DIMENSIONALITY REDUCTION

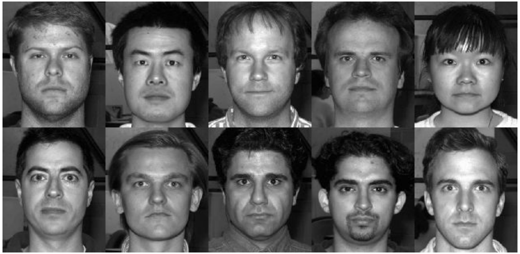
OVERVIEW

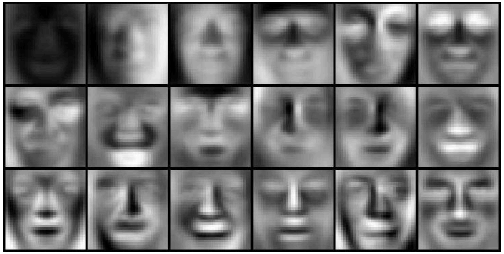
For many machine learning applications, a major issue can arise when performing recognition in high-dimensional spaces. This issue is known as the curse of dimensionality: we have too many parameter degrees of freedom for our model and, potentially, not enough observations to learn those parameters.

To deal with the curse of dimensionality, we can map the observations to a lower-dimensional subspace. In particular, we seek a function $f : \mathcal{R}^m \mapsto \mathcal{R}^k$, with $k \ll m$, such that the characteristics of the original observations in $\mathcal{R}^m$ are maintained in $\mathcal{R}^k$. This function performs dimensionality reduction on the observations.

DIMENSIONALITY REDUCTION

MOTIVATION: FACE RECOGNITION EXAMPLE



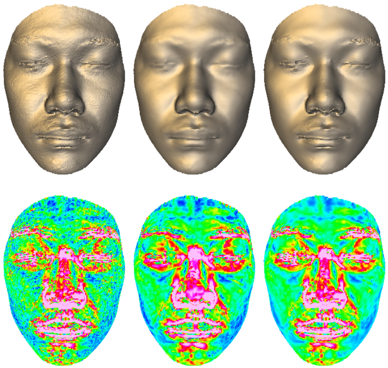


Suppose that we wanted to build a system to recognize faces in images. One option would be to unwrap each image, which is of size $\mathcal{R}^{n\times m}$, into a column vector, of size $\mathcal{R}^{nm\times 1}$. A simple distance test could be used to determine which face from a database best matches the query face. More advanced classification methods could also be applied.

As the size of the input image grows, there may be redundant information. We want to remove those unnecessary details so that our classification scheme focuses only on informative features of the face.

DIMENSIONALITY REDUCTION

MOTIVATION: DENOISING EXAMPLE



Suppose that we had a series of noisy, three-dimensional facial scans. We would like a way to remove the noise from these scans, while preserving important features of the face for further analysis.

One way to do this would be to decompose the facial scan into patches. For each patch, we find an orthogonal basis. Important facial features would have a high span along multiple bases. Any noise would, ideally, have a low span along the remaining bases. We could ignore those dimensions with low span then use the data from the remaining dimensions to construct a denoised facial scan.

PRINCIPAL COMPONENTS ANALYSIS

OVERVIEW

There are a variety of linear and non-linear techniques for performing dimensionality reduction. One of the most classical approaches is principal components analysis (PCA), which is a linear method. Non-linear extensions of principal components analysis also exist.

The goal of principal components analysis is to maximize the retained variance of the original observations when performing any reduction. Likewise, it attempts to minimize the least-squares reconstruction error between the reduced observations and the original, higher-dimensional observations. Both objectives attempt to best preserve as much information as possible about the original observations.

PRINCIPAL COMPONENTS ANALYSIS

OVERVIEW

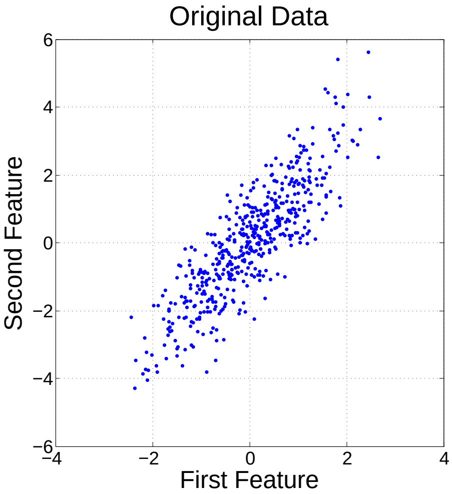
Expressed mathematically, principal components analysis maps an observation matrix $X \in \mathcal{R}^{n \times m}$, with n being the number of observations and m being the number of feature dimensions, to a reduced observation matrix $Y \in \mathcal{R}^{n \times k}$, where $k \ll m$ is the dimensionality of the subspace. It does this via a projection matrix $P \in \mathcal{R}^{m \times k}$:

$$Y = XP \mapsto \begin{bmatrix} y_{1,1} & \cdots & y_{1,k} \\ y_{2,1} & \cdots & y_{2,k} \\ \vdots & \ddots & \vdots \\ y_{n,1} & \cdots & y_{n,k} \end{bmatrix} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,m} \\ x_{2,1} & \cdots & x_{2,m} \\ \vdots & \ddots & \vdots \\ x_{n,1} & \cdots & x_{n,m} \end{bmatrix} \begin{bmatrix} p_{1,1} & \cdots & p_{1,k} \\ p_{2,1} & \cdots & p_{2,k} \\ \vdots & \ddots & \vdots \\ p_{m,1} & \cdots & p_{m,k} \end{bmatrix}.$$

Each column of the projection matrix is a principal component: these are unit vectors that yield orthogonal directions. These basis vectors maximize the variance of X in orthogonal directions with respect to each other. Hence, the variance decreases from $p_{1:m,1}$ to $p_{1:m,k}$.

PRINCIPAL COMPONENTS ANALYSIS

ILLUSTRATIVE EXAMPLE

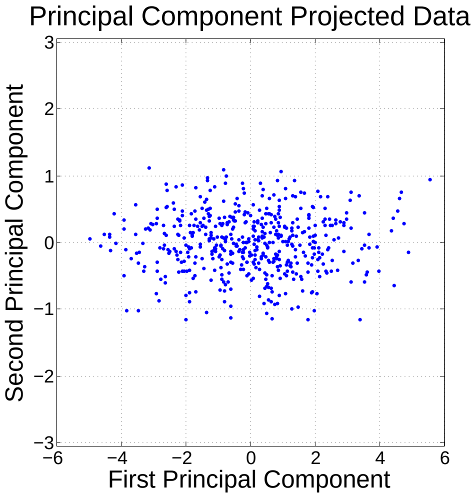


For what is to come, we first consider an illustrative example. Suppose that we wish to reduce a two-dimensional dataset of Gaussian-distributed samples to a one-dimensional representation. Obviously, there will be some loss of information when performing this reduction, since both feature dimensions contain non-negligible variance.

The first step of principal component analysis is to find the orthogonal unit vectors. These vectors will point in the directions with the largest variance.

PRINCIPAL COMPONENTS ANALYSIS

ILLUSTRATIVE EXAMPLE

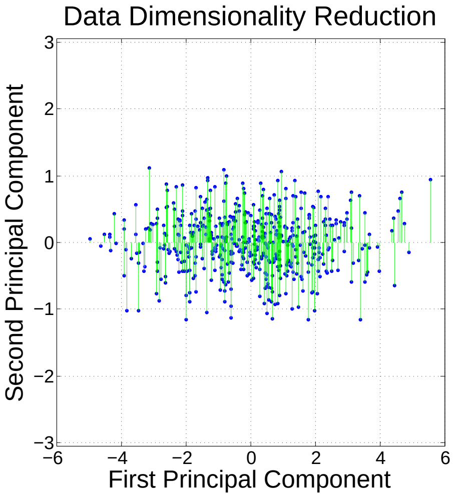


Once we have found the principal components, we can project onto those vectors. For this dataset, the projection entails rotating the original samples so that the direction with maximal variance is parallel to the abscissa (first principal component). The direction with the next maximal variance is parallel to the ordinate (second principal component).

Notice that no data reduction has taken place yet. We still have a two-dimensional dataset.

PRINCIPAL COMPONENTS ANALYSIS

ILLUSTRATIVE EXAMPLE

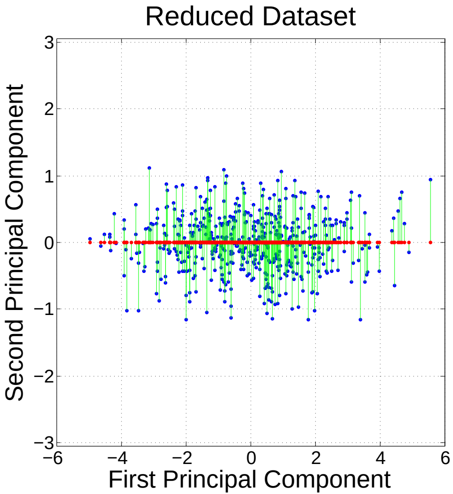


We can now project the data onto one of the axes to remove the other dimension. Principal component analysis keeps the dimensions with the largest data variance. Since the abscissa (first principal component) has more variance than the ordinate (second principal component), we will discard the second feature.

If we decided to discard the first principal component dimension, the least-squares reconstruction error would be lower than in the case where we discarded the second principal component dimension.

PRINCIPAL COMPONENTS ANALYSIS

ILLUSTRATIVE EXAMPLE



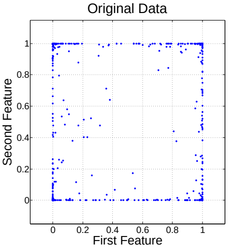
Projecting the data onto the first principal component axis essentially involves zeroing the contribution of the second principal component on the samples. Notice that the reduced representation still captures the distribution of the data along the first principal component.

When working with high-dimensional datasets, it may be possible to remove a great number of dimensions without impacting the reconstruction error. For smaller-dimensional datasets, a higher proportion of dimensions may contain useful feature information.

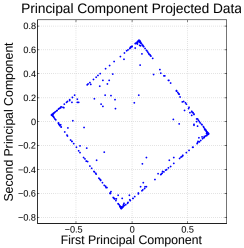
PRINCIPAL COMPONENTS ANALYSIS

CAVEAT

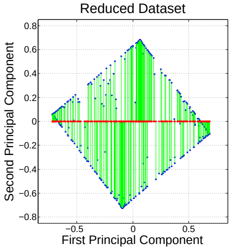
Original Data



Principal Component Projected Data



Reduced Dataset



Principal components analysis assumes approximate normality of the observation distribution. It may fail to produce a good low-dimensional projection for non-normal distributions. It may also fail to produce a good reduced dataset for observations that lie on a complicated manifold.

For such datasets, it will be helpful to consider more advanced, non-linear techniques for dimensionality reduction. Such methods include: kernel PCA (kPCA), isometric mapping (ISOMAP), local linear embedding (LLE), and Laplacian eigenmaps (LE).

PRINCIPAL COMPONENTS ANALYSIS

SINGULAR VALUE DERIVATION

There are multiple ways to derive the principal components. The first approach is based on singular value decomposition. Singular value decomposition is an extension of eigendecomposition for potentially non-square matrices. The two are equivalent when applied to square, symmetric, positive semi-definite matrices.

The idea behind singular value decomposition is to factor a matrix $X \in \mathcal{R}^{n \times m}$ into a series of matrices: $X = U \Sigma V^\top$. The matrix $\Sigma \in \mathcal{R}^{n \times m}$ is a diagonal matrix, which contains the singular values. The matrices $U \in \mathcal{R}^{n \times n}$ and $V \in \mathcal{R}^{m \times m}$ are orthonormal bases for the column and row spaces of X, respectively. A column space is the vector space made up of all linear combinations of the columns of a given matrix. A row space is the vector space made up of all linear combinations of the rows of a given matrix.

PRINCIPAL COMPONENTS ANALYSIS

SINGULAR VALUE DERIVATION

Another way of explaining singular value decomposition is that we wish to find a mapping between bases in the column space and row space: $\sigma_i u_i = X v_i$, $i = 1, \ldots, m$. The σ_i's can be understood as `stretch factors' that help match the u_i's with the v_i's.

We can re-write this equation in a matrix-based format as $U\Sigma = XV$. When solving this equation for X, we find that $X = U\Sigma V^{-1}$. Due to the properties of unitary matrices, inverses and transposes are equivalent, so $X = U\Sigma V^{\top}$.

PRINCIPAL COMPONENTS ANALYSIS

SINGULAR VALUE DERIVATION

Ideally, we would like to use singular value decomposition on some form of $X \in \mathcal{R}^{n \times m}$, since it yields orthogonal bases. Such bases would be taken as the principal components.

Since principal components analysis is trying to maximize the variance of the observations, we will consider applying singular value decomposition to the scatter of X. The scatter of X is defined as $X^\top X \in \mathcal{R}^{m \times m}$. We assume that X has zero mean and unit standard deviation before computing the scatter. The scatter matrix singular value decomposition is given by

$$X^\top X = (U \Sigma V^\top)^\top (U \Sigma V^\top) = V \Sigma^\top U^\top U \Sigma V^\top = V \Sigma^2 V^\top.$$

The U's cancel out, since $U^\top U = I$ for unitary matrices. We only have a single type of orthonormal basis in this expression, $V \in \mathcal{R}^{m \times m}$, which we take to be the principal components. The singular values Σ^2 give the observation span along each direction.

PRINCIPAL COMPONENTS ANALYSIS

SINGULAR VALUE INTERPRETATION

It can be shown that the singular value decomposition of $X^\top X$ can be equivalent to finding the eigendecomposition of $X^\top X$. In either case, we arrive at an orthonormal space that defines the basis for that space. When using singular value decomposition, this basis is defined by the right-singular vectors. When using eigendecomposition, the basis is given by the m linearly independent eigenvectors. In either case, we are *projecting the observations onto each of these basis directions*. If X does not have zero mean, there is a possibility that this basis will be shifted away from the origin and rotated.

The singular values or the square roots of the eigenvalues become the scale along each direction given by the identity matrix. These values are always ordered by magnitude, or power. Dimensions with larger variance appear in the top-left corner of Σ^2, while dimensions with lower variance appear in the bottom-right corner.

PRINCIPAL COMPONENTS ANALYSIS

SINGULAR VALUE INTERPRETATION

Due to this natural ordering of variance by power, we have an easy way to reduce observation dimensionality. We simply stop the projection at a certain eigenvalue by putting the remaining eigenvalues to zero. This effectively projects the observations onto a lower-dimensional space that preserves the maximal power for that number of dimensions. Obviously, including more dimensions will preserve more power.

Thus far, we have considered static time observations. In the case of time-series observations, principal component analysis will have an effect similar to low-pass filtering. That is, high-frequency components of the signal will be removed.

PRINCIPAL COMPONENTS ANALYSIS

SINGULAR VALUE DERIVATION

We have provided a means of finding an m-dimensional basis from $X^\top X \in \mathcal{R}^m$. It will be computationally efficient to find such a basis whenever the number of observations is much greater than the number of features.

Whenever we have many more features than observations, we will run into computational issues. We can instead find an n-dimensional basis from $XX^\top \in \mathcal{R}^n$. In this situation, the basis is given by the n linearly independent eigenvectors of $XX^\top$. Equivalently, they are given by the left-singular vectors of $XX^\top$ when using the singular value decomposition. Again, the square roots of the eigenvalues or singular values are ordered by magnitude.

PRINCIPAL COMPONENTS ANALYSIS

SINGULAR VALUE DERIVATION

There is a relationship between the eigenvectors and eigenvalues of $X^\top X \in \mathcal{R}^m$ and $XX^\top \in \mathcal{R}^n$.

In the case of $X^\top X$, we have that: $X^\top X v_i = \alpha_i v_i$, where $\alpha_i \in \mathcal{R}$ and $v_i \in \mathcal{R}^m$. Pre-multiplying by X, we find that $XX^\top X v_i = \alpha_i X v_i$. If we denote $XX^\top$ by C, we get $CXv_i = \alpha_i X v_i$ or $C\mu_i = \alpha_i \mu_i$, where $\mu_i = Xv_i$. We therefore see that the m eigenvalues of $X^\top X$ correspond to the top m eigenvalues of $XX^\top$. The eigenvectors are related by $\mu_i = Xv_i$.

PRINCIPAL COMPONENTS ANALYSIS

ALGORITHMIC OVERVIEW

The singular-value-decomposition-based approach to principal components analysis can be stated as follows:

Step 1: Subtract the mean response for each dimension of X from X.

Step 2: Compute the scatter matrix $X^\top X \in \mathcal{R}^{m\times m}$; can also compute another form of the scatter matrix $XX^\top \in \mathcal{R}^{n\times n}$.

Step 3: Perform a singular value decomposition or eigendecomposition of either $X^\top X$ or $XX^\top$. The principal components are taken as the eigenvectors

We can construct a reduced-dimensional representation as follows:

Step 4: Keep only the top k eigenvalues and eigenvectors $V \in \mathcal{R}^{m\times k}$. The reduced representation is given by $XV \in \mathcal{R}^{n\times k}$.

PRINCIPAL COMPONENTS ANALYSIS

LAGRANGIAN DERIVATION: FIRST PRINCIPAL COMPONENT

The second approach that we consider is based on Lagrange multiplier theory. We will iteratively derive each principal component, starting with $p_{1:m,1}$, which is the first column vector of the projection matrix. The higher-order principal components, $p_{1:m,2}$, $p_{1:m,3}$, etc., can be derived in a similar fashion.

Our aim is to find, for $x \in \mathcal{R}^m$,

$$p_{1:m,1}^\top x = \sum_{j=1}^{m} p_{j,1} x_j \text{ with } \max \operatorname{var}(p_{1:m,1}^\top x) = p_{1:m,1}^\top \Sigma p_{1:m,1}.$$

Here $\Sigma \in \mathcal{R}^{m \times m}$ is the symmetric, positive semi-definite covariance matrix. It measures the amount of change between multi-dimensional random variables. When the variables display opposite value behaviors (i.e., one variable increases while the other decreases), the covariance will be negative. When the variables display similar behaviors (i.e., both variables increase and decrease together), the covariance will be positive. Non-correlated variables have zero covariance.

PRINCIPAL COMPONENTS ANALYSIS

LAGRANGIAN DERIVATION: FIRST PRINCIPAL COMPONENT

When dealing with a finite number of observations, we may not have access to the true covariance matrix. We will instead consider the sample covariance matrix, which is an unbiased estimator of the true covariance matrix. The sample covariance $\Sigma \in \mathcal{R}^{m \times m}$ for an observation matrix $X \in \mathcal{R}^{n \times m}$ is given by

$$\frac{1}{n-1} \sum_{i=1}^{n} \left(x_i - \frac{1}{n} \sum_{j=1}^{n} x_j \right) \left(x_i - \frac{1}{n} \sum_{j=1}^{n} x_j \right)^\top .$$

This matrix defines the scatter of the observations in each feature dimension.

PRINCIPAL COMPONENTS ANALYSIS

LAGRANGIAN DERIVATION: FIRST PRINCIPAL COMPONENT

Without constraints, we could simply pick a very large $p_{1:m,1}$ to ensure that the variance is maximized. We assume, however, that there is a normalization constraint, $p^\top_{1:m,1} p_{1:m,1} = 1$, which ensures that $p_{1:m,1}$ is a unit-length vector.

To maximize $p^\top_{1:m,1} \Sigma p_{1:m,1}$ subject to $p^\top_{1:m,1} p_{1:m,1} = 1$, we use the technique of Lagrange multipliers. We apply this technique by introducing a multiplier $\lambda_1 \in \mathcal{R}$ as follows

$$\max_{p_{1:m,1}} \left(p^\top_{1:m,1} \Sigma p_{1:m,1} - \lambda_1 (p^\top_{1:m,1} p_{1:m,1} - 1) \right).$$

The maximization with respect to $p_{1:m,1}$ can be performed by differentiating with respect to $p_{1:m,1}$. The value of the multiplier at the solution of the problem is equal to the rate of change in the maximal value of the objective function as the constraint is relaxed.

PRINCIPAL COMPONENTS ANALYSIS

LAGRANGIAN DERIVATION: FIRST PRINCIPAL COMPONENT

The differentiation with respect to $p_{1:m,1}$ and setting this expression equal to zero results in

$$\frac{d}{dp_{1:m,1}}\left(p_{1:m,1}^{\top}\Sigma p_{1:m,1} - \lambda_1(p_{1:m,1}^{\top}p_{1:m,1} - 1)\right) = 0.$$

We find that this expression is equivalent to $\Sigma p_{1:m,1} - \lambda_1 p_{1:m,1} = 0$. After moving $\lambda p_{1:m,1}$ to the right-hand side, we find that $\Sigma p_{1:m,1} = \lambda_1 p_{1:m,1}$. This is nothing more than the eigenvalue equation: $p_{1:m,1}$ is an eigenvector and λ_1 is the corresponding largest eigenvalue. This eigenvalue captures the variance of $p_{1:m,1}^{\top}x$: $\text{var}(p_{1:m,1}^{\top}x) = \lambda_1$.

PRINCIPAL COMPONENTS ANALYSIS

LAGRANGIAN DERIVATION: SECOND PRINCIPAL COMPONENT

As with the first principal component, we would like to find a $p_{1:m,2}^\top x$ that maximizes $p_{1:m,2}^\top \Sigma p_{1:m,2}$. To avoid repeatedly choosing the same principal component, $p_{1:m,1}$, when performing this maximization, we want $p_{1:m,2}^\top x$ to be uncorrelated with $p_{1:m,1}^\top x$, for $x \in \mathcal{R}^m$. This constraint also ensures that the principal components are orthogonal and hence form a basis.

The uncorrelation constraint can be represented in a variety of ways. Recall that uncorrelated variables have zero covariance. This suggests that $\text{cov}(p_{1:m,1}^\top x, p_{1:m,2}^\top x) = 0$ would be the proper constraint to add when maximizing $p_{1:m,2}^\top \Sigma p_{1:m,2}$. It can be seen that the covariance is equivalent to $p_{1:m,1}^\top \Sigma p_{1:m,2}$ and also $p_{1:m,2}^\top \Sigma p_{1:m,1}$. We can simplify these expressions by using the eigenvalue from the first principal component: $p_{1:m,1}^\top \Sigma p_{1:m,2} = \lambda_1 p_{1:m,1}^\top p_{1:m,2}$, which implies that the corresponding constraint becomes $\lambda_1 p_{1:m,1}^\top p_{1:m,2} = 0$.

PRINCIPAL COMPONENTS ANALYSIS

LAGRANGIAN DERIVATION: SECOND PRINCIPAL COMPONENT

We again use the method of Lagrange multipliers to introduce this constraint directly into the maximization problem. This converts the constrained optimization into an unconstrained optimization problem:

$$\max_{p_{1:m,2}}\left(p_{1:m,2}^{\top}\Sigma p_{1:m,2}-\lambda_2(p_{1:m,2}^{\top}p_{1:m,2}-1)-\gamma_2\lambda_1 p_{1:m,1}^{\top}p_{1:m,2}\right).$$

Here, $\lambda_2 \in \mathcal{R}$ is a Lagrange multiplier to handle the unit normalization constraint. We have included a second multiplier, $\gamma_2 \in \mathcal{R}$, to represent the zero covariance constraint for the two principal components.

PRINCIPAL COMPONENTS ANALYSIS

LAGRANGIAN DERIVATION: SECOND PRINCIPAL COMPONENT

The differentiation with respect to $p_{1:m,2}$ and setting this expression equal to zero results in

$$\frac{d}{dp_{1:m,2}}\left(p_{1:m,2}^{\top}\Sigma p_{1:m,2}-\lambda_2(p_{1:m,2}^{\top}p_{1:m,2}-1)-\gamma_2\lambda_1 p_{1:m,1}^{\top}p_{1:m,2}\right)=0.$$

We find that this expression is equivalent to $\Sigma p_{1:m,2}-\lambda_2 p_{1:m,2}-\gamma_2\lambda_1 p_{1:m,1}=0.$

If we left-multiply $p_{1:m,1}$ into $\Sigma p_{1:m,2}-\lambda_2 p_{1:m,2}-\gamma_2\lambda_1 p_{1:m,1}=0$, we find that

$$\left(p_{1:m,1}^{\top}\Sigma p_{1:m,2}-\lambda_2 p_{1:m,1}^{\top}p_{1:m,2}-\gamma_2\lambda_1 p_{1:m,1}^{\top}p_{1:m,1}\right)=0.$$

The first term in this equation is zero, since we assume that $\mathrm{cov}(p_{1:m,1}^{\top}x, p_{1:m,2}^{\top}x)=p_{1:m,1}^{\top}\Sigma p_{1:m,2}=0$. The second term is zero for a similar reason. Due to the normalization constraint, $p_{1:m,1}^{\top}p_{1:m,1}=1$. These results imply that $\gamma_2=0$.

PRINCIPAL COMPONENTS ANALYSIS

LAGRANGIAN DERIVATION: SECOND PRINCIPAL COMPONENT

From the knowledge that $\gamma_2 = 0$, we can update the following expression

$$\left(\Sigma p_{1:m,2} - \lambda_2 p_{1:m,2} - \gamma_2 \lambda_1 p_{1:m,1} \right) = 0.$$

to

$$\left(\Sigma p_{1:m,2} - \lambda_2 p_{1:m,2} \right) = 0.$$

This is clearly another eigenvalue equation. This eigenvalue equation does not explicitly depend on knowledge of the first principal component $p_{1:m,1}$. We therefore will choose $p_{1:m,2}$ to be the eigenvector associated with the second largest eigenvalue λ_2. As before, we have that var$(p_{1:m,2}^\top x) = \lambda_2$.

PRINCIPAL COMPONENTS ANALYSIS

ALGORITHMIC OVERVIEW

We can extend the preceding computation to an arbitrary number of principal components:

Step 1: Subtract the mean from each observation to form a new vector $\varphi_i \in \mathcal{R}^m$: $\varphi_i = x_i - \frac{1}{n}\sum_{j=1}^{n} x_j$, $i = 1, \dots, n$.

Step 2: Compute the sample covariance matrix $\Sigma \in \mathcal{R}^{m\times m}$: $\Sigma = \frac{1}{n}\sum_{j=1}^{n} \varphi_j \varphi_j^\top$.

Step 3: Find the eigenvalues $\lambda_1 \in \mathcal{R}$, ..., $\lambda_m \in \mathcal{R}$ and eigenvectors $p_{1:m,1} \in \mathcal{R}^m$, ..., $p_{1:m,m} \in \mathcal{R}^m$ of the sample covariance matrix. The principal components are taken as the eigenvectors.

We can construct a reduced-dimensional representation as follows:

Step 4: Keep only the top k eigenvalues and eigenvectors. The reduced representation is given by $[\varphi_1, \dots, \varphi_n][p_{1:m,1}, \dots, p_{1:m,k}] \in \mathcal{R}^{n\times k}$.

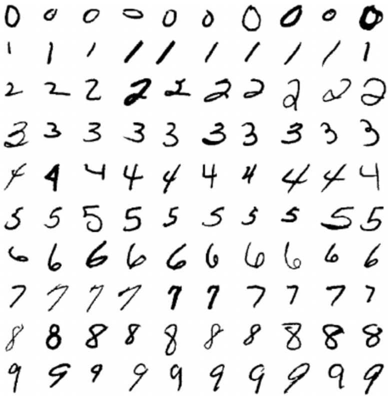
DIMENSIONALITY REDUCTION EXAMPLE

MNIST DATASET

Objective: Want to develop a system that can determine the hand-written numerical digits from images.

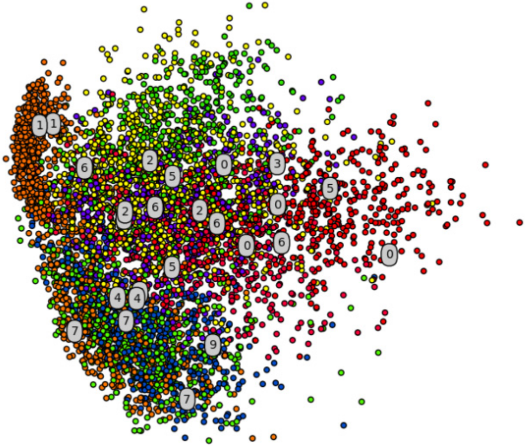
Data (MNIST): Pre-segmented 28 × 28 pixel images of single digits. 70k images are provided from different people.

Process: Unwrap each image so that it is a vector of size 781 × 1. Reduce the vector dimensionality using dimensionality reduction (avoid curse of dimensionality). Learn a mapping from a set of training data (e.g., 40k to 60k images of different digits from different people) to the set of integers. Use a set of testing data (e.g., 10k to 20k images that were not in the training set) to evaluate the mapping quality.



DIMENSIONALITY REDUCTION EXAMPLE

MNIST DATASET: PRINCIPAL COMPONENTS ANALYSIS

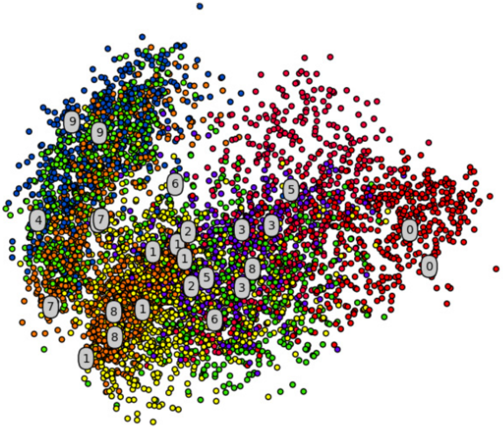


When applying principal components analysis to the MNIST dataset, we find that there is a great amount of overlap between the various hand-written digits. The one and seven digits are intermixed and difficult to separate. Likewise, the zero, two, six, and five digits at the center have significant overlap. This suggests that the recognition rate may be low.

Principal components analysis does not account for the class labels when performing reduction. Therefore, we may run into issues when using supervised machine learning methods.

DIMENSIONALITY REDUCTION EXAMPLE

MNIST DATASET: ISOMAP

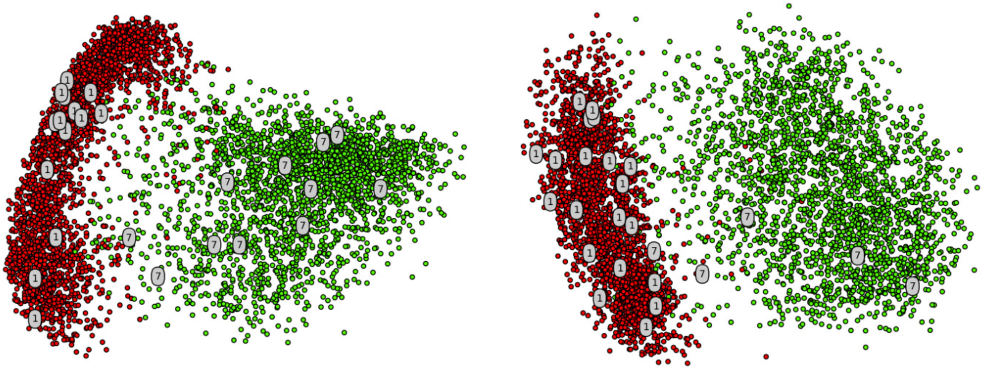


It is possible that the MNIST digits may lie along a complicated surface. Principal components analysis is often not able to deal with such surfaces well. We could instead consider a manifold-learning-based approach, like isometric embedding (ISOMAP). ISOMAP attempts to maintain the distances between observations during reduction, rather than trying to maximize variance.

Note that, like principal components analysis, ISOMAP does not account for the class labels during the projection process.

DIMENSIONALITY REDUCTION

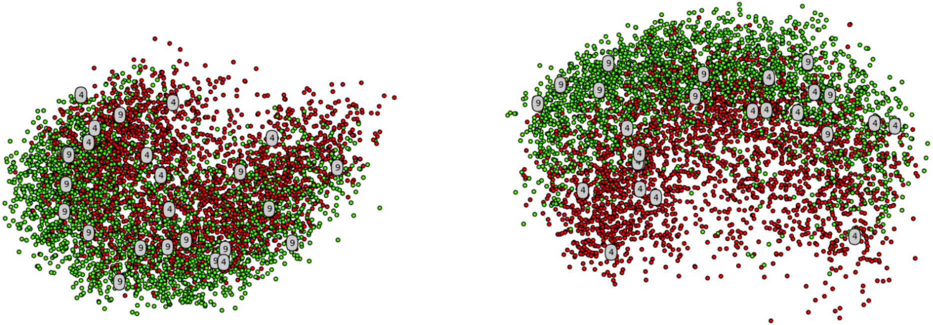
MNIST DATASET: PCA VS ISOMAP



The overlap between the one and seven digits is more pronounced for principal components analysis (left) than it is for ISOMAP (right). The reduced representation from ISOMAP is nearly linearly separable for these digits.

DIMENSIONALITY REDUCTION

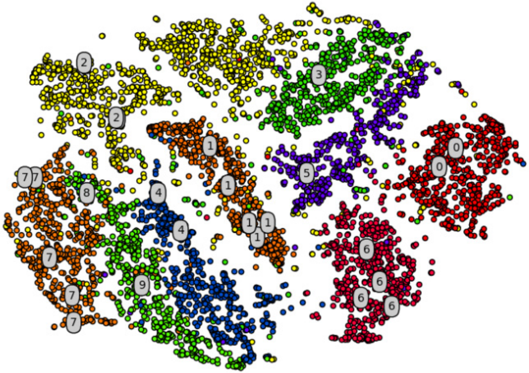
MNIST DATASET: PCA VS ISOMAP



The separability of the four and nine digits is significantly less for principal components analysis (left) than it is for ISOMAP (right). This implies that the recognition rate for the ISOMAP-reduced data should be better for a variety of supervised machine learning methods.

DIMENSIONALITY REDUCTION EXAMPLE

MNIST DATASET: ADVANCED TECHNIQUES



Using more advanced techniques can improve the separation of the digits and hence yield better recognition performance. These techniques may assume that the observations naturally group into multiple, complicated surfaces. The reduction of these surfaces can be considered independently. As well, such techniques may also make use of the class labels when they are available.