formally described as follows. Let

$$\Gamma = \{x_i\}_{i=1}^{N} \tag{1}$$

be the original high-dimensional dataset given as a set of column vectors where $x_i \in \mathbb{R}^n$, n is the dimension of the ambient space and N is the size of the dataset. All dimensionality reduction methods embed the vectors into a lower dimensional space $\mathbb{R}^q$ where $q \ll n$. Their output is a set of column vectors in

the lower dimensional space

$$\widetilde{\Gamma} = \{\widetilde{x}_i\}_{i=1}^{N}\,,\ \widetilde{x}_i \in \mathbb{R}^q \tag{2}$$

where q is chosen such that it approximates the intrinsic dimensionality of Γ [25, 24]. We refer to the vectors in the set $\widetilde{\Gamma}$ as the *embedding vectors*.

Dimensionality reduction techniques employ two approaches: feature selection and feature extraction. Feature selection methods reduce the dimensionality by choosing q features from the feature vectors according to given criteria. The same features are chosen from all vectors. Current state-of-the-art feature selection methods include, for example, Manhattan non-negative matrix factorization [21], manifold elastic net [67] and geometric mean for subspace selection [57]. Feature extraction methods, on the other hand, derive features which are functions of the original features.

Dimensionality techniques can also be divided into *global* and *local* methods. The former derive embeddings in which *all* points satisfy a given criterion. Examples for global methods include:

- Principal Component Analysis (PCA) [27] which finds a low-dimensional embedding of the data points that best preserves their variance as measured in the ambient (high-dimensional) space;

- Kernel PCA (KPCA) [54] which is a generalization of PCA that is able to preserve non-linear structures. This ability relies on the *kernel trick* i.e. any algorithm whose description involves only dot products and does not require explicit usage of the variables can be extended to a non-linear version by using Mercer kernels [55]. When this principle is applied to dimensionality reduction it means that non-linear structures correspond to linear structures in some high-dimensional space. These structures can be detected by linear methods using kernels.

- Multidimensional scaling (MDS) [30, 14] algorithms which find an embedding that best preserves the inter-point distances among the vectors according to a given metric. This is achieved by minimizing a loss/cost *stress function* that measures the error between the pairwise distances of the embedding and their corresponding distances in the original dataset.

- ISOMAP [59] which applies MDS using the *geodesic distance* metric. The geodesic distance between a pair of points is defined as the length of the shortest path connecting these points that passes only through points in the dataset.

- Random projections [10, 16] in which every high-dimensional vector is projected onto a random matrix in order to obtain the embedding vector. This method is described in details in Section 4.

Contrary to global methods, local methods construct embeddings in which only *local* neighborhoods are required to meet a given criterion. The global description of the dataset is derived by the aggregation of the local neighborhoods. Common local methods include Local Linear Embedding (LLE) [49], Laplacian Eigenmaps [4], Hessian Eigenmaps [17] and Diffusion Maps [12, 50] which is used in this paper and is described in Section 3. The patch alignment framework [65] provides a unified framework to local dimensionality reduction techniques that employ two steps: (a) an optimization step where the local criterion is applied; and an alignment step in which the embedding is found. Examples that fit this framework include Local Linear Embedding (LLE) [49], Laplacian Eigenmaps [4], Hessian Eigenmaps [17], Local tangent space alignment [66] and Discriminative Locality Alignment (DLA) [65].

A key aspect of dimensionality reduction is how to efficiently embed a *new point* into a *given* dimension-reduced space. This is commonly referred to as *out-of-sample extension* where the sample stands for the original dataset whose dimensionality was reduced and does not include the new point. An accurate embedding of a new point requires the recalculation of the entire embedding. This is impractical in many cases, for example, when the time and space complexity that are required for the dimensionality reduction is quadratic (or higher) in the size of the dataset. An efficient out-of-sample extension algorithm embeds the new point without recalculating the entire embedding - usually at the expense of the embedding accuracy.

The Nyström extension [40] algorithm, which is used in this paper, embeds a new point in linear time using the quadrature rule when the dimensionality reduction involves eigen-decomposition of a kernel matrix. Algorithms such as Laplacian Eigenmaps, ISOMAP, LLE, and Diffusion Maps are examples that fall into this category and, thus, the embeddings that they produce can be extended using the Nyström extension [23, 5]. A formal description of the Nyström extension is given in the Sec. 3.2.

The main contribution of this paper is a novel framework for the construction of ensemble classifiers based on dimensionality reduction and out-of-sample extension. This approach achieves both the diversity and accuracy which are required for the construction of an effective ensemble classifier and it is general in the sense that it can be used with any inducer and any dimensionality reduction algorithm as long as it can be coupled with an out-of-sample extension method that suits it.

The rest of this paper is organized as follows. In Section 2 we describe the proposed approach. In Sections 3, 4 and 5 we introduce ensemble classifiers that are based on the Diffusion Maps, random projections and random subspaces dimensionality reduction algorithms, respectively. Experimental results are given in Section 6. We conclude and describe future work in Section 7.

2 Dimensionality reduction ensemble classifiers

The proposed approach achieves the diversity requirement of ensemble classifiers by applying a given dimensionality reduction algorithm to a given training set using different values for its input parameters. An input parameter that is common to all dimensionality reduction techniques is the dimension of the embedding space. In order to obtain sufficient diversity, the dimensionality reduction algorithm that is used should incorporate additional input parameters or, alternatively, incorporate a randomization step. For example, the Diffusion Maps [12] dimensionality algorithm uses an input parameter that defines the size of the local neighborhood of a point. Variations of this notion appear in other local dimensionality reduction methods such as LLE [49] and Laplacian Eigenmaps [4]. The Random Projections [16] (Section 4) and Random Subspaces [26, 48] (Section 5) methods, on the other hand, do not include input parameters other than the dimensionality of the embedding space. However, they incorporate a randomization step which diversifies the data (this approach already demonstrated good results using Random Projections in [52] and we extend them in this paper). In this sense, PCA is not suitable for the proposed framework since it does not include a randomization step and the only input parameter it has is the dimension of the embedding space (this parameter can also be set according to the total amount of variance of the original dataset that the embedding is required to maintain). Thus, PCA offers no way to diversify the data. On the other hand, dimensionality reduction algorithms that are suitable for the proposed method include ISOMAP [59], LLE [49], Hessian LLE [17], Local tangent space alignment [66] and Discriminative Locality Alignment (DLA) [65]. These methods are suitable since they require as input the number of nearest neighbors to determine the size of the local neighborhood of each data point. Laplacian Eigenmaps [4] and KPCA [54] are also suitable for the proposed framework as they include a continuous input variable to determine the radius of the local neighborhood of each point.

After the training sets are produced by the dimensionality reduction algorithms, each set is used to train a classifier to produce one of the ensemble members. The training process is illustrated in Fig. 1.

Employing dimensionality reduction to a training set has the following ad-

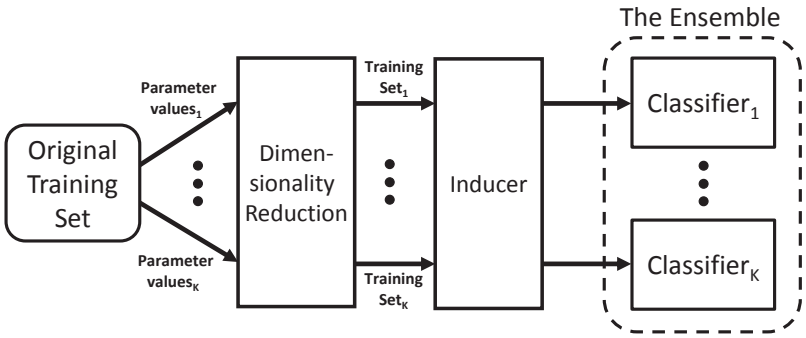

Figure 1: Ensemble training.

vantages:

- It reduces noise and decorrelates the data.
- It reduces the computational complexity of the classifier construction and consequently the complexity of the classification.
- It can alleviate over-fitting by constructing combinations of the variables [42].

These points meet the accuracy and diversity criteria which are required to construct an effective ensemble classifier and thus render dimensionality reduction a technique which is tailored for the construction of ensemble classifiers. Specifically, removing noise from the data contributes to the accuracy of the classifier while diversity is obtained by the various dimension-reduced versions of the data.

In order to classify test samples, they are first embedded into the low-dimensional space of each of the training sets using out-of-sample extension. Next, each ensemble member is applied to its corresponding embedded test sample and the produced results are processed by a voting scheme to derive the result of the ensemble classifier. Specifically, each classification is given as a vector containing the probabilities of each possible label. These vectors are aggregated and the ensemble classification is chosen as the label with the largest probability. Figure 2 depicts the classification process of a test sample.

3 Diffusion Maps

The Diffusion Maps (DM) [12] algorithm embeds data into a low-dimensional space where the geometry of the dataset is defined in terms of the connectivity between every pair of points in the ambient space. Namely, the similarity

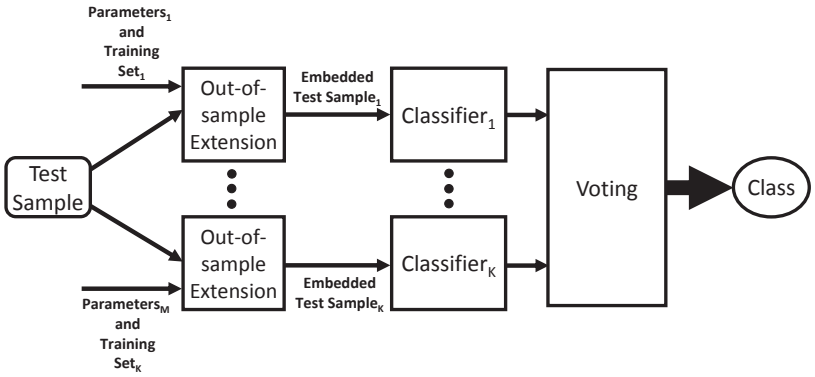

Figure 2: Classification process of a test sample.

between two points x and y is determined according to the number of paths connecting x and y via points in the dataset. This measure is robust to noise since it takes into account all the paths connecting x and y. The Euclidean distance between x and y in the dimension-reduced space approximates their connectivity in the ambient space.

Formally, let Γ be a set of points in $\mathbb{R}^n$ as defined in Eq. 1. A weighted undirected graph $G(V,E), |V| = N, |E| \ll N^2$ is constructed, where each vertex $v \in V$ corresponds to a point in Γ. The weights of the edges are chosen according to a weight function $w_\varepsilon(x,y)$ which measures the similarities between every pair of points where the parameter ε defines a local neighborhood for each point. The weight function is defined by a kernel function obeying the following properties:

symmetry: $\forall x_i, x_j \in \Gamma,\ w_\varepsilon(x_i,x_j) = w_\varepsilon(x_j,x_i)$

non-negativity: $\forall x_i, x_j \in \Gamma,\ w_\varepsilon(x_i,x_j) \geq 0$

positive semi-definite: for every real-valued bounded function f defined on Γ, $\sum_{x_i,x_j\in\Gamma} w_\varepsilon(x_i,x_j) f(x_i) f(x_j) \geq 0$.

fast decay: $w_\varepsilon(x_i,x_j) \to 0$ when $\|x_i - x_j\| \gg \varepsilon$ and $w_\varepsilon(x_i,x_j) \to 1$ when $\|x_i - x_j\| \ll \varepsilon$. This property facilitates the representation of w_ε by a sparse matrix.

A common choice that meets these criteria is the Gaussian kernel:

$$w_\varepsilon(x_i,x_j) = e^{-\frac{\|x_i - x_j\|^2}{2\varepsilon}}.$$

A weight matrix w_ε is used to represent the weights of the edges. Given a graph G, the Graph Laplacian normalization [11] is applied to the weight matrix

w_ε and the result is given by M:

$$M_{i,j} \triangleq m(x,y) = \frac{w_\varepsilon(x,y)}{d(x)}$$

where $d(x) = \sum_{y\in\Gamma} w_\varepsilon(x,y)$ is the degree of x. This transforms w_ε into a Markov transition matrix corresponding to a random walk through the points in Γ. The probability to move from x to y in *one* time step is denoted by $m(x,y)$. These probabilities measure the connectivity of the points within the graph.

The transition matrix M is conjugate to a symmetric matrix A whose elements are given by $A_{i,j} \triangleq a(x,y) = \sqrt{d(x)}m(x,y)\frac{1}{\sqrt{d(y)}}$. Using matrix notation, A is given by $A = D^{\frac{1}{2}}MD^{-\frac{1}{2}}$, where D is a diagonal matrix whose values are given by $d(x)$. The matrix A has n real eigenvalues $\{\lambda_l\}_{l=0}^{n-1}$ where $0 \leq \lambda_l \leq 1$, and a set of orthonormal eigenvectors $\{v_l\}_{l=1}^{N-1}$ in $\mathbb{R}^n$. Thus, A has the following spectral decomposition:

$$a(x,y) = \sum_{k\geq 0} \lambda_k v_l(x) v_l(y). \tag{3}$$

Since M is conjugate to A, the eigenvalues of both matrices are identical. In addition, if $\{\phi_l\}$ and $\{\psi_l\}$ are the left and right eigenvectors of M, respectively, then the following equalities hold:

$$\phi_l = D^{\frac{1}{2}}v_l, \quad \psi_l = D^{-\frac{1}{2}}v_l. \tag{4}$$

From the orthonormality of $\{v_i\}$ and Eq. 4 it follows that $\{\phi_l\}$ and $\{\psi_l\}$ are bi-orthonormal i.e. $\langle\phi_m, \psi_l\rangle = \delta_{ml}$ where $\delta_{ml} = 1$ when $m = l$ and $\delta_{ml} = 0$, otherwise. Combing Eqs. 3 and 4 together with the bi-orthogonality of $\{\phi_l\}$ and $\{\psi_l\}$ leads to the following eigen-decomposition of the transition matrix M

$$m(x,y) = \sum_{l\geq 0} \lambda_l \psi_l(x) \phi_l(y). \tag{5}$$

When the spectrum decays rapidly (provided ε is appropriately chosen - see Sec. 3.1), only a few terms are required to achieve a given accuracy in the sum. Namely,

$$m(x,y) \backsimeq \sum_{l=0}^{n(p)} \lambda_l \psi_l(x) \phi_l(y)$$

where $n(p)$ is the number of terms which are required to achieve a given precision p.

We recall the *diffusion distance* between two data points x and y as it was

defined in [12]:

$$D^2(x,y) = \sum_{z\in\Gamma} \frac{(m(x,z) - m(z,y))^2}{\phi_0(z)}. \tag{6}$$

This distance reflects the geometry of the dataset and it depends on the number of paths connecting x and y. Substituting Eq. 5 in Eq. 6 together with the bi-orthogonality property allows to express the diffusion distance using the right eigenvectors of the transition matrix M:

$$D^2(x,y) = \sum_{l\geq 1} \lambda_l^2 (\psi_l(x) - \psi_l(y))^2. \tag{7}$$

Thus, the family of Diffusion Maps $\{\Psi(x)\}$ which is defined by

$$\Psi(x) = (\lambda_1\psi_1(x), \lambda_2\psi_2(x), \lambda_3\psi_3(x), \cdots) \tag{8}$$

embeds the dataset into a Euclidean space. In the new coordinates of Eq. 8, the *Euclidean* distance between two points in the embedding space is equal to the *diffusion* distance between their corresponding two high dimensional points as defined by the random walk. Moreover, this facilitates the embedding of the original points into a low-dimensional Euclidean space $\mathbb{R}^q$ by:

$$\Xi_t : x_i \to \left(\lambda_2^t\psi_2(x_i), \lambda_3^t\psi_3(x_i), \ldots, \lambda_{q+1}^t\psi_{q+1}(x_i)\right). \tag{9}$$

which also endows coordinates on the set Γ. Since $\lambda_1 = 1$ and $\psi_1(x)$ is constant, the embedding uses $\lambda_2, \ldots, \lambda_{q+1}$. Essentially, $q \ll n$ due to the fast decay of the eigenvalues of M. Furthermore, q depends only on the dimensionality of the data as captured by the random walk and not on the original dimensionality of the data. Diffusion maps have been successfully applied for acoustic detection of moving vehicles [51] and fusion of data and multicue data matching [33].

3.1 Choosing ε

The choice of ε is critical to achieve the optimal performance by the DM algorithm since it defines the size of the local neighborhood of each point. On one hand, a large ε produces a coarse analysis of the data as the neighborhood of each point will contain a large number of points. In this case, the diffusion distance will be close to 1 for most pairs of points. On the other hand, a small ε might produce many neighborhoods that contain only a single point. In this case, the diffusion distance is zero for most pairs of points. The best choice lies between these two extremes. Accordingly, the ensemble classifier which is based on the the Diffusion Maps algorithm will construct different versions of

the training set using different values of ε which will be chosen between the shortest and longest pairwise distances.

3.2 The Nyström out-of-sample extension

The Nyström extension [40] is an extrapolation method that facilitates the extension of any function $f : \Gamma \to \mathbb{R}$ to a set of new points which are added to Γ. Such extensions are required in on-line processes in which new samples arrive and a function f that is defined on Γ needs to be extrapolated to include the new points. These settings exactly fit the settings of the proposed approach since the test samples are given *after* the dimensionality of the training set was reduced. Specifically, the Nyström extension is used to embed a new point into the reduced-dimension space where every coordinate of the low-dimensional embedding constitutes a function that needs to be extended.

We describe the Nyström extension scheme for the Gaussian kernel that is used by the Diffusion Maps algorithm. Let Γ be a set of points in $\mathbb{R}^n$ and Ψ be its embedding (Eq. 8). Let $\bar{\Gamma}$ be a set in $\mathbb{R}^n$ such that $\Gamma \subset \bar{\Gamma}$. The Nyström extension scheme extends Ψ onto the dataset $\bar{\Gamma}$. Recall that the eigenvectors and eigenvalues form the dimension-reduced coordinates of Γ (Eq. 9). The eigenvectors and eigenvalues of a Gaussian kernel with width ε which is used to measure the pairwise similarities in the training set Γ are computed according to

$$\lambda_l \varphi_l(x) = \sum_{y \in \Gamma} e^{-\frac{\|x-y\|^2}{2\varepsilon}} \varphi_l(y), \ x \in \Gamma. \tag{10}$$

If $\lambda_l \neq 0$ for every l, the eigenvectors in Eq. 10 can be extended to any $x \in \mathbb{R}^n$ by

$$\bar{\varphi}_l(x) = \frac{1}{\lambda_l} \sum_{y \in \Gamma} e^{-\frac{\|x-y\|^2}{2\varepsilon}} \varphi_l(y), \ x \in \mathbb{R}^n. \tag{11}$$

Let f be a function on the training set Γ and let $x \notin \Gamma$ be a new point. In the Diffusion Maps setting, we are interested in approximating

$$\Psi(x) = (\lambda_2 \psi_2(x), \lambda_3 \psi_3(x), \cdots, \lambda_{q+1} \psi_{q+1}(x)).$$

The eigenfunctions $\{\varphi_l\}$ are the outcome of the spectral decomposition of a symmetric positive matrix. Thus, they form an orthonormal basis in $\mathbb{R}^N$ where N is the number of points in Γ. Consequently, any function f can be written as a linear combination of this basis:

$$f(x) = \sum_l \langle \varphi_l, f \rangle \varphi_l(x), \ x \in \Gamma.$$

Using the Nyström extension, as given in Eq. 11, f can be defined for any point in $\mathbb{R}^n$ by

$$\bar{f}(x) = \sum_l \langle \varphi_l, f \rangle \bar{\varphi}_l(x), \ x \in \mathbb{R}^n. \tag{12}$$

The above extension facilitates the decomposition of every diffusion coordinate ψ_i as $\psi_i(x) = \sum_l \langle \varphi_l, \psi_i \rangle \varphi_l(x), \ x \in \Gamma$. In addition, the embedding of a new point $\bar{x} \in \bar{\Gamma} \backslash \Gamma$ can be evaluated in the embedding coordinate system by $\bar{\psi}_i(\bar{x}) = \sum_l \langle \varphi_l, \psi_i \rangle \bar{\varphi}_l(\bar{x})$.

Note that the scheme is ill conditioned since $\lambda_l \longrightarrow 0$ as $l \longrightarrow \infty$. This can be solved by cutting-off the sum in Eq. 12 and keeping only the eigenvalues (and their corresponding eigenfunctions) that satisfy $\lambda_l \geq \delta \lambda_0$ (where $0 < \delta \leq 1$ and the eigenvalues are given in descending order of magnitude):

$$\bar{f}(x) = \sum_{\lambda_l \geq \delta \lambda_0} \langle \varphi_l, f \rangle \bar{\varphi}_l(x), \ x \in \mathbb{R}^n. \tag{13}$$

The result is an extension scheme with a condition number δ. In this new scheme, f and $\bar{f}$ do not coincide on Γ but they are relatively close. The value of ε controls this error. Thus, choosing ε carefully may improve the accuracy of the extension.

3.3 Ensemble via Diffusion maps

Let Γ be a training set as described in Eq. 1. Every dimension-reduced version of Γ is constructed by applying the Diffusion Maps algorithm to Γ where the parameter ε is randomly chosen from the set of all pairwise Euclidean distances between the points in Γ i.e. from $\{\| x - y \|\}_{x,y \in \Gamma}$. The dimension of the reduced space is fixed for all the ensemble members at a given percentage of the ambient space dimension. We denote by $\widetilde{\Gamma}(\varepsilon_i) \subseteq \mathbb{R}^q$ the training set that is obtained from the application of the diffusion maps algorithm to Γ using the randomly chosen value ε_i where $i = 1, \ldots, K$ and K is the number of ensemble members. The ensemble members are constructed by applying a given induction algorithm to each training set $\widetilde{\Gamma}(\varepsilon_i)$. In order to classify a new sample, it is first embedded into the dimension-reduced space $\mathbb{R}^q$ of each classifier using the Nyström extension (Section 3.2). Then, every ensemble member classifies the new sample and the voting scheme which is described in Section 2 is used to produce the ensemble classification. Note that in order for the Nyström extension to work, each ensemble member must store the eigenvectors and eigenvalues which were produced by the Diffusion Maps algorithm.

4 Random Projections

The Random projections algorithm implements the Johnson and Lindenstrauss lemma [29] (see Section 1.2). In order to reduce the dimensionality of a given training set Γ, a set of random vectors $\Upsilon = \{\rho_i\}_{i=1}^{n}$ is generated where $\rho_i \in \mathbb{R}^q$ are column vectors and $\|\rho_i\|_{l_2} = 1$. Two common ways to choose the entries of the vectors $\{\rho_i\}_{i=1}^{n}$ are:

1. From a uniform (or normal) distribution over the q dimensional unit sphere.

2. From a Bernoulli +1/-1 distribution. In this case, the vectors are normalized so that $\|\rho_i\|_{l_2} = 1$ for $i = 1, \ldots, n$.

Next, the vectors in Υ are used to form the columns of a $q \times n$ matrix

$$R = (\rho_1 | \rho_2 | \ldots | \rho_n)\,. \tag{14}$$

The embedding $\widetilde{x}_i$ of x_i is obtained by

$$\widetilde{x}_i = R \cdot x_i$$

Random projections are well suited for the construction of ensembles of classifiers since the randomization meets the diversity criterion (Section 1.1) while the bounded distortion rate provides the accuracy.

Random projections have been successfully employed for dimensionality reduction in [19] as part of an ensemble algorithm for clustering. An Expectation Maximization (of Gaussian mixtures) clustering algorithm was applied to the dimension-reduced data. The ensemble algorithm achieved results that were superior to those obtained by: (a) a single run of random projection/clustering; and (b) a similar scheme which used PCA to reduce the dimensionality of the data.

4.1 Out-of-sample extension

In order to embed a new sample y into the dimension-reduced space $\mathbb{R}^q$ of the i-th ensemble member, the sample is simply projected onto the random matrix R that was used to reduce the dimensionality of the member's training set. The embedding of y is given by $\tilde{y} = R \cdot y$. Accordingly, each random matrix needs to be stored as part of its corresponding ensemble member in order to allow out-of-sample extension.

4.2 Ensemble via Random Projections

In order to construct the dimension-reduced versions of the training set, K random matrices $\{R_i\}_{i=1}^{K}$ are constructed (recall that K is the number of ensemble members). The training set is projected onto each random matrix R_i and the dataset which is produced by each projection is denoted by $\Gamma(R_i)$. The ensemble members are constructed by applying a given inducer to each of the dimension-reduced datasets in $\{\Gamma(R_i)\}_{i=1}^{K}$.

A new sample is classified by first embedding it into the dimension-reduced space $\mathbb{R}^q$ of every classifier using the scheme in Section 4.1. Then, each ensemble member classifies the new sample and the voting scheme from Section 2 is used to determine the classification by the ensemble.

5 Random Subspaces

The Random subspaces algorithm reduces the dimensionality of a given training set Γ by projecting the vectors onto a random subset of attributes. Formally, let $\{i_k\}_{k=1}^{q}$ be a randomly chosen subset of attributes. The embedding $\widetilde{x}$ of $x = (x_1, \ldots, x_n)$ is obtained by $\widetilde{x} = \left(x_{i_1}, \ldots, x_{i_q}\right)$. Accordingly, each random set of attributes needs to be stored as part of its corresponding ensemble member.

This method is a special case of the random projections dimensionality reduction algorithm described in Sec. 4 where the rows (and column) of the matrix R in eq. 14 are unique indicator vectors.

Random subspaces have been used to construct decision forests [26] - an ensemble of tree classifiers - and also to construct ensemble *regressors* [48]. Ensemble regressors employ a multivariate function instead of a voting scheme to combine the individual results of the ensemble members. The training sets that are constructed by the Random subspaces method are dimension-reduced versions of the original dataset and therefore this method is investigated in our experiments. This method combined with support vector machines has been successfully applied to relevance feedback in image retrieval [58].

5.1 Out-of-sample extension

In order to embed a new sample y into the dimension-reduced space $\mathbb{R}^q$ of the i-th ensemble member, the sample is simply projected onto $\{i_k\}_{k=1}^{q}$ - the member's subset of attributes. The embedding of $y = (y, \ldots, y_n)$ is given by $\tilde{y} = \left(y_{i_1}, \ldots, y_{i_q}\right)$.

Table 1: Properties of the benchmark datasets used for the evaluation.

Dataset Name	Instances	Features	Labels
Musk1	476	166	2
Musk2	6598	166	2
Pima-diabetes	768	8	2
Ecoli	335	7	8
Glass	214	9	7
Hill Valley with noise	1212	100	2
Hill Valley without noise	1212	100	2
Ionosphere	351	34	2
Iris	150	4	3
Isolet	7797	617	26
Letter	20000	16	26
Madelon	2000	500	2
Multiple features	2000	649	10
Sat	6435	36	7
Waveform with noise	5000	40	3
Waveform without noise	5000	21	3
Yeast	1484	8	10

5.2 Ensemble via Random Subspaces

In order to construct the dimension-reduced versions of the training set, K subsets of features are randomly chosen. The training set is projected onto each attribute subset and the ensemble members are constructed by applying a given inducer to each of the dimension-reduced datasets.

A new sample is classified by first embedding it into the dimension-reduced space $\mathbb{R}^q$ of every classifier using the scheme in Section 5.1. Then, each ensemble member classifies the new sample and the voting scheme from Section 2 is used to determine the ensemble's classification.

6 Experimental results

In order to evaluate the proposed approach, we used the WEKA framework [22]. We tested our approach on 17 datasets from the UCI repository [1] which contains benchmark datasets that are commonly used to evaluate machine learning algorithms. The list of datasets and their properties are summarized in Table 1.

6.1 Experiment configuration

In order to reduce the dimensionality of a given training set, one of two schemes was employed depending on the dimensionality reduction algorithm at hand.

Algorithm 1 Steps for constructing the training set of a single ensemble member using the Diffusion Maps algorithm.

Input: Dataset Γ, target dimension q

Output: A dimension reduced training set $\widetilde{\widetilde{\Gamma}}$.

1. Select a random value $\varepsilon \in \{\| x - y \|\}_{x,y \in \Gamma}$
2. Select a random sample $\bar{\Gamma}$ of 600 unique elements from Γ.
3. Apply the Diffusion Maps algorithm to $\bar{\Gamma}$ resulting in $\widetilde{\Gamma}$
4. Extend $\widetilde{\Gamma}$ to include the points in $\Gamma \backslash \bar{\Gamma}$ using the Nyström extension - resulting in $\widetilde{\widetilde{\Gamma}}$.

The first scheme was used for the Random Projection and the Random Subspaces algorithms and it applied the dimensionality reduction algorithm to the dataset without any pre-processing of the dataset. However, due to the space and time complexity of the Diffusion Maps algorithm, which is quadratic in the size of the dataset, a different scheme was used. First, a random value $\varepsilon \in \{\| x - y \|\}_{x,y \in \Gamma}$ was selected. Next, a random sample of 600 unique data items was drawn (this size was set according to time and memory limitations). The Diffusion Maps algorithm was then applied to the sample which produced a dimension-reduced training set. This set was then extended using the Nyström extension to include the training samples which were not part of the sample. These steps are summarized in Algorithm 1.

All ensemble algorithms were tested using the following inducers: (a) nearest-neighbors (WEKA's B1 inducer); (b) decision tree (WEKA's J48 inducer); and (c) Naïve Bayes. The ensembles were composed of ten classifiers (the information theoretic problem of choosing the optimal size of an ensemble is out of the scope of this paper. This problem is discussed, for example, in [31]). The dimension-reduced space was set to half of the original dimension of the data. Ten-fold cross validation was used to evaluate each ensemble's performance on each of the datasets.

The constructed ensemble classifiers were compared with: a non-ensemble classifier which applied the induction algorithm to the dataset without dimensionality reduction (we refer to this classifier as the *plain* classifier). The constructed ensemble classifiers were also compared with the Bagging [8], AdaBoost [20] and Rotation Forest [45] ensemble algorithms. In order to see whether the Diffusion Maps ensemble classifier can be further improved as part of a multi-strategy ensemble (Section 1.1), we constructed an ensemble classifier whose members applied the AdaBoost algorithm to their Diffusion Maps dimension-reduced training sets.

We used the default values of the parameters of the WEKA built-in ensemble classifiers in all the experiments. For the sake of simplicity, in the following we refer to the ensemble classifiers which use the Diffusion Maps and Random Projections dimensionality algorithms as the DME and RPE classifiers, respectively. The ensemble classifier which is based on the random subspaces dimensionality reduction algorithm is referred to as the RSE classifier.

6.2 Results

Tables 2, 3, and 4 describe the results obtained by the decision tree, nearest-neighbor and Naïve Bayes inducers, respectively. In each of the tables, the first column specifies the name of the tested dataset and the second column contains the results of the plain classifier. The second to last row contains the average improvement percentage of each algorithm compared to the plain classifier. We calculate the average rank of each inducer across all datasets in the following manner: for each of the datasets, the algorithms are ranked according to the accuracy that they achieved. The average rank of a given inducer is obtained by averaging its obtained ranks over all the datasets. The average rank is given in the last row of each table.

The results of the experimental study indicate that dimensionality reduction is a promising approach for the construction of ensembles of classifiers. In 113 out of 204 cases the dimensionality reduction ensembles outperformed the plain algorithm with the following distribution: RPE (33 cases out of 113), DM+AdaBoost (30 cases), RSE (27 cases) and DM (23 cases).

Ranking all the algorithms according to the average accuracy improvement percentage produces the following order: Rotation Forest (6.4%), Random projection (4%), DM+AdaBoost (2.1%), Bagging (1.5%), AdaBoost (1%), DM (0.7%) and Random subspaces (-6.7%). Note that the RSE algorithm achieved an average decrease of 6.7% in accuracy. A closer look reveals that this was caused by a particularly bad performance when the Naïve Bayes inducer was used (26% average decrease in accuracy). In contrast, improvement averages of 1.7% and 4.4% were achieved when the RSE algorithm used the nearest-neighbors and J48 inducers, respectively. This may be due to datasets whose features are not independent - a situation which does not conform with the basic assumption of the Naïve Bayes inducer. For example, the *Isolet* dataset is composed of acoustic recordings that are decomposed to *overlapping* segments where features of each segment constitute an instance in the dataset. In these settings, the features are not independent. Since the other algorithms, including the plain one, achieve much better results when applied to this dataset, we can assume that because the RSE algorithm chooses a random subset of features,

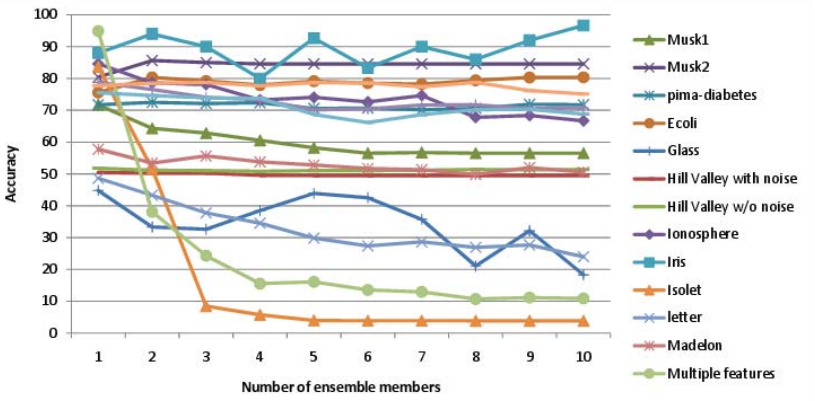

Figure 3: Accuracy of the RSE algorithm using the Naïve Bayes inducer.

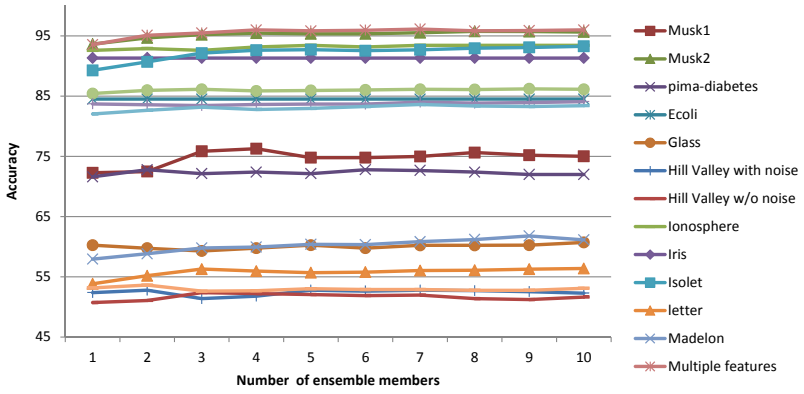

Figure 4: Accuracy of the DM ensemble using the Naïve Bayes inducer.

the chance of obtaining independent features is lower compared to when all features are selected. Moreover, given the voting scheme in Section 2, ensemble members which produce wrong classifications with high probabilities damage accurate classifications obtained by other ensemble members. Figure 3 demonstrates how the accuracy decreases as the number of members increases when RSE is paired with the Naïve Bayes inducer. This phenomenon is contrasted in Fig. 4 where the behavior that is expected from the ensemble is observed. Namely, an increase in accuracy when the number of ensemble members is increased when an ensemble different from the RSE is used (e.g. the DME).

In order to compare the 8 algorithms across all inducers and datasets we applied the procedure presented in [15]. The null hypothesis that all methods have the same accuracy could not be rejected by the adjusted Friedman test with a confidence level of 90% (specifically $F(7,350)=0.79 < 1.73$ with p-value>0.1). Furthermore, the results show there is a dependence between the

inducer, dataset and chosen dimensionality reduction algorithm. In the following we investigate the dependence between the latter two for each of the inducers.

6.2.1 Results for the nearest neighbor inducer (IB1)

In terms of the average improvement, the RPE algorithm is ranked first with an average improvement percentage of 5.8%. We compared the various algorithms according to their average rank following the steps described in [15]. The RSE and RPE achieved the first and second average rank, respectively. They were followed by Bagging (3^{rd}) and Rotation Forest (4^{th}).

Using the adjusted Friedman test we rejected the null hypothesis that all methods achieve the same classification accuracy with a confidence level of 95% and (7, 112) degrees of freedom (specifically F(7, 112)=2.47 $>$ 2.09 and p-value$<$0.022). Following the rejection of the null hypothesis, we employed the Nemenyi post-hoc test where in the experiment settings two classifiers are significantly different with a confidence level of 95% if their average ranks differ by at least $CD = 2.55$. The null hypothesis that any of the non-plain algorithms has the same accuracy as the plain algorithm could not be rejected at confidence level 95%.

6.2.2 Results for the decision tree inducer (J48)

Inspecting the average improvement, the RPE and RSE algorithms are ranked second and third, respectively, after the Rotation Forest algorithm. Following the procedure presented by Demsar [15], we compared the various algorithms according to their average rank. The RSE and DM+AdaBoost achieved the second and third best average rank, respectively, after the Rotation Forest algorithm.

The null hypothesis that all methods obtain the same classification accuracy was rejected by the adjusted Friedman test with a confidence level of 95% and (7, 112) degrees of freedom (specifically F(7, 112)=5.17 $>$ 2.09 and p-value$<$0.0001). As the null hypothesis was rejected, we employed the Nemenyi post-hoc test ($CD = 2.55$). Only the Rotation Forest algorithm significantly outperformed the plain and the DM algorithms. The null hypothesis that the RPE, RSE, DM and DM+AdaBoost algorithms have the same accuracy as the plain algorithm could not be rejected at confidence level 90%.

6.2.3 Results for the Naïve Bayes inducer

The DM+AdaBoost algorithm achieved the best average improvement and it is followed by the Rotation Forest algorithm. The DM, RPE and RSE are ranked

Table 2: Results of the ensemble classifiers based on the nearest-neighbor inducer (WEKA's IB1).
RPE is the Random Projection ensemble algorithm; **RSE** is the Random Subspaces ensemble algorithm; **DME** is the Diffusion Maps ensemble classifier; **DME+AdaBoost** is the multi-strategy ensemble classifier which applied AdaBoost to the Diffusion Maps dimension-reduced datasets.

Dataset	Plain NN	RPE	RSE	Bagging	DME	DME+AdaBoost	AdaBoost	Rotation Forest
Musk1	84.89 ± 4.56	86.15 ± 2.94	86.98 ± 4.18	86.77 ± 4.32	84.46 ± 4.31	84.87 ± 4.52	87.42 ± 4.24	84.88 ± 3.92
Musk2	95.80 ± 0.34	95.62 ± 0.38	96.04 ± 0.33	95.89 ± 0.31	95.39 ± 0.39	95.94 ± 0.49	96.03 ± 0.35	95.60 ± 0.62
pima-diabetes	70.17 ± 4.69	72.14 ± 4.03	70.83 ± 3.58	70.44 ± 3.89	66.79 ± 4.58	66.40 ± 4.82	67.30 ± 5.61	70.04 ± 4.17
Ecoli	80.37 ± 6.38	83.02 ± 3.52	83.05 ± 6.94	80.96 ± 5.43	77.37 ± 6.63	76.48 ± 8.23	78.87 ± 7.19	81.56 ± 4.97
Glass	70.52 ± 8.94	76.67 ± 7.22	77.58 ± 6.55	70.52 ± 8.94	72.88 ± 8.51	71.97 ± 7.25	70.95 ± 8.12	70.04 ± 8.24
Hill Valley with noise	59.83 ± 5.48	68.74 ± 3.58	59.75 ± 4.29	59.74 ± 4.77	50.49 ± 4.75	50.41 ± 4.49	58.42 ± 3.80	79.30 ± 3.60
Hill Valley w/o noise	65.84 ± 4.31	79.21 ± 3.19	66.66 ± 4.48	65.67 ± 4.26	55.36 ± 5.60	54.45 ± 5.18	63.20 ± 4.28	92.74 ± 2.10
Ionosphere	86.33 ± 4.59	90.02 ± 5.60	90.30 ± 4.32	86.90 ± 4.85	92.88 ± 4.09	93.44 ± 4.68	87.48 ± 3.55	86.61 ± 4.26
Iris	95.33 ± 5.49	93.33 ± 8.31	92.00 ± 10.80	96.00 ± 4.66	94.00 ± 5.84	94.00 ± 5.84	95.33 ± 5.49	94.00 ± 5.84
Isolet	89.94 ± 0.71	90.61 ± 0.86	90.57 ± 0.70	89.59 ± 0.65	91.32 ± 0.72	91.54 ± 0.87	89.00 ± 0.86	89.78 ± 0.78
Letter	96.00 ± 0.60	93.64 ± 0.32	94.08 ± 0.76	96.00 ± 0.57	90.58 ± 0.70	90.50 ± 0.76	95.10 ± 0.43	96.25 ± 0.55
Madelon	54.15 ± 4.28	68.95 ± 3.59	55.65 ± 2.63	54.80 ± 3.29	65.60 ± 1.94	65.10 ± 2.38	54.35 ± 4.76	55.20 ± 3.54
Multiple features	97.80 ± 0.63	95.65 ± 1.20	97.90 ± 0.66	97.85 ± 0.75	95.45 ± 1.42	95.55 ± 1.12	97.45 ± 0.64	97.70 ± 0.59
Sat	90.21 ± 1.16	91.34 ± 0.75	91.47 ± 0.71	90.37 ± 1.13	89.74 ± 0.57	89.40 ± 0.53	89.01 ± 1.32	90.82 ± 1.07
Waveform with noise	73.62 ± 1.27	80.14 ± 1.65	78.14 ± 2.35	73.74 ± 1.69	81.78 ± 0.93	80.72 ± 0.98	70.80 ± 2.03	73.94 ± 1.69
Waveform w/o noise	76.90 ± 2.01	81.22 ± 0.90	81.22 ± 1.47	77.14 ± 1.55	83.92 ± 1.38	83.12 ± 1.16	75.08 ± 1.70	77.72 ± 1.39
Yeast	52.29 ± 2.39	55.53 ± 4.39	49.32 ± 4.44	52.49 ± 2.16	48.99 ± 3.15	48.59 ± 4.31	51.35 ± 1.84	53.30 ± 2.44
Average improvement	-	**5.8%**	**1.7%**	**0.4%**	**-0.2%**	**-0.6%**	**-1%**	**4.6%**
Average rank	**4.97**	**3.26**	**3.15**	**4.35**	**5.18**	**5.29**	**5.44**	**4.35**

Table 3: Results of the Random Projection ensemble classifier based on the decision-tree inducer (WEKA's J48).
RPE is the Random Projection ensemble algorithm; **RSE** is the Random Subspaces ensemble algorithm; **DME** is the Diffusion Maps ensemble classifier; **DME+AdaBoost** is the multi-strategy ensemble classier which applied AdaBoost to the Diffusion Maps dimension-reduced datasets.

Dataset	Plain J48	RPE	RSE	Bagging	DME	DME+AdaBoost	AdaBoost	Rotation Forest
Musk1	84.90 ± 6.61	85.31 ± 6.25	88.45 ± 8.20	86.56 ± 6.93	78.60 ± 7.78	84.89 ± 5.44	88.46 ± 6.38	91.60 ± 3.10
Musk2	96.88 ± 0.63	96.30 ± 0.78	98.26 ± 0.39	97.65 ± 0.50	96.76 ± 0.72	97.23 ± 0.67	98.77 ± 0.35	98.18 ± 0.67
pima-diabetes	73.83 ± 5.66	73.83 ± 4.86	73.71 ± 6.04	75.26 ± 2.96	72.27 ± 3.11	72.40 ± 3.68	72.40 ± 4.86	76.83 ± 4.80
Ecoli	84.23 ± 7.51	86.00 ± 6.20	84.49 ± 7.28	84.79 ± 6.11	83.02 ± 4.10	81.27 ± 5.74	83.04 ± 7.37	86.60 ± 4.30
Glass	65.87 ± 8.91	72.94 ± 8.19	76.62 ± 7.38	75.19 ± 6.40	65.39 ± 10.54	68.12 ± 11.07	79.37 ± 6.13	74.22 ± 9.72
Hill Valley with noise	49.67 ± 0.17	71.28 ± 4.69	49.67 ± 0.17	54.62 ± 3.84	52.39 ± 3.56	52.39 ± 5.03	49.67 ± 0.17	74.51 ± 2.59
Hill Valley w/o noise	50.49 ± 0.17	86.38 ± 3.77	50.49 ± 0.17	50.99 ± 1.28	51.23 ± 4.40	52.39 ± 3.34	50.49 ± 0.17	83.83 ± 3.94
Ionosphere	91.46 ± 3.27	94.32 ± 3.51	93.75 ± 4.39	91.75 ± 3.89	88.04 ± 4.80	94.87 ± 2.62	93.17 ± 3.57	94.89 ± 3.45
Iris	96.00 ± 5.62	95.33 ± 6.32	94.67 ± 4.22	94.67 ± 6.13	92.00 ± 8.20	90.67 ± 9.53	93.33 ± 7.03	96.00 ± 4.66
Isolet	83.97 ± 1.65	87.37 ± 1.46	92.45 ± 1.14	90.46 ± 1.29	90.10 ± 0.62	93.86 ± 0.43	93.39 ± 0.67	93.75 ± 0.76
Letter	87.98 ± 0.51	88.10 ± 0.52	93.50 ± 0.92	92.73 ± 0.69	89.18 ± 0.79	91.46 ± 0.78	95.54 ± 0.36	95.41 ± 0.46
Madelon	70.35 ± 3.78	59.20 ± 2.57	76.95 ± 2.69	65.10 ± 3.73	76.15 ± 3.43	72.90 ± 2.27	66.55 ± 4.09	68.30 ± 2.98
Multiple features	94.75 ± 1.92	95.35 ± 1.31	97.35 ± 0.88	96.95 ± 1.07	93.25 ± 1.64	94.90 ± 1.73	97.60 ± 1.13	97.95 ± 1.04
Sat	85.83 ± 1.04	90.15 ± 0.93	91.10 ± 0.91	90.09 ± 0.78	91.34 ± 0.48	91.67 ± 0.37	90.58 ± 1.12	90.74 ± 0.69
Waveform with noise	75.08 ± 1.33	81.84 ± 1.43	82.02 ± 1.50	81.72 ± 1.43	86.52 ± 1.78	86.62 ± 1.76	80.48 ± 1.91	83.76 ± 2.07
Waveform w/o noise	75.94 ± 1.36	82.56 ± 1.56	82.52 ± 1.67	81.48 ± 1.27	86.96 ± 1.49	86.36 ± 0.94	81.46 ± 1.83	84.94 ± 1.47
Yeast	55.99 ± 4.77	57.82 ± 3.28	55.32 ± 4.06	59.23 ± 3.25	54.85 ± 3.94	55.39 ± 2.94	56.39 ± 5.08	60.71 ± 3.82
Average improvement	**-**	**8.5%**	**4.4%**	**3.8%**	**2.2%**	**3.5%**	**3.6%**	**12.2%**
Average rank	**6.26**	**4.56**	**4.03**	**4.44**	**5.68**	**4.41**	**4.5**	**2.15**

5^{th}, 7^{th} and 8^{th} in terms of the average improvement (possible reasons for the RSE algorithm's low ranking were described in the beginning of this section).

Employing the procedure presented in [15], we compared the algorithms according to their average ranks. The DM+AdaBoost and DM ensembles achieved the second and fourth best average ranks, respectively while the Rotation Forest and Bagging algorithms achieved the first and third places, respectively. The null hypothesis that all methods have the same classification accuracy was rejected by the adjusted Friedman test with a confidence level of 95% and (7, 112) degrees of freedom (specifically F(7, 112)=7.37 $>$ 2.09 and p-value$<$1e-6). Since the null hypothesis was rejected, we employed the Nemenyi post-hoc test. As expected, the RSE was significantly inferior to all other algorithms. Furthermore, the Rotation Forest algorithm was significantly better than the RPE algorithms. However, we could not reject at confidence level 95% the null hypothesis that the RPE, DM, DM+AdaBoost and the plain algorithm have the same accuracy.

When we compare the average accuracy improvement across all the inducers, the RPE and DM+AdaBoost were ranked second and third - improving the plain algorithm by 4% and 2.1%, respectively. The Rotation Forest algorithm is ranked first with 6.4% improvement. Comparing only the proposed ensembles according to their average rank as described in [15] yielded the following ranking: DM+AdaBoost, RPE, RSE, DM. The null hypothesis that the RPE, RSE, DM and DM+AdaBoost algorithms have the same accuracy as the plain algorithm could not be rejected at confidence level 90%. Thus, according to the average accuracy improvement across all the inducers, RPE performs best. However, according to the average rank, DM+AdaBoost performs best.

6.3 Discussion

The results indicate that when a dimensionality reduction algorithm is coupled with an appropriate inducer, an effective ensemble can be constructed. For example, the RPE algorithm achieves the best average improvements when it is paired with the nearest-neighbor and the decision tree inducers. However, when it is used with the Naïve Bayes inducer, it fails to improve the plain algorithm. On the other hand, the DM+AdaBoost ensemble obtains the best average improvement when it is used with the Naïve Bayes inducer (better than the current state-of-the-art Rotation Forest ensemble algorithm) and it is less accurate when coupled with the decision tree and nearest-neighbor inducers.

Furthermore, using dimensionality reduction as part of a multi-strategy ensemble classifier improved in most cases the results of the ensemble classifiers which employed only one of the strategies. Specifically, the DM+AdaBoost al-

Table 4: Results of the ensemble classifiers based on the Naïve Bayes inducer.
RPE is the Random Projection ensemble algorithm; **RSE** is the Random Subspaces ensemble algorithm; **DME** is the Diffusion Maps ensemble classifier; **DME+AdaBoost** is the multi-strategy ensemble classier which applied AdaBoost to the Diffusion Maps dimension-reduced datasets.

Dataset	Plain NB	RPE	RSE	Bagging	DME	DME+AdaBoost	AdaBoost	Rotation Forest
Musk1	75.25 ± 6.89	69.80 ± 8.98	56.52 ± 0.70	75.24 ± 7.11	55.90 ± 5.09	74.80 ± 2.88	77.10 ± 4.50	76.29 ± 6.76
Musk2	83.86 ± 2.03	77.36 ± 2.21	84.59 ± 0.07	83.71 ± 1.68	94.13 ± 0.50	95.74 ± 0.56	89.51 ± 1.98	83.98 ± 1.83
pima-diabetes	76.31 ± 5.52	70.18 ± 3.69	71.74 ± 5.37	76.83 ± 5.66	72.13 ± 4.50	71.88 ± 4.37	76.18 ± 4.69	74.09 ± 4.80
Ecoli	85.40 ± 5.39	86.92 ± 3.16	80.37 ± 5.91	87.18 ± 4.49	84.52 ± 5.43	84.52 ± 5.43	85.40 ± 5.39	86.31 ± 6.17
Glass	49.48 ± 9.02	48.07 ± 11.39	15.61 ± 10.16	50.82 ± 10.46	59.29 ± 11.09	60.24 ± 10.36	49.48 ± 9.02	54.16 ± 8.92
Hill Valley with noise	49.50 ± 2.94	49.75 ± 3.40	49.50 ± 2.94	50.74 ± 2.88	50.82 ± 2.93	53.63 ± 3.77	49.25 ± 3.39	52.14 ± 4.21
Hill Valley w/o noise	51.57 ± 2.64	50.82 ± 3.00	51.40 ± 2.52	51.90 ± 3.16	51.74 ± 3.25	52.06 ± 2.53	51.57 ± 2.61	52.56 ± 3.51
Ionosphere	82.62 ± 5.47	83.21 ± 6.42	67.80 ± 12.65	81.48 ± 5.42	92.59 ± 4.71	93.17 ± 3.06	92.04 ± 4.37	84.63 ± 5.02
Iris	96.00 ± 4.66	94.67 ± 4.22	96.67 ± 3.51	95.33 ± 5.49	91.33 ± 6.32	91.33 ± 6.32	93.33 ± 7.03	98.00 ± 3.22
Isolet	85.15 ± 0.96	89.06 ± 0.83	3.85 ± 0.00	85.58 ± 0.95	91.83 ± 0.96	92.97 ± 0.87	85.15 ± 0.96	90.68 ± 0.62
Letter	64.11 ± 0.76	59.27 ± 2.52	24.82 ± 8.81	64.18 ± 0.81	58.31 ± 0.70	56.90 ± 1.52	64.11 ± 0.76	67.51 ± 0.96
Madelon	58.40 ± 0.77	59.80 ± 2.06	50.85 ± 2.35	58.40 ± 0.84	55.10 ± 4.40	60.55 ± 4.01	53.65 ± 3.59	58.80 ± 1.51
Multiple features	95.35 ± 1.40	83.40 ± 2.22	10.90 ± 1.47	95.15 ± 1.25	89.05 ± 2.09	96.05 ± 1.28	96.40 ± 0.91	95.25 ± 1.57
Sat	79.58 ± 1.46	81.90 ± 1.13	69.82 ± 4.57	79.61 ± 1.50	85.63 ± 1.25	86.23 ± 1.16	79.58 ± 1.46	83.36 ± 1.52
Waveform with noise	80.00 ± 1.96	80.46 ± 1.76	72.04 ± 7.11	80.00 ± 2.01	84.36 ± 1.81	84.48 ± 1.46	80.00 ± 1.96	81.80 ± 1.81
Waveform w/o noise	81.02 ± 1.33	80.48 ± 2.03	74.48 ± 5.23	81.06 ± 1.35	82.94 ± 1.62	83.44 ± 1.72	81.02 ± 1.33	83.26 ± 1.59
Yeast	57.61 ± 3.01	55.39 ± 2.33	38.27 ± 7.89	57.82 ± 2.69	53.44 ± 3.96	53.37 ± 3.94	57.61 ± 3.01	55.46 ± 3.23
Average improvement	-	**-2.3%**	**-26.1%**	**0.4%**	**0.2%**	**3.4%**	**0.6%**	**2.3%**
Average rank	**4.71**	**5.41**	**7.15**	**3.97**	**4.29**	**3.06**	**4.59**	**2.82**

gorithm achieved higher average ranks compared to the DM and AdaBoost algorithms when the J48 and Naïve Bayes inducers were used. When the nearest-neighbor inducer was used, the DM+AdaBoost algorithm was ranked after the DM algorithm and before the AdaBoost ensemble which was last.

7 Conclusion and future work

In this paper we presented dimensionality reduction as a general framework for the construction of ensemble classifiers which use a single induction algorithm. The dimensionality reduction algorithm was applied to the training set where each combination of parameter values produced a different version of the training set. The ensemble members were constructed based on the produced training sets. In order to classify a new sample, it was first embedded into the dimension-reduced space of each training set using out-of-sample extension such as the Nyström extension. Then, each classifier was applied to the embedded sample and a voting scheme was used to derive the classification of the ensemble. This approach was demonstrated using three dimensionality reduction algorithms - Random Projections, Diffusion Maps and Random subspaces. A fourth ensemble algorithm employed a multi-strategy approach combining the Diffusion Maps dimensionality reduction algorithm with the AdaBoost ensemble algorithm. The performance of the obtained ensembles was compared with the Bagging, AdaBoost and Rotation Forest ensemble algorithms.

The results in this paper show that the proposed approach is effective in many cases. Each dimensionality reduction algorithm achieved results that were superior in many of the datasets compared to the plain algorithm and in many cases outperformed the reference algorithms. However, when the Naïve Bayes inducer was combined with the Random Subspaces dimensionality reduction algorithm, the obtained ensemble did not perform well in some of the datasets. Consequently, a question that needs further investigation is how to couple a given dimensionality reduction algorithm with an appropriate inducer to obtain the best performance. Ideally, rigorous criteria should be formulated. However, until such criteria are found, pairing dimensionality reduction algorithms with inducers in order to find the best performing pair can be done empirically using benchmark datasets. Furthermore, other dimensionality reduction techniques should be explored. For this purpose, the Nyström out-of-sample extension may be used with any dimensionality reduction method that can be formulated as a kernel method [23]. Additionally, other out-of-sample extension schemes should also be explored e.g. the Geometric Harmonics [13]. Lastly, a heterogeneous model which combines several dimensionality reduction techniques is currently being investigated by the authors.

Acknowledgments

The authors would like to thank Myron Warach for his insightful remarks.

References

[1] A. Asuncion and D. J. Newman. UCI machine learning repository, 2007.

[2] A. Averbuch, N. Rabin, A. Schclar, and V. A. Zheludev. Dimensionality reduction for detection of moving vehicles. *Pattern Analysis and Applications*, 15(1):19–27, 2012.

[3] A. Averbuch, V. Zheludev, N. Rabin, and A. Schclar. Wavelet based detection of moving vehicles. *International Journal of Wavelets, Multiresolution and Information Processing*, http://dx.doi.org/10.1007/s11045-008-0058-z, 2008.

[4] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[5] Y. Bengio, O. Delalleau, N. Le Roux, J. F. Paiement, P. Vincent, and M. Ouimet. Learning eigenfunctions links spectral embedding and kernel pca. *Neural Computation*, 16(10):2197–2219, 2004.

[6] E. Bingham and H. Mannila. Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2001)*, pages 245–250, San Francisco, CA, USA, August 26-29 2001.

[7] J. Bourgain. On lipschitz embedding of finite metric spaces in Hilbert space. *Israel Journal of Mathematics*, 52:46–52, 1985.

[8] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[9] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Chapman & Hall, Inc., New York, 1993.

[10] E. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, February 2006.

[11] F. R. K. Chung. *Spectral Graph Theory*. AMS Regional Conference Series in Mathematics, 92, 1997.

[12] R. R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis: special issue on Diffusion Maps and Wavelets*, 21:5–30, July 2006.

[13] R. R. Coifman and S. Lafon. Geometric harmonics: a novel tool for multiscale out-of-sample extension of empirical functions. *Applied and Computational Harmonic Analysis: special issue on Diffusion Maps and Wavelets*, 21:31–52, July 2006.

[14] T. Cox and M. Cox. *Multidimensional scaling*. Chapman & Hall, London, UK, 1994.

[15] J. Demsar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

[16] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, April 2006.

[17] D. L. Donoho and C. Grimes. Hessian eigenmaps: new locally linear embedding techniques for high-dimensional data. In *Proceedings of the National Academy of Sciences*, volume 100(10), pages 5591–5596, May 2003.

[18] H. Drucker. Improving regressor using boosting. In D. H. Fisher Jr., editor, *Proceedings of the 14th International Conference on Machine Learning*, pages 107–115. Morgan Kaufmann, 1997.

[19] X. Z. Fern and C. E. Brodley. Random projection for high dimensional data clustering: A cluster ensemble approach. In *International Conference on Machine Learning (ICML'03)*, pages 186–193, 2003.

[20] Y. Freund and R. Schapire. Experiments with a new boosting algorithm. machine learning. In *Proceedings for the Thirteenth International Conference*, pages 148–156, San Francisco, 1996. Morgan Kaufmann.

[21] N. Guan, D. Tao, Z. Luo, and J. Shawe-Taylor. Mahnmf: Manhattan non-negative matrix factorization. *CoRR*, abs/1207.3438, 2012.

[22] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: An update. *SIGKDD Explorations*, 11:1, 2009.

[23] J. Ham, D. Lee, S. Mika, and B. Scholköpf. A kernel view of the dimensionality reduction of manifolds. In *Proceedings of the 21st International Conference on Machine Learning (ICML'04)*, pages 369–376, New York, NY, USA, 2004.

[24] C. Hegde, M. Wakin, and R. G. Baraniuk. Random projections for manifold learning. In *Neural Information Processing Systems (NIPS)*, December 2007.

[25] M. Hein and Y. Audibert. Intrinsic dimensionality estimation of submanifolds in Euclidean space. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 289–296, 2005.

[26] T. K. Ho. The random subspace method for constructing decision forests. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.

[27] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, 1933.

[28] L. O. Jimenez and D. A. Landgrebe. Supervised classification in high-dimensional space: geometrical, statistical and asymptotical properties of multivariate data. *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews,*, 28(1):39–54, February 1998.

[29] W. B. Johnson and J. Lindenstrauss. Extensions of Lipshitz mapping into Hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

[30] J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29:1–27, 1964.

[31] L. I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004.

[32] L. I. Kuncheva. Diversity in multiple classifier systems (editorial). *Information Fusion*, 6(1):3–4, 2004.

[33] S. Lafon, Y. Keller, and R. R. Coifman. Data fusion and multicue data matching by diffusion maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:1784–1797, 2006.

[34] W. Leigh, R. Purvis, and J. M. Ragusa. Forecasting the nyse composite index with technical analysis, pattern recognizer, neural networks, and genetic algorithm: a case study in romantic decision support. *Decision Support Systems*, 32(4):361–377, 2002.

[35] J. Li, N. M. Allinson, D. Tao, and X. Li. Multitraining support vector machine for image retrieval. *IEEE Transactions on Image Processing*, 15(11):3597–3601, 2006.

[36] M. Linial, N. Linial, N. Tishby, and G. Yona. Global self-organization of all known protein sequences reveals inherent biological signatures. *Journal of Molecular Biology*, 268(2):539–556, May 1997.

[37] Y. Luo, D. Tao, B. Geng, C. Xu, and S. Maybank. Manifold regularized multi-task learning for semi-supervised multi-label image classification. *IEEE Transaction on Image Processing*, 22(2):523–536, 2013.

[38] P. Mangiameli, D. West, and R. Rampal. Model selection for medical diagnosis decision support systems. *Decision Support Systems*, 36(3):247–259, 2004.

[39] D. D. Margineantu and T. G. Dietterich. Pruning adaptive boosting. In *Proceedings of the 14th International Conference on Machine Learning*, pages 211–218, 1997.

[40] E. J. Nyström. Über die praktische auflösung von linearen integralgleichungen mit anwendungen auf randwertaufgaben der potentialtheorie. *Commentationes Physico-Mathematicae*, 4(15):1–52, 1928.

[41] D. Opitz and R. Maclin. Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 11:169 to 198, 1999.

[42] F. Plastria, S. Bruyne, and E. Carrizosa. Dimensionality reduction for classification. *Advanced Data Mining and Applications*, 1:411–418, 2008.

[43] R. Polikar. "ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, 6:21 t o 45, 2006.

[44] R. R. Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., 1993.

[45] J. J. Rodriguez, L. I. Kuncheva, and C. J. Alonso. Rotation forest: A new classifier ensemble method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1619–1630, 2006.

[46] L. Rokach. Mining manufacturing data using genetic algorithm-based feature set decomposition. *International Journal of Intelligent Systems Technologies and Applications*, 4(1/2):57–78, 2008.

[47] L. Rokach. Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography. *Computational Statistics & Data Analysis*, In Press, Corrected Proof:–, 2009.

[48] N. Rooney, D. Patterson, A. Tsymbal, and 10 February 2004 S. Anand. Random subspacing for regression ensembles. Technical report, Department of Computer Science, Trinity College Dublin, Ireland, 2004.

[49] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, December 2000.

[50] A. Schclar. A diffusion framework for dimensionality reduction. In *Soft Computing for Knowledge Discovery and Data Mining (Editors: O. Maimon and L. Rokach)*, pages 315–325. Springer, 2008.

[51] A. Schclar, A. Averbuch, N. Rabin, V. Zheludev, and K. Hochman. A diffusion framework for detection of moving vehicles. *Digital Signal Processing*, 20:111–122, January 2010.

[52] A. Schclar and L. Rokach. Random projection ensemble classifiers. In *Lecture Notes in Business Information Processing, Enterprise Information Systems 11th International Conference Proceedings (ICEIS'09)*, pages 309–316, Milan, Italy, May 2009.

[53] A. Schclar, A. Tsikinovsky, L. Rokach, A. Meisels, and L. Antwarg. Ensemble methods for improving the performance of neighborhood-based collaborative filtering. In *RecSys*, pages 261–264, 2009.

[54] B. Schölkopf, A. Smola, and K. R. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[55] B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

[56] D. P. Solomatine and D. L. Shrestha. Adaboost.rt: A boosting algorithm for regression problems. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, pages 1163–1168, 2004.

[57] D. Tao, X. Li, X. Wu, and J. S. Maybank. Geometric mean for subspace selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):260–274, February 2009.

[58] D. Tao, X. Tang, X. Li, and X. Wu. Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7):1088–1099, July 2006.

[59] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, December 2000.

[60] G. Valentini, M. Muselli, and F. Ruffino. Bagged ensembles of svms for gene expression data analysis. In *Proceeding of the International Joint*

Conference on Neural Networks - IJCNN, pages 1844–1849, Portland, OR, USA, July 2003. Los Alamitos, CA: IEEE Computer Society.

[61] V. N. Vapnik. *The Nature of Statistical Learning Theory (Information Science and Statistics)*. Springer, November 1999.

[62] G. I. Webb. Multiboosting: A technique for combining boosting and wagging. In *Machine Learning*, pages 159–196, 2000.

[63] G. I. Webb and Z. Zheng. Multi-strategy ensemble learning: Reducing error by combining ensemble learning techniques. *IEEE Transactions on Knowledge and Data Engineering*, 16:2004, 2004.

[64] Z. Yang, X. Nie, W. Xu, and J. Guo. An approach to spam detection by naive bayes ensemble based on decision induction. In *Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications (ISDA'06)*, 2006.

[65] T. Zhang, D. Tao, X. Li, and J. Yang. Patch alignment for dimensionality reduction. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1299–1313, September 2009.

[66] Z. Zhang and H. Zha. Principal manifolds and nonlinear dimension reduction via local tangent space alignment, 2002.

[67] T. Zhou, D. Tao, and X. Wu. Manifold elastic net: a unified framework for sparse dimension reduction. *Data Mining and Knowledge Discovery*, 22(3):340–371, May 2011.