Robust Linear Dimensionality Reduction

Yehuda Koren and Liran Carmel

***Abstract*— We present a novel family of data-driven linear transformations, aimed at finding low dimensional embeddings of multivariate data, in a way that optimally preserves the structure of the data. The well-studied PCA and Fisher's LDA are shown to be special members in this family of transformations, and we demonstrate how to generalize these two methods such as to enhance their performance. Furthermore, our technique is the only one, to the best of our knowledge, that reflects in the resulting embedding both the data coordinates and pairwise relationships between the data elements. Even more so, when information on the clustering (labeling) decomposition of the data is known, this information can also be integrated in the linear transformation, resulting in embeddings that clearly show the separation between the clusters, as well as their internal structure. All this makes our technique very flexible and powerful, and lets us cope with kinds of data that other techniques fail to describe properly.**

***Index Terms*— Dimensionality reduction, visualization, classification, feature extraction, projection, linear transformation, principal component analysis, Fisher's linear discriminant analysis.**

I. Introduction

DIMENSIONALITY reduction is one of the key techniques in data analysis, aimed at revealing meaningful structures and unexpected relationships in multivariate data. It assembles numerous methods, all striving to present high dimensional data in a low dimensional space, in a way that faithfully captures desired structural elements of the data. Dimensionality reduction is used for many purposes. For example, it is beneficial as a visualization tool to present multivariate data in a human accessible form, as a method of feature extraction, and as a preliminary transformation applied to the data prior to the use of other analysis tools like clustering and classification. There are many criteria that can be used to sort the various methods of dimensionality reduction. In this paper, we have found it very useful to use two dichotomies — coordinate based methods versus weight based ones (which is essentially the dichotomy between entities and relationships), and linear methods versus nonlinear ones.

Almost always, multivariate data are supplied in one of two basic forms. Either each data element is a vector of (potentially many) variables, or some numeric value is provided to describe the relationships between each pair of data elements. In the first case, we use the term *coordinates* to denote the different entries of the data elements, and those dimensionality reduction techniques that can deal with such data are called *coordinate based* methods. In the second case, we use the term *weights* for the pairwise relationships between the data elements, and those dimensionality reduction techniques that can deal with such data are called *weight based* methods.

Weight based methods attempt to assign coordinates to the data elements in the low dimensional space, such that their embedding reflects in some sense their pairwise relationships. Distances, similarities and dissimilarities are the most commonly used types of weights. *Multidimensional scaling* is the customary notion for these methods that use distances or dissimilarities as weights. See description of many such techniques in, e.g., [5], [10], [12].

Coordinate based methods compute a mapping of high dimensional data into lower dimensional ones. A *linear* method is one for which the mapping can be described by a linear transformation. Here, we shall denote any other method as *nonlinear* method. Many coordinate based methods, linear as well as nonlinear, are introduced in, e.g., [5], [12].

In this paper we present a novel family of dimensionality reduction techniques, which show a rather broad spectrum of appealing properties. One of the most salient of these is that it "spoils" the dichotomy coordinates/weights by allowing for the merger of both forms in a single framework. One way to look at our methods is as coordinate based methods, that are capable of taking into consideration pairwise weights too, if these are available.

Another prominent property of the methods to be described here is that they are linear. Despite being more limited than their nonlinear counterparts, linear dimensionality reduction techniques possess several significant advantages:

1) The low dimensional data is reliable in the sense that it is guaranteed to show genuine properties of the original data. In contrast, nonlinear techniques might unrecognizably deform the topology of the original high dimensional data.
2) The low dimensional axes are meaningful as they are linear combinations of the original axes. Sometimes, analysis of these combinations can induce interesting domain-specific interpretations.
3) The transformation matrix can be stored in memory and be used whenever new data elements should undergo the same transformation as the original data.
4) In general, the computational complexity of linear methods is very low, both in time and in space.

We shall later prove that our methods generalize the well known principal component analysis and Fisher's linear discriminant analysis, which are both linear dimensionality reduction techniques.

In this paper we will put special emphasize on yet another important property of our methods, namely their robustness. As dimensionality reduction techniques are data-driven, they might exhibit undesirable sensitivity to outliers (extreme observations that are well separated from the remainder of the data).

Y. Koren is with AT&T Labs – Research, NJ 07208, USA. Email: yehuda@research.att.com

L. Carmel is with the National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health, Bethesda, MD 20894, USA. Email: carmel@ncbi.nlm.nih.gov

We will show that we can make our methods especially robust in this aspect, by using a special from of pairwise weighting.

The last property of our methods which we would like to mention here is that they can also take into consideration knowledge about data labeling, which is the situation where the data elements are partitioned into disjoint clusters by some external source, normally a clustering algorithm or a domain specific knowledge. Reducing dimensionality of such data requires special effort since, besides the desire to convey the overall structure, we would also like to reflect the inter-cluster and intra-cluster relationships. We will show how we can design our methods to show the different clusters as separate as possible, as well as to preserve intra-cluster structure. The resulting embeddings can be very instructive in validating the results of a clustering algorithm or in revealing the structure of the clusters and their relationships.

II. Basic Notions

Throughout the paper, we shall always assume n data elements represented by m coordinates, arranged row-wise in an $n \times m$ coordinate matrix X (such that $X_{i\alpha}$ is the αth coordinate of the ith data element). Hereinafter, we shall use Greek indices for coordinates and Latin indices for data elements. Without loss of generality we assume that the data is centered, meaning that the mean of each coordinate over the entire dataset is zero, $\sum_{i=1}^{n} X_{i\alpha} = 0$ for $\alpha = 1,\ldots,m$. This can always be achieved by a harmless translation of the data. We shall denote by S the $m \times m$ *biased covariance matrix*, $S = \frac{1}{n} X^T X$.

Dimensionality reduction aims at finding a meaningful representation of the data in p dimensions. By definition $p < m$, but in this paper we shall also always assume that $p < n$, which means that we require a minimal number of data elements. As p is typically small, this requirement is met for any plausible dataset. In linear dimensionality reduction, it is customary to characterize the low dimensional space by p *direction vectors* $v^1,\ldots,v^p \in \mathbb{R}^m$, so that the α-coordinate vector of the transformed data ($1 \leqslant \alpha \leqslant p$) is obtained by projecting the data on the αth direction vector, $Xv^\alpha \in \mathbb{R}^n$. Consequently, we shall call the vectors $Xv^1,\ldots,Xv^p$ the *coordinate vectors*.

We denote by dist_{ij} the Euclidean distance between elements i and j (in the original space), $\mathrm{dist}_{ij} = \sqrt{\sum_{\alpha=1}^{m}(X_{i\alpha} - X_{j\alpha})^2}$. When referring to Euclidean distances in a p-dimensional embedding of the data, we shall add the superscript p, thus $\mathrm{dist}^p_{ij} = \sqrt{\sum_{\alpha=1}^{p}((Xv^\alpha)_i - (Xv^\alpha)_j)^2}$.

The *Laplacian* is a key entity for describing pairwise relationships between data elements. This is an $n \times n$ symmetric positive-semidefinite matrix, characterized by having zero row and column sums. Its usefulness stems from the fact that the quadratic form associated with it is just a weighted sum of all pairwise squared distances:

Lemma II.1 *Let L be an $n \times n$ Laplacian, and let $x \in \mathbb{R}^n$. Then*

$$x^T L x = \sum_{i<j} -L_{ij}(x_i - x_j)^2.$$

Similarly, for p coordinate vectors $x^1,\ldots,x^p \in \mathbb{R}^n$ we have:

$$\sum_{\alpha=1}^{p} (x^\alpha)^T L x^\alpha = \sum_{i<j} -L_{ij} \cdot \left(\sum_{\alpha=1}^{p} ((x^\alpha)_i - (x^\alpha)_j)^2 \right) = \\ = \sum_{i<j} -L_{ij} \cdot \left(\mathrm{dist}^p_{ij}\right)^2.$$

The proof of this lemma is direct.

Next, we develop some essential mathematical background that is needed for subsequent derivations. Different parts of this material can be found in standard linear algebra textbooks. The casual reader can make do with understanding Theorem II.1, Theorem II.2 and Corollary II.1, and does not have to delve into the proofs. In the following, δ_{ij} is the Kronecker delta (defined as 1 for $i = j$ and as 0 otherwise), and A^α is the αth column of matrix A.

Theorem II.1 *Let A be an $n \times n$ symmetric matrix. Denote by $\lambda_1 \geqslant \cdots \geqslant \lambda_n$ its sorted eigenvalues, and by $u^1,\ldots,u^n$ the corresponding eigenvectors. Then $u^1,\ldots,u^p$ are the maximizer of the constrained maximization problem*

$$\max_{v^1,\ldots,v^p} \quad \sum_{\alpha=1}^{p} (v^\alpha)^T A v^\alpha \\ \text{subject to: } (v^\alpha)^T v^\beta = \delta_{\alpha\beta}, \quad \alpha,\beta = 1,\ldots,p. \tag{1}$$

Similarly, the minimizer of the same problem are the p lowest eigenvectors $u_{n-p+1},\ldots,u_n$.

For the proof, we first need the following lemma.

Lemma II.2 *Let X be an $n \times p$ matrix with orthonormal columns (i.e., $X^T X = I$), and let $v^1,\ldots,v^{p-1} \in \mathbb{R}^n$ be any vectors. Then there is an $n \times p$ matrix with orthonormal columns, Y, such that for every $2 \leqslant \alpha \leqslant p$, Y^α is orthogonal to $v^1,\ldots,v^{\alpha-1}$. Additionally, for every $n \times n$ matrix A,* $\mathrm{trace}(X^T A X) = \mathrm{trace}(Y^T A Y)$.

Proof: In this proof we use the symbol $\mathscr{R}(X)$ for the range of matrix X (defined as $\mathrm{span}(X^1,\ldots,X^p)$). Let us denote the projection of v^1 into $\mathscr{R}(X)$ by $\tilde{v}^1$. If $\tilde{v}^1 = 0$ then we set $Y^1 = X^1$ and $\hat{Y}^2,\ldots,\hat{Y}^p = X^2,\ldots,X^p$. Obviously, $\hat{Y}^2,\ldots,\hat{Y}^p$ are orthogonal to v^1. If $\tilde{v}^1 \neq 0$, we rotate $X^1, X^2,\ldots,X^p$ within $\mathscr{R}(X)$ obtaining $Y^1, \hat{Y}^2,\ldots,\hat{Y}^p$, such that $Y^1 = \tilde{v}^1/\|\tilde{v}^1\|$. Since rotations do not alter orthogonality relations, we still have that $\hat{Y}^2\ldots,\hat{Y}^p$ are orthogonal to v^1. We continue recursively with the vectors $v^2,\ldots,v^{p-1}$ and the matrix $(\hat{Y}^2\ldots\hat{Y}^p)$. Note that during the recursion, the v^1 and Y^1 orthogonality that were already achieved are not ruined, since we limit the rotations to the space $\mathrm{span}(\hat{Y}^2,\ldots,\hat{Y}^p)$ which is orthogonal to v^1 and to Y^1. At the end of the process we obtain p orthonormal vectors $Y^1,\ldots,Y^p$ that satisfy the requested property.

Since all rotations are performed within $\mathscr{R}(X)$, there is some $p \times p$ matrix R such that $Y = XR$. Since X and Y have orthonormal columns we use $I = Y^T Y = R^T X^T X R = R^T R$ to conclude that R is an orthonormal matrix, and so $RR^T = I$. We use the fact that the trace is cyclically-commutative to obtain

$$\mathrm{trace}(Y^T A Y) = \mathrm{trace}(R^T X^T A X R) = \mathrm{trace}(RR^T X^T A X) = \\ = \mathrm{trace}(X^T A X).$$

■

Now we can turn to prove Theorem II.1.

Proof: Let $v^1,\ldots,v^p$ be arranged in the $n\times p$ matrix V. This allows us to rewrite (1) in the simple matrix notation

$$\begin{aligned}&\max_{V}\quad \mathrm{trace}(V^TAV)\\ &\text{subject to: } V^TV=I.\end{aligned}\tag{2}$$

Let $V_0=(v_0^1,\ldots,v_0^p)$ be the maximizer of (2). Since the eigenvectors $u^1,\ldots,u^n$ form a basis of $\mathbb{R}^n$, we can decompose each v_0^α as a linear combination $v_0^\alpha=\sum_{\beta=1}^n c_\beta^\alpha u^\beta$. Lemma II.2 allows us to assume, without loss of generality, that for every $2\leqslant\alpha\leqslant p$, v_0^α is orthogonal to the eigenvectors $u^1,\ldots,u^{\alpha-1}$. We may therefore take $c_\beta^\alpha=0$ for $\beta<\alpha$, and write $v_0^\alpha=\sum_{\beta=\alpha}^n c_\beta^\alpha u^\beta$. Next, we use the constraint $(v_0^\alpha)^Tv_0^\alpha=1$ to obtain an equation for the coefficients c_β^α,

$$1=(v_0^\alpha)^Tv_0^\alpha=\left(\sum_{\beta=\alpha}^n c_\beta^\alpha u^\beta\right)^T\left(\sum_{\beta=\alpha}^n c_\beta^\alpha u^\beta\right)=\sum_{\beta=\alpha}^n\left(c_\beta^\alpha\right)^2,$$

where the last equality stems from the orthonormality of $u^1,\ldots,u^n$. Using this result, we can expand the quadratic form $(v_0^\alpha)^TAv_0^\alpha$ as

$$\begin{aligned}(v_0^\alpha)^TAv_0^\alpha&=\left(\sum_{\beta=\alpha}^n c_\beta^\alpha u^\beta\right)^TA\left(\sum_{\beta=\alpha}^n c_\beta^\alpha u^\beta\right)=\\ &=\left(\sum_{\beta=\alpha}^n c_\beta^\alpha u^\beta\right)^T\left(\sum_{\beta=\alpha}^n c_\beta^\alpha\lambda_\beta u^\beta\right)=\\ &=\sum_{\beta=\alpha}^n\left(c_\beta^\alpha\right)^2\lambda_\beta\leqslant\sum_{\beta=\alpha}^n\left(c_\beta^\alpha\right)^2\lambda_\alpha=\lambda_\alpha.\end{aligned}$$

Thus, the maximum of the target function is bounded by

$$\mathrm{trace}(V_0^TAV_0)=\sum_{\alpha=1}^p(v_0^\alpha)^TAv_0^\alpha\leqslant\sum_{\alpha=1}^p\lambda_\alpha.$$

Since $\sum_{\alpha=1}^p(u^\alpha)^TAu^\alpha=\sum_{\alpha=1}^p\lambda_\alpha$, we conclude that indeed the highest eigenvectors $u^1,\ldots,u^p$ are the maximizer of (1).

The proof for the minimization problem goes along exactly the same lines. ■

Theorem II.1 requires orthonormality relations between the vectors $v^1,\ldots,v^p$. In the following theorem we generalize this result by allowing for these vector to be mutually conjugate. Hereinafter, the generalized eigenvalue problem $Ax=\lambda Bx$ is referred to as the generalized eigenvalue problem of (A,B).

Theorem II.2 *Let A be an $n\times n$ symmetric matrix, and let B be an $n\times n$ positive definite matrix. Denote by $\lambda_1\geqslant\cdots\geqslant\lambda_n$ the sorted generalized eigenvalues of the generalized eigenvalue problem of (A,B), and by $u^1,\ldots,u^n$ the corresponding generalized eigenvectors. Then $u^1,\ldots,u^p$ are the maximizer of the constrained maximization problem*

$$\begin{aligned}&\max_{v^1,\ldots,v^p}\quad\sum_{\alpha=1}^p(v^\alpha)^TAv^\alpha\\ &\text{subject to: } (v^\alpha)^TBv^\beta=\delta_{\alpha\beta},\quad\alpha,\beta=1,\ldots,p.\end{aligned}\tag{3}$$

Similarly, the minimizer of the same problem are the p lowest generalized eigenvectors $u_{n-p+1},\ldots,u_n$.

Proof: Since B is positive definite it can be decomposed into $B=C^TC$, with C an $n\times n$ invertible matrix (e.g., using Cholesky decomposition). Making the substitution $v^\alpha=C^{-1}\hat{v}^\alpha$ in (3), we reformulate the problem as

$$\begin{aligned}&\max_{\hat{v}^1,\ldots,\hat{v}^p}\quad\sum_{\alpha=1}^p(\hat{v}^\alpha)^TC^{-T}AC^{-1}\hat{v}^\alpha\\ &\text{subject to: } (\hat{v}^\alpha)^T\hat{v}^\beta=\delta_{\alpha\beta},\quad\alpha,\beta=1,\ldots,p.\end{aligned}\tag{4}$$

Let $\hat{v}_0^1,\ldots,\hat{v}_0^p$ be the maximizer of (4). According to Theorem II.1 these are simply the p highest eigenvectors of $C^{-T}AC^{-1}$, obeying therefore $C^{-T}AC^{-1}\hat{v}_0^\alpha=\lambda_\alpha\hat{v}_0^\alpha$. Using this equation, and transforming back into the vectors $v_0^\alpha=C^{-1}\hat{v}_0^\alpha$, we get $C^{-T}Av_0^\alpha=\lambda_\alpha Cv_0^\alpha$, which is just $Av_0^\alpha=\lambda_\alpha Bv^\alpha$. Hence, the maximizer of (3) is nothing but the p highest generalized eigenvectors of (A,B). An identical proof can be used to prove the claim for the minimization. ■

The optimization problem (3) is a fundamental building block in this paper. Sometimes, it is more comprehensible to use a different, but completely equivalent, formulation.

Corollary II.1 *The problem*

$$\begin{aligned}&\max_{v^1,\ldots,v^p}\quad\frac{\sum_{\alpha=1}^p(v^\alpha)^TAv^\alpha}{\sum_{\alpha=1}^p(v^\alpha)^TBv^\alpha}\\ &\text{subject to: } (v^\alpha)^TBv^\beta=\delta_{\alpha\beta},\quad\alpha,\beta=1,\ldots,p.\end{aligned}\tag{5}$$

has the same maximizer as problem (3)*. The same goes with the minimization problem.*

The proof of this corollary is immediate.

III. A Generalized Projection Scheme

An important and fundamental family of linear dimensionality reduction transformations is the set of projections into a low dimensional space. In algebraic terms, projections are characterized by having all the direction vectors orthonormal, namely

$$(v^\alpha)^Tv^\beta=\delta_{\alpha\beta},\quad\alpha,\beta=1,\ldots,p.\tag{6}$$

In a sense, projections preserve the structure of the data more authentically than other linear transformations, since they can be achieved by a rigid rotation of the data, followed by a selection of a subgroup of the axes.

A. Derivation of PCA

Principal component analysis (PCA) is probably the most widely used and well studied projection used for dimensionality reduction. A comprehensive discussion on PCA can be found in many textbooks, see, e.g., [5], [12]. PCA projects (possibly) correlated variables into a (possibly lower number of) uncorrelated variables called *principal components*. The first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible. By using only the first few principal components, PCA makes it possible to reduce the number of significant dimensions of the data, while maintaining the maximum possible variance

thereof. Formally, it can be shown that the orthonormal direction vectors $v^1,\ldots,v^p$ in PCA should be taken as the p highest unit eigenvectors of the covariance matrix S.

Much intuition on this technique is gained by understanding that PCA is the best variance-preserving projection. Here, we would like to gain even more intuition by deriving PCA using a different, although related, motivation. This derivation will later enable us to suggest significant generalizations of PCA. In the following theorem we show that PCA finds the projection that maximizes the sum of all squared pairwise distances between the projected data elements.

Theorem III.1 *PCA computes the p-dimensional projection that maximizes*

$$\sum_{i<j}\left(\mathrm{dist}_{ij}^p\right)^2. \tag{7}$$

This theorem implies intimate relationships between PCA and multidimensional scaling. Despite the fact that the former is coordinate based, while the latter is weight based, both methods share, in a sense, similar objectives. Clearly $\mathrm{dist}_{ij}^p \leqslant \mathrm{dist}_{ij}$ for any p-dimensional projection and any two elements i,j, and so

$$\sum_{i<j}\left(\mathrm{dist}_{ij}^p\right)^2 \leqslant \sum_{i<j}(\mathrm{dist}_{ij})^2.$$

Consequently, Theorem III.1 shows that PCA computes the projection that maximizes the preservation of pairwise distances, similar to what multidimensional scaling strives to achieve.

Before proving Theorem III.1, we define the $n\times n$ *unit Laplacian*, denoted by L^u, as $L^u_{ij}=\delta_{ij}\cdot n-1$. The unit Laplacian satisfies the following lemma:

Lemma III.1 *The matrices X^TL^uX and S are identical up to a positive multiplicative factor, $X^TL^uX=n^2\cdot S$.*

Proof: We shall examine a specific entry of the matrix,

$$\begin{aligned}(X^TL^uX)_{\alpha\beta}&=\sum_{i,j=1}^n L^u_{ij}X_{i\alpha}X_{j\beta}=\sum_{i,j=1}^n(n\cdot\delta_{ij}-1)X_{i\alpha}X_{j\beta}=\\&=\sum_{i=1}^n n\cdot X_{i\alpha}X_{i\beta}-\sum_{i,j=1}^n X_{i\alpha}X_{j\beta}=\\&=n(X^TX)_{\alpha\beta}-\sum_{i=1}^n X_{i\alpha}\cdot\sum_{j=1}^n X_{j\beta}=n^2S_{\alpha\beta}.\end{aligned}$$

The last equality is due to the fact that the coordinates are centered. ∎

Now we can turn to prove Theorem III.1.

Proof: Recall that the data coordinates in the p-dimensional projection are given by $Xv^1,\ldots,Xv^p$. By Lemma II.1 we get

$$\sum_{i<j}(\mathrm{dist}_{ij}^p)^2=\sum_{\alpha=1}^p(Xv^\alpha)^TL^u(Xv^\alpha)=\sum_{\alpha=1}^p(v^\alpha)^TX^TL^uXv^\alpha.$$

Hence, a projection maximizing (7) can be formally posed as the solution of

$$\begin{aligned}&\max_{v^1,\ldots,v^p}\ \sum_{\alpha=1}^p(v^\alpha)^TX^TL^uXv^\alpha\\&\text{subject to: }(v^\alpha)^Tv^\beta=\delta_{\alpha\beta},\quad\alpha,\beta=1,\ldots,p.\end{aligned} \tag{8}$$

By Theorem II.1 the maximizer of (8) is the set of p highest eigenvectors of the matrix X^TL^uX. By Lemma III.1, these are also the p highest eigenvectors of the covariance matrix (multiplication of a matrix by a positive constant does not change the eigenvectors or their order). Hence, the solution of (8) is achieved exactly by the first p principal components. ∎

B. Weighted PCA

Formulating PCA as in (8) implies a straightforward generalization — simply replace the unit Laplacian with a general one in the target function. In the notation of Theorem III.1, this means that the p-dimensional projection will maximize a weighted sum of squared distances, instead of an unweighted sum. Hence, it would be natural to call such a projection method by the name *weighted PCA*.

Let us formalize this idea. Let $\{d_{ij}\}_{i,j=1}^n$ be symmetric nonnegative pairwise weights, with d_{ij} measuring how important it is for us to place the data elements i and j further apart in the low dimensional space. By convention, $d_{ij}=0$ for $i=j$. For this reason, these weights will be called *dissimilarities* in the context of weighted PCA. Normally, they are either supplied from an external source, or calculated from the data coordinates, in order to reflect any desired relationships between the data elements. Generalizing (7), we now seek for the projection that maximizes

$$\sum_{i<j}d_{ij}\left(\mathrm{dist}_{ij}^p\right)^2. \tag{9}$$

The $n\times n$ Laplacian L^d associated with the dissimilarities is

$$L^d_{ij}=\begin{cases}\sum_{j=1}^n d_{ij} & i=j\\ -d_{ij} & i\neq j.\end{cases}$$

This Laplacian is intimately related to the weighted PCA, as is clear from the following proposition.

Proposition III.1 *The p-dimensional projection that maximizes*

$$\sum_{i<j}d_{ij}\left(\mathrm{dist}_{ij}^p\right)^2$$

is obtained by taking the direction vectors to be the p highest eigenvectors of the matrix X^TL^dX.

Proof: The proof is the same as that of Theorem III.1, just replace L^u with L^d. ∎

Here we see a first example of a very fundamental property of our dimensionality reduction method. We have a full control over all the pairwise relationships between the data elements, involving very large matrices (the Laplacian L^d is $n\times n$, with $\frac{1}{2}n(n-1)$ degrees of freedom). However, the massive calculations (finding the eigenvectors) are carried out over the $m\times m$ matrix X^TL^dX. As m is typically much smaller than n, the computation is very fast.

When would we like to apply such a weighted version of PCA? Well, there may be many occasions. Sometimes, we may have external knowledge about the dissimilarities, which we would like to incorporate in the projection. An example of such a dataset will be discussed later, in Subsection IV-A.

In other cases, dissimilarities can be calculated directly from the coordinates, and used to overcome certain drawbacks of the standard PCA. In the next two subsections, we suggest two different ways to calculate dissimilarities — the first is designed to cope with outliers in the data, and the second is designed to cope with labeled data.

1) Normalized PCA: As we have shown in Theorem III.1, PCA strives to maximize the sum of all squared distances. The fact that the distances are squared puts much more emphasis on the preservation of large distances, frequently at the expense of the preservation of shorter distances. In many cases, for example when outliers are present, this behavior might impair the results of PCA. Since pairwise distances involving outliers are significantly larger than the other pairwise distances, PCA tends to preserve outlying structures, sometimes by significantly slanting the projection. Indeed, PCA is known for its extreme sensitivity to outliers, which frequently appear in real-world datasets. We illustrate this phenomenon in Figure 1, where we present a synthetic two-dimensional dataset, comprising a bulk of 50 normally-distributed points as well as two outlying points. As can be seen in the figure, the one-dimensional projection computed by PCA projects the data in a direction that emphasizes the outliers while hiding almost all of the structure of the bulky region.

The concept of weighted PCA may be used to significantly improve the outlier robustness of PCA, by underweighting distant data elements. A natural way to do this is to take the dissimilarities as

$$d_{ij} = \frac{1}{\mathrm{dist}_{ij}} .$$

The resulting projections are well balanced, aiming at preserving both large and small pairwise distances. We have found this method, which we call *normalized PCA*, to be superior to the standard PCA, especially when the data contain outliers. Figure 1 exemplifies this, as the one-dimensional projection achieved by normalized PCA is demonstrated to preserve much better the overall structure of the dataset.

As another instructive example, Figure 2 shows three two-dimensional projections of the four-dimensional *sleep* dataset, consists of the body weight, brain weight, maximum life span and gestation time of thirty mammals. This is a portion of a larger dataset (columns with missing data omitted), originally referred to in [1], and is publicly available in [11]. Figure 2a shows the projection obtained by PCA. We see that the data are concentrated in one elongated cluster, except for three outliers — man, Asian elephant and African elephant. Ideally, we would expect the first principal component to account for the variability in the main cluster, namely to point to the direction shown by the arrow in the figure. This, however, does not happen since the first principal component "works hard" to separate the outliers from the main bulk and from each other. Applying normalized PCA gives the projection shown in Figure 2b, where we see a significant "straightening" of the first principal component. Yet, it still does not point in the direction of the arrow, and is nonetheless influenced by the three outliers. It looks as if the outliers are still dominant, and that a more drastic underweighting is needed. Therefore, we have tried to use a version of normalized PCA,

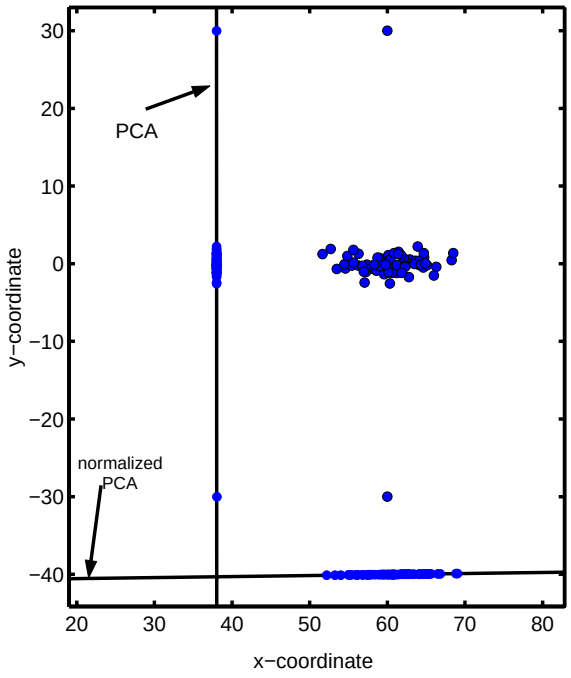


Fig. 1. Two one-dimensional projections of an originally two-dimensional dataset that contains two outliers. The PCA projection is deceived by the outliers, unlike the normalized PCA projection that maintains much of the structure of the data.

taking the dissimilarities as the inverse squared distances, $d_{ij} = 1/\mathrm{dist}_{ij}^2$. This weighting scheme results in the projection shown in Figure 2c. Now, the first principal component is what we have been aiming for in the first place. The second principal component also accounts much less for the outliers, and consequently shows much more clearly the fine structure of the main cluster.

The last example demonstrated a powerful property of weighted PCA. The choice of dissimilarities is completely up to the user, and can be specifically tailored for particular application. For example, an even more dramatic underweighting of outliers may be attained if we take the dissimilarities to be proportional to a decaying exponential function of the original pairwise distances.

Yet another enlightening example is shown in Figure 3, which draws two-dimensional projections of a portion of *Alpadyin's handwritten digits* dataset. This dataset, developed by Alpaydin and Kaynak [2] and publicly available in [3], consists of $\sim$380 64-dimensional samples of each of the ten digits. In the figure we show the drawings of the three digits 0,4 and 6. Figure 3a shows the projection obtained by PCA, from which we see that it was guided by the large intra-cluster variability of the numeral 4. Astonishingly, using the normalized PCA weighting scheme, see Figure 3b, we obtain clusters that are far more separated, even though we have not supplied to the algorithm any information about the clustering decomposition of the data. This occurs due to the fact that approximately the same set of axes efficiently captures the maximal fraction of both the intra-cluster and inter-cluster (weighted) variability.

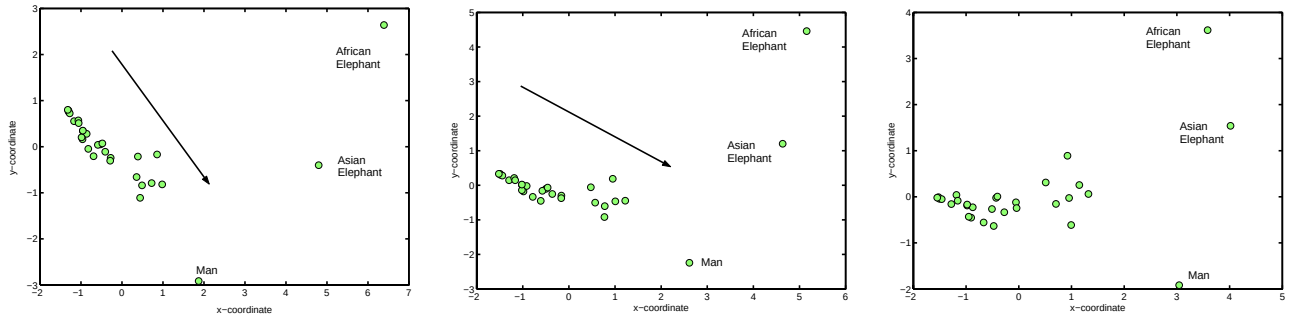


Fig. 2. Two-dimensional projections of the four-dimensional sleep dataset. (a) PCA. (b) Normalized PCA. (c) Weighted PCA, with the weights taken as the square of those in normalized PCA.

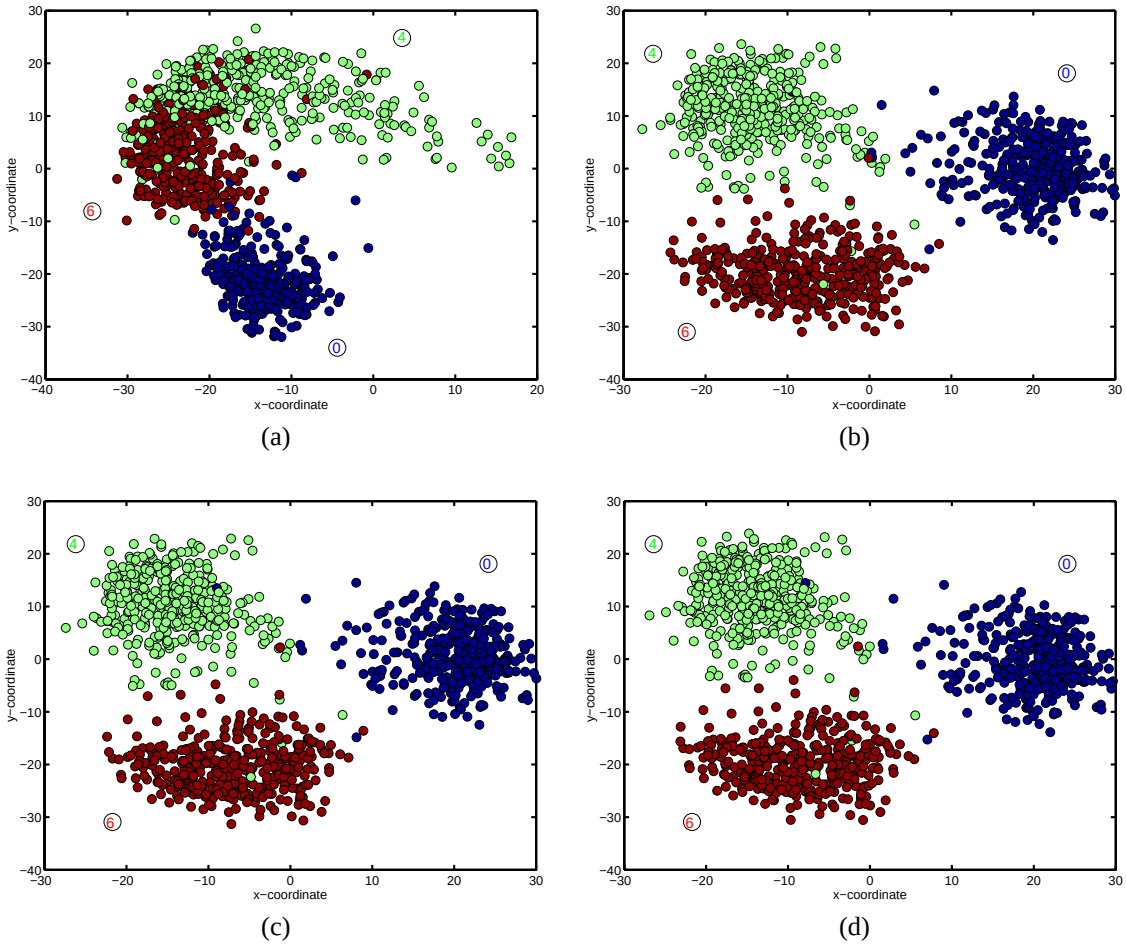


Fig. 3. Two-dimensional projections of the 64-dimensional Alpaydin's handwritten digits dataset. (a) PCA. (b) Normalized PCA. (c) Supervised PCA with binary weights. (d) Supervised PCA with normalized weights.

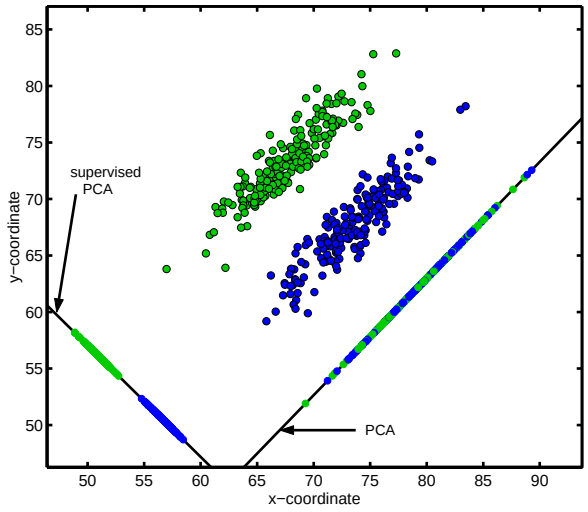


Fig. 4. Two one-dimensional projections of an originally two-dimensional synthetic dataset that contains two clusters. The PCA projection fuses the clusters, while the supervised PCA projection keeps them wide apart.

2) Supervised PCA: As the previous example shows, when the data are labeled, a projection is often required to emphasize the discrimination between the clusters. PCA, and even normalized PCA, may fail to accomplish this, no matter how easy the task is, as they are unsupervised techniques. The directions that maximize the scatter of the data might not be as adequate to discriminate between clusters.

Fortunately, our weighted PCA scheme can straightforwardly take into consideration data labeling. Assume that we are given any previously assigned pairwise dissimilarity values, for example calculated as in normalized PCA, or simply uniform constants. Then, we may artificially underweight the dissimilarities between intra-cluster pairs of data elements, thus instructing the projection that it is more important to show the inter-cluster separation. Technically, this is done by multiplying the intra-cluster dissimilarities by some decay factor $0 \leqslant t \leqslant 1$, obtaining the modified dissimilarities

$$d_{ij}^{\text{labeled}} = \begin{cases} t \cdot d_{ij} & i \text{ and } j \text{ have the same label} \\ d_{ij} & \text{otherwise.} \end{cases}$$

Typically, we use $t = 0$, which means that the internal structure of each cluster is set only indirectly according to the inter-cluster relationships of its members.

Figure 4 spectacularly shows the effect of supervised PCA. A two-dimensional synthetic dataset is drawn, comprising two normally-distributed clusters (200 points each), together with two one-dimensional projections. As is well apparent, the one-dimensional PCA projection completely merges the two clusters, whereas by setting all the intra-cluster dissimilarities to zero, we obtain a one-dimensional projection that clearly captures the dataset clustering decomposition.

Another fascinating example on a real-world dataset is shown in Figures 3c-d. Here, we refer again to Alpadyin's handwritten digits dataset, but this time use the supervised PCA scheme to achieve the drawings. Figure 3c used binary weights only, with a dissimilarity of one between any two samples not from within the same cluster. Figure 3d, which is almost undistinguishable from Figure 3c, uses normalized weights as in normalized PCA, with a decay factor $t = 0$. Our first conclusion is that binary weights give results which are essentially equivalent to those achieved by other, more sophisticated, weighting schemes. This is a general conclusion that we have also observed in other datasets that do not contain outlying clusters, with the immediate implication of saving computation time. The second and more surprising conclusion is that supervised PCA gives practically the same results as normalized PCA. This is definitely a feature of this particular dataset, and is not a general property. In some sense, the two methods are complete opposites. While normalized PCA tries to maintain the dissimilarities between similar samples, supervised PCA tries to maintain the dissimilarities between the dissimilar samples.

IV. Ratio Optimization

So far we have enriched the standard PCA by modifying the matrix A in the maximization problem (1). Further strengthening of the method can be achieved by changing the form of the target function itself and writing it as a ratio. This enables the insertion of richer objectives into the optimization, as maximizing a ratio reflects some compromise between maximizing the numerator and minimizing the denominator. Next, we shall see some powerful methods developed in accord with this idea.

A. Maximization of Weighted Pairwise Dissimilarities

By Corollary II.1, the weighted PCA can be formulated as

$$\max_{v^1,\ldots,v^p} \frac{\sum_{i<j} d_{ij} \left(\text{dist}_{ij}^p\right)^2}{\sum_{\alpha=1}^p (v^\alpha)^T v^\alpha}.$$

We suggest replacing it by altering the denominator,

$$\max_{v^1,\ldots,v^p} \frac{\sum_{i<j} d_{ij} \left(\text{dist}_{ij}^p\right)^2}{\sum_{\alpha=1}^p (v^\alpha)^T X^T X v^\alpha}. \tag{10}$$

Notice that $v^T X^T X v$ is the variance of the projection in direction v, and the denominator sums the variances along all axes. Therefore, while the numerator strives at maximizing weighted pairwise distances, as before, the denominator prevents "blowing up" of the result by minimizing the scatter of the data along the principal axes. This target function seems to be perfectly suitable for labeled data, where intra-cluster dissimilarities have been decayed. In this case we expect highly dissimilar data elements (belonging to different clusters) to be placed distantly to maximize the numerator. But data elements of the same cluster have (almost) no influence on the numerator so they are placed closely to minimize the denominator.

Obviously, a target function alone does not suffice for a proper dimensionality reduction scheme, as it should be accompanied by a precise definition of the relationships between the direction vectors. In the spirit of Section III, we may

require orthonormality relationships, resulting in the scheme

$$\max_{v^1,\ldots,v^p} \quad \frac{\sum_{i<j} d_{ij} \left(\text{dist}_{ij}^p\right)^2}{\sum_{\alpha=1}^p (v^\alpha)^T X^T X v^\alpha} \qquad (11)$$
$$\text{subject to: } (v^\alpha)^T v^\beta = \delta_{\alpha\beta}. \quad \alpha,\beta = 1,\ldots,p.$$

Such a projection scheme is in many ways the more natural way to work in multiple dimensions. In fact, the Foley-Sammon transformation [7] is an example of a scheme that can be put in such a form. However, we shall not dwell on this scheme for two reasons:

1) Notice that (11) digresses from the mathematical framework developed in this paper as it cannot be put in the form of (5).
2) In the realm of dimensionality reduction, there seems to be a subtle, yet profound, drawback of using (11). The orthonormality constraints enforce orthonormal direction vectors, but do not impose any orthogonality relationships between the coordinate vectors. This might result in highly correlated coordinate vectors, with the ramification that some of them are superfluous and do not add new information. Even worse, this behavior is actually the one we would expect in large datasets, since it is reasonable to be able to find two orthogonal direction vectors that show approximately the same projection of the data, both with maximal, or close to maximal, values of the target function.

Consequently, we suggest replacing the normality constraints on the direction vectors by such constraints on the coordinate vectors. As the columns of X are centered, so are the coordinate vectors. Orthogonality constraint on the latter, thus, means that they are *uncorrelated* (two centered vectors are uncorrelated when they are orthogonal) and consequently each axis conveys new information that does not exist in the rest of the axes. Formally, the coordinate vectors are $Xv^1,\ldots,Xv^p \in \mathbb{R}^n$, and they would be orthonormal if $(Xv^\alpha)^T Xv^\beta = \delta_{\alpha\beta}$ for $\alpha,\beta = 1,\ldots,p$. This is the same as writing $(v^\alpha)^T X^T X v^\beta = \delta_{\alpha\beta}$, which means that the direction vectors are required to be $X^T X$ orthonormal. Consequently, we may suggest the following dimensionality reduction scheme

$$\max_{v^1,\ldots,v^p} \quad \frac{\sum_{i<j} d_{ij} \left(\text{dist}_{ij}^p\right)^2}{\sum_{\alpha=1}^p (v^\alpha)^T X^T X v^\alpha} \qquad (12)$$
$$\text{subject to: } (v^\alpha)^T X^T X v^\beta = \delta_{\alpha\beta}. \quad \alpha,\beta = 1,\ldots,p.$$

From Lemma II.1, Theorem II.2 and Corollary II.1, the maximizer of this problem is the p highest generalized eigenvectors of $(X^T L^d X,\ X^T X)$.

A demonstration of this scheme is given in Figure 5. The dataset comprises hand-written digits taken from `www.cs.toronto.edu/~roweis/data.html`. It contains 39 samples per digit, with each sample being a 20×16 bitmap, resulting in $320 (= 20 \times 16)$ binary coordinates. In Figure 5a we see a two-dimensional embedding of this 320-dimensional dataset using the method we have described here. Inter-cluster dissimilarities were all set to 1, whereas intra-cluster dissimilarities were set to 0. The result reflects a good separation between clusters, especially in light of the comparison with PCA, whose two-dimensional projection is shown in Figure 5b.

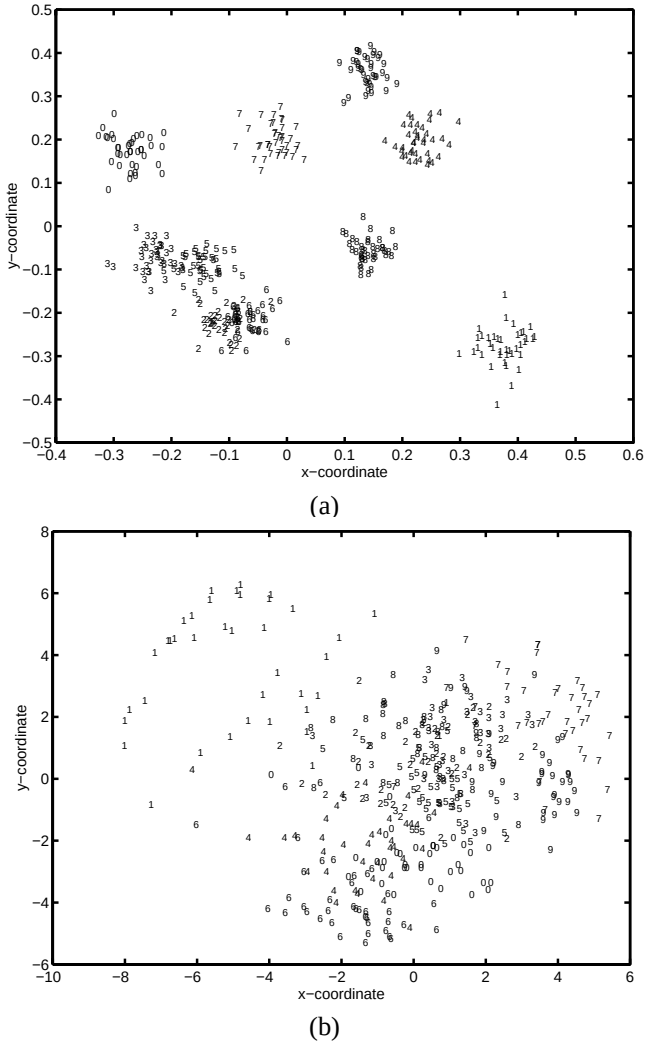
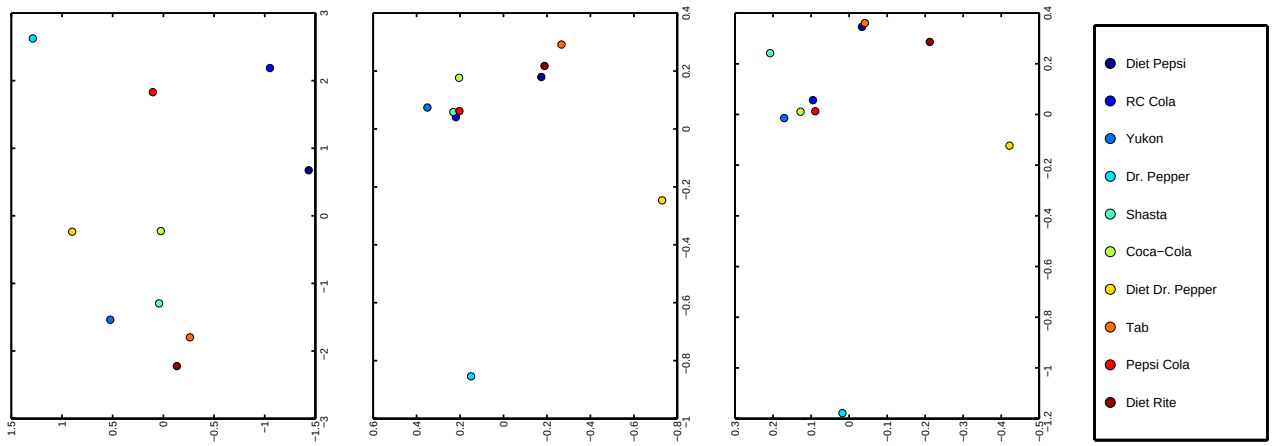


Fig. 5. Hand-written digits dataset containing 390 samples in 320 dimensions. **(a)** The two-dimensional embedding of our method, exhibiting good separation of clusters. **(b)** The two-dimensional embedding of PCA, exhibiting poor cluster discrimination.

Another example is shown in Figure 6, drawing two-dimensional embeddings of the Colas dataset, taken from [10]. These data were collected in an experiment aimed at comparing tastes of ten colas as perceived by a human panel. Subjects were asked to perform two tasks: to rate each individual cola with regard to 13 flavor descriptors, and to rank the level of dissimilarity between each pair of colas. At the end of the day, the resulting dataset comprises a 10×13 coordinate matrix[1], as well as a 10×10 matrix of pairwise dissimilarities. Figure 6a shows the two-dimensional projection computed by PCA. This embedding reflects well the difference in the coordinates of the ten colas, but cannot account for the measured dissimilarities. The embedding produced by a nonlinear dimensionality reduc-

[1] Actually we could show, using PCA, that the true dimensionality is seven, and therefore we have worked with the 10×7 coordinate matrix, obtained by taking the projection of the data onto the space spanned by the first seven principal components.



(a) (b) (c)

Fig. 6. Three two-dimensional embeddings of the colas dataset. Each of ten colas is characterized by 13 coordinates reflecting its flavor as assessed by human subjects. These subjects also produced pairwise dissimilarities between the different colas. **(a)** A PCA projection of the dataset accounting only for the coordinates. **(b)** The (nonlinear) eigenprojection embedding, accounting only for the dissimilarities. **(c)** Our method, taking into account both coordinates and dissimilarities.

tion technique, the *eigenprojection method* [8], [9], is shown in Figure 6b. This method is capable of accounting for the measured dissimilarities, but cannot consider the coordinates. In Figure 6c, we use our method with the weights d_{ij} being the measured dissimilarities, utilizing thereby all available information — coordinates and dissimilarities. Comparison of this embedding to the former two shows a clear resemblance to the nonlinear eigenprojection embedding, validating our success in incorporating the dissimilarities into the final result. However, unlike the eigenprojection, here the direction vectors are interpretable linear combinations of the original taste descriptors, indicating which of them influence the way people sense different colas.

B. Minimization of Weighted Pairwise Similarities

Working with ratios makes it feasible to handle pairwise weights given in the form of similarities. We define the *similarities* $\{s_{ij}\}_{i,j=1}^{n}$ as a set of symmetric non-negative weights, with s_{ij} measuring how important it is for us to place the data elements i and j close to each other. By convention, $s_{ij}=0$ for $i=j$. In analogy with (10) and (12), we can now define the complementary minimization problem,

$$\min_{v^1,\ldots,v^p} \frac{\sum_{i<j} s_{ij}\left(\text{dist}_{ij}^{p}\right)^2}{\sum_{\alpha=1}^{p}(v^\alpha)^T X^T X v^\alpha} \qquad (13)$$
$$\text{subject to: } (v^\alpha)^T X^T X v^\beta = \delta_{\alpha\beta}, \quad \alpha,\beta = 1,\ldots,p.$$

Here we strive to shorten the distance between highly similar data elements. It is important to keep in mind that when solving a minimization problem, the denominator is essential. Otherwise, we could have minimized the numerator of (13) by projecting the data along an uninteresting direction where they have almost no variability.

By defining the Laplacian L^s as

$$L_{ij}^{s} = \begin{cases} \sum_{j=1}^{n} s_{ij} & i=j \\ -s_{ij} & i\neq j \end{cases}$$

and using Lemma II.1 and Theorem II.2, we can easily see that (13) is solved by the p lowest generalized eigenvectors of $(X^T L^s X,\ X^T X)$.

Similarity values appear frequently in data analysis. Two simple ways of extracting them from the coordinates are by using decreasing functions of the distances or by computing correlation coefficients. Sometimes it is beneficial to zero low similarity values, thus obtaining a sparse Laplacian with non-zero entries only between close elements. In this case, it is sometimes advisable to set all these non-zero entries to the value 1, thus getting a binary similarity matrix.

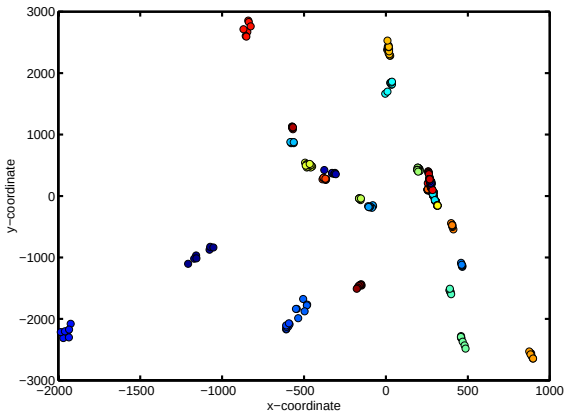
The similarity-based approach can also be used for labeled data. Here, we have to decay all the similarities between elements from different clusters, using some decay factor $0 \leqslant t \leqslant 1$,

$$s_{ij}^{\text{labeled}} = \begin{cases} s_{ij} & i \text{ and } j \text{ have the same label} \\ t\cdot s_{ij} & \text{otherwise} \end{cases}$$

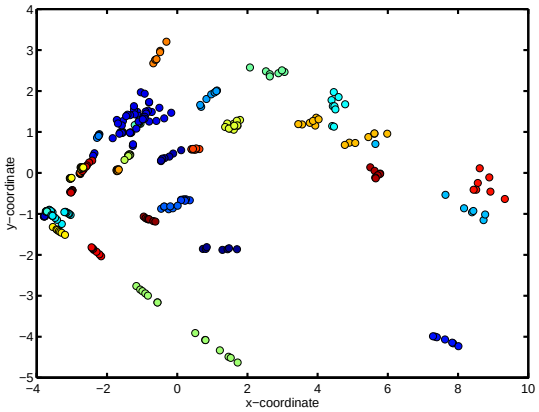
Typically, we set $t=0$, which means that we do not want the low dimensional embedding to reflect any proximity relations between elements from different clusters.

We cannot give a conclusive advice on whether to prefer working with similarities or with dissimilarities. In general, it depends on which kind of relationships is easier to be measured on the specific data.

An example in which working with similarities is convenient is the *odors* dataset, which comprises 30 volatile pure chemicals that were chosen to represent a broad range of smells. The odor emission of each sample was measured using an *electronic nose*, resulting in a 16-dimensional vector representing that sample. In total, we have performed 300 measurements to yield a 300×16 coordinate matrix, partitioned into 30 clusters. In a separate work [4], we have developed a technique to derive from the raw data pairwise similarity values. Figure 7a shows a two-dimensional embedding of this dataset using our method, where inter-cluster similarities were set to zero. In general, the clusters, which are color-coded in the figure, are well separated. For comparison, we show



(a)



(b)

Fig. 7. The odors dataset containing 300 measurements classified into 30 clusters, shown in color-coding in the figure. **(a)** The embedding computed by our method, clearly showing sharp separation between the clusters. **(b)** The projection computed by PCA.

in Figure 7b the two-dimensional projection of this dataset computed by PCA, clearly exhibiting a reduced ability to show the separation between the different clusters.

C. Inter-cluster repulsion and intra-cluster attraction

Lemma III.1 shows that we may replace the matrix $X^T X$ by the matrix $X^T L^u X$ in the denominators of problems (12) and (13). This suggests a generalization of these problems by using a general Laplacian, rather than L^u, in the denominator.

Thus, if we are given pairwise similarities s_{ij}, as well as pairwise dissimilarities d_{ij}, we may replace (13) by

$$\min_{v^1,\ldots,v^p} \frac{\sum_{i<j} s_{ij} \left(\text{dist}_{ij}^p\right)^2}{\sum_{i<j} d_{ij} \left(\text{dist}_{ij}^p\right)^2} \quad \text{subject to: } (v^\alpha)^T X^T L^d X v^\beta = \delta_{\alpha\beta} \quad \alpha,\beta = 1,\ldots,p, \tag{14}$$

whose solution is given by the p lowest generalized eigenvectors of $(X^T L^s X,\ X^T L^d X)$. Since $(v^\alpha)^T X^T L^d X v^\alpha = \sum_{i<j}^n d_{ij} \cdot ((Xv^\alpha)_i - (Xv^\alpha)_j)^2$, the constraint states that the weighted sum of squared distances should be uniform along all axes. Minimizing this target function is achieved by simultaneously minimizing the distances between highly similar elements (to minimize the numerator) and maximizing the distances between highly dissimilar elements (to maximize the denominator). For labeled data we may decay inter-cluster similarities and intra-cluster dissimilarities, usually setting them to zero. Consequently, in problem (14) we strive to minimize the weighted sum of intra-cluster squared distances while maximizing the weighted sum of inter-cluster squared distances.

Similarly, we can generalize (12) by

$$\max_{v^1,\ldots,v^p} \frac{\sum_{i<j} d_{ij} \left(\text{dist}_{ij}^p\right)^2}{\sum_{i<j} s_{ij} \left(\text{dist}_{ij}^p\right)^2} \quad \text{subject to: } (v^\alpha)^T X^T L^s X v^\beta = \delta_{\alpha\beta} \quad \alpha,\beta = 1,\ldots,p. \tag{15}$$

Here, the solution is given by the p highest generalized eigenvectors of $(X^T L^d X,\ X^T L^s X)$.

Problems (14) and (15) allow for more degrees of freedom than the previous methods discussed in this paper. They let us use pairwise weights not only in the target function that has to be maximized/minimized but also in the orthonormality constraint. Therefore, they are very suitable for labeled data, as they can induce "attraction" between elements of the same cluster, and "repulsion" between elements of different clusters. An important application of these methods is a robust form of Fisher's Linear Discriminant Analysis, to which we now turn.

D. Normalized LDA

We start by introducing some notations that are required for this subsection. Let c be the total number of clusters, and let n_i be the number of data elements in the ith cluster. We use the symbols μ_i and S_i for the mean vector and biased covariance matrix of the ith cluster. The matrix $S_w = \frac{1}{n}\sum_{i=1}^c n_i S_i$ is called the *average intra-cluster covariance matrix*. The *inter-cluster covariance matrix* is defined as $S_b = \frac{1}{n}\sum_{i=1}^c n_i \mu_i \mu_i^T$. For more details on these magnitudes see, e.g., [12].

It was Fisher [6] who first suggested maximizing what is now known as the *Fisher criterion* $(v^T S_b v)/(v^T S_w v)$, where v is some direction vector. Here, the ratio is again used for achieving a balance — to maximally separate between the clusters (the role of the numerator), and at the same time to keep the clusters as compact as possible (the role of the denominator). It can be proved that the maximizer of the Fisher criterion is the same as the maximizer of

$$\frac{v^T S_b v}{v^T S v}.$$

Since S and $X^T X$ are identical up to a constant (n), this last form of the Fisher criterion bears much resemblance to the scheme (12), implying a profound connection between them. To see this connection, we will show how the Fisher criterion can be rewritten in the form of (12). To show it explicitly, let us define the *inter-cluster Laplacian* L^b as

$$L_{ij}^b = \begin{cases} -1 + \frac{n}{n_g} & i,j \text{ are both in cluster } g \\ -1 & i,j \text{ are in different clusters.} \end{cases} \tag{16}$$

Then, we may prove the following equivalent of Lemma III.1.

Lemma IV.1 *The matrices $X^T L^b X$ and S_b are identical up to a positive multiplicative factor, $X^T L^b X = n^2 \cdot S_b$.*

Proof: Let us define the $n \times c$ matrix G such that G_{ij} is one when the ith data element belongs to the jth cluster, and zero otherwise. From Lemma III.1, the Laplacian associated with S_g (the biased covariance matrix of the gth cluster) is

$$L_{ij}^{S_g} = G_{ig}G_{jg}\left(\frac{1}{n_g}\delta_{ij} - \frac{1}{n_g^2}\right).$$

Since the clusters are disjoint, the Laplacian associated with S_w is just $L^{S_w} = \frac{1}{n}\sum_{g=1}^{c} n_g L^{S_g}$, namely

$$L_{ij}^{S_w} = \frac{1}{n}\sum_{g=1}^{c} G_{ig}G_{jg}\left(\delta_{ij} - \frac{1}{n_g}\right).$$

Based on the relation $S_b = S - S_w$, and using again Lemma III.1, the Laplacian associated with S_b can be written as

$$L_{ij}^{S_b} = \frac{1}{n^2}L_{ij}^{u} - L_{ij}^{S_w} = \frac{1}{n}\delta_{ij} - \frac{1}{n^2} - \frac{1}{n}\sum_{g=1}^{c} G_{ig}G_{jg}\left(\delta_{ij} - \frac{1}{n_g}\right).$$

But, $\sum_g G_{ig}G_{jg}\delta_{ij} = \delta_{ij}$, and therefore

$$L_{ij}^{S_b} = -\frac{1}{n^2} + \frac{1}{n}\sum_{g=1}^{c}\frac{1}{n_g}G_{ig}G_{jg}.$$

Multiplying it by n^2, we immediately obtain (16). ∎

The Fisher criterion can therefore be written as $(v^T X^T L^b X v)/(v^T X^T L^u X v)$.

Fisher, however, was interested only in finding a single direction vector. Later, several alternatives were suggested how to extend his idea for finding a series of direction vectors. The most popular technique is known as *Fisher's linear discriminant analysis* (LDA), which poses the following maximization problem

$$\begin{aligned} &\max_{v^1,\ldots,v^p} \quad \frac{\sum_{\alpha=1}^{p}(v^\alpha)^T S_b v^\alpha}{\sum_{\alpha=1}^{p}(v^\alpha)^T S v^\alpha} \\ &\text{subject to: } (v^\alpha)^T S v^\beta = \delta_{\alpha\beta} \quad \alpha,\beta = 1,\ldots,p. \end{aligned} \tag{17}$$

This is a classical problem, and the reader is referred to, e.g., [5], [12] for more details. Despite its usefulness, we can point out two drawbacks of LDA:

1) A simple maximization of the inter-cluster variance is sensitive to outliers, reflected in the tendency of LDA to prefer showing a few remotely located clusters at the expense of masking closer clusters. To illuminate this point we have generated a synthetic two-dimensional dataset, shown in Figure 8, comprising 10 clusters, each with 100 elements. Two of the clusters are placed distantly from the rest. As can be seen, the one-dimensional projection computed by LDA emphasizes the two outlying clusters, but completely masks the other eight clusters, which might be the more fundamental portion of the data.
2) When used for visualization, the attempt to minimize the variance of a cluster does not take into consideration the shape and size of this cluster. No matter if the cluster is dense or heterogeneous, or if the cluster is elongated or spherical, LDA strives to embed it as a small sphere. This may be desired for classification, but prevents a reliable visual assessment of the cluster properties. Again, we shall make this claim more tangible by showing a synthetic example. Figure 9 shows a two-dimensional dataset comprising two normally-distributed clusters, each with 200 elements. One cluster is symmetric having the same variance along both axes, whereas the other cluster is elliptic, and its variance along the x-axis is 10 times larger than the variance along the y-axis. As can be seen in the figure, one-dimensional projection computed by the LDA makes the two clusters of approximately the same scatter. Consequently, the heterogeneity of the elliptic cluster cannot be discerned.

In order to overcome these drawbacks, we may be aided by our previous observation that the scheme (12) (and of course (15)), is a generalization of LDA. LDA is restored if we choose the dissimilarities in (12) as dictated by (16). Similarly to our proof of Theorem III.1, it can be shown that LDA strives to maximize the ratio between inter-cluster pairwise squared distances and intra-cluster pairwise squared distances. Consequently, it is mainly concerned with the larger pairwise distances. We claim that we can remedy the two aforementioned shortcomings of LDA by an appropriate choice of the pairwise weights in (14) or (15) which reduce the dominance of large distances. Here we would like to suggest a particular weighting scheme, which we call *normalized LDA*,

$$d_{ij} = \begin{cases} 0 & i \text{ and } j \text{ have the same label} \\ \frac{1}{\text{dist}_{ij}} & \text{otherwise} \end{cases}$$

$$s_{ij} = \begin{cases} \frac{1}{\text{dist}_{ij}} & i \text{ and } j \text{ have the same label} \\ 0 & \text{otherwise} \end{cases}$$

Normalized LDA is far more robust with respect to a few outlying clusters, corresponding to large distances from the rest of the data. Such distances will have smaller impact as their weights (the respective values of the d_{ij}'s) are reduced. This is beautifully demonstrated in Figure 8 where the one-dimensional projection of normalized LDA captures well the eight clusters in a row, reflecting the main trend in the data. Similarly, it is not very important for normalized LDA to place distant elements of the same cluster in close proximity, as their respective weights (the respective values of the s_{ij}'s) are small. This can be seen in the normalized LDA one-dimensional projection in Figure 9, where the different structure of the clusters is preserved, without ruining their separation.

V. Conclusions

We propose a novel family of linear transformations to achieve low dimensional embedding of multivariate data. These transformations have a significant advantage over other techniques in their ability to simultaneously account for many properties of the data such as coordinates, pairwise similarities, pairwise dissimilarities, and their clustering decomposition. Therefore, we exhaust all kinds of available information so as to make an instructive and reliable low dimensional embedding. In fact, the derivation of these transformations integrates two apparently very different approaches — those that are coordinate-based and those that are weight-based. This reveals interesting relationships between the linear PCA and LDA and the nonlinear eigenprojection and MDS.

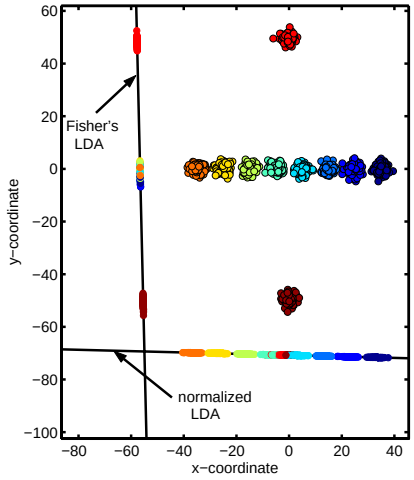


Fig. 8. Two one-dimensional projections of two-dimensional data composed of ten clusters, two of them are outliers. The LDA projection, striving to maximize the inter-cluster variance, emphasizes only the outlying clusters. However, the normalized LDA separates those eight clusters that form the main trend of the data.

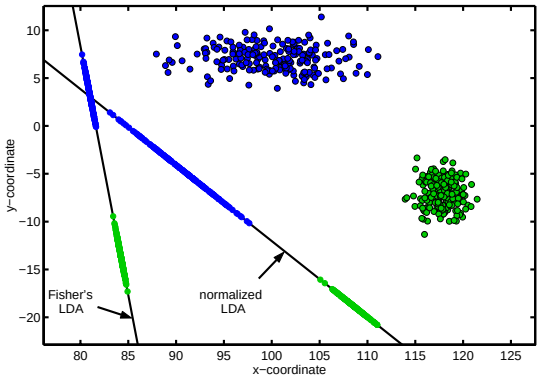


Fig. 9. Two one-dimensional projections of two-dimensional data composed of two clusters of very different shapes. The LDA projection, striving to diminish the intra-cluster variance, produces very similar projections for both clusters. However, the normalized LDA succeeds in showing the different intra-structure of the clusters.

Our methods contain PCA and LDA as special cases, but offer more powerful and robust variants that can better capture the essence of the data under inspection. Such two interesting variants, which address several shortcomings of PCA and LDA, are normalized PCA and normalized LDA. One of their advantages is an improved robustness towards the presence of outliers, samples or clusters, in the data.

All formulations lead to optimal solutions that can be directly computed by eigenvector decomposition of $m \times m$ matrices, where m is the dimensionality of the data. This is also the case in PCA and LDA. However, the power of our formulations lies in the fact that these $m \times m$ matrices are derived by matrix multiplications that involve an $n \times n$ Laplacian matrix, where n is the number of data elements (typically, $n >> m$). Therefore, we fine-tune the $m \times m$ matrix by appropriately altering the $n \times n$ entries of the Laplacian, and so the pairwise relationships between data elements are directly reflected in the $m \times m$ matrix.

One of the most important properties of our methods is that they can adequately address labeled data by capturing well the inter-cluster structure of the data, as well as the cluster shapes. This is naturally highly beneficial when we are interested in data exploration.

ACKNOWLEDGEMENT

We would like to thank UCI repository of machine learning databases [3] for the use of several of their public datasets.

REFERENCES

[1] T. Allison and D. Cicchetti, "Sleep in Mammals: Ecological and Constitutional Correlates", *Science* **194** (1976) 732–734.
[2] E. Alpaydin and C. Kaynak, "Cascading Classifiers", *Kybernetika* **34** (1998) 369–374.
[3] C. L. Blake and C. J. Merz (1998), UCI Repository of machine learning databases [`http://www.ics.uci.edu/ mlearn/MLRepository.html`]. Irvine, CA: University of California, Department of Information and Computer Science.
[4] L. Carmel, Y. Koren and D. Harel, "Visualizing and Classifying Odors Using a Similarity Matrix", *Proc. 9th International Symposium on Olfaction and Electronic Nose (ISOEN'02)*, Aracne, pp. 141–146, 2003.
[5] B. S. Everitt and G. Dunn, *Applied Multivariate Data Analysis*, Arnold, 1991.
[6] R. A. Fisher, "The use of multiple measurements in taxonomic problems", *Ann. Eugen.* **7** (1936), 179–188.
[7] D. H. Foley and J. W. Sammon Jr., "An optimal set of discriminant vectors", *IEEE Transactions on Computers* **c-24** (1975), 281–289.
[8] K. M. Hall, "An *r*-dimensional Quadratic Placement Algorithm", *Management Science* **17** (1970), 219–229.
[9] Y. Koren, L. Carmel and D. Harel, "ACE: A Fast Multiscale Eigenvectors Computation for Drawing Huge Graphs", *Proc. IEEE Information Visualization (InfoVis'02)*, IEEE, pp. 137–144, 2002.
[10] S. S. Schiffman, M. L. Reynolds and F. W. Young, *Introduction to Multidimensional Scaling: Theory, Methods and Applications*, Academic Press, 1981.
[11] StatLib dataset index, Carnegie Mellon University, `http://lib.stat.cmu.edu`. Dataset *sleep*, submitted by Roger Johnson.
[12] A. R. Webb, *Statistical Pattern Recognition*, John Wiley and Sons, 2002.

Yehuda Koren received his BA degree in computer science from the Open University, Israel, in 1997, and his MSc degree from the Weizmann Institute of Science, Israel, in 1999. He completed his PhD in the Department of Computer Science and Applied Mathematics of the Weizmann Institute of Science. Currently, he works in AT&T Labs – Research. Among his primary research interests are algorithms for drawing large graphs, data analysis and visualization, multiscale optimization, and clustering algorithms.

Liran Carmel received his BSc in physics from Tel-Aviv University, Israel, in 1991, and his MSc degree in physics from the Technion — Israel Institute of Technology, in 1998. He completed his PhD in the Department of Computer Science and Applied Mathematics at the Weizmann Institute of Science, Israel. Currently, he works in the National Center for Biotechnology Information in the National Institutes of Health. His research deals with materializing odor digitization, transmission and reproduction, and it involves the development of many kind of data visualization and classification algorithms.