Dimensionality Problem

In practical system it is not uncommon to encounter vectors of fifty, hundred or even more features. Most often we are unable to assess, which features are more important – i.e. they discriminate classes well, and which are less important. One can be also (almost) sure that the features are not independent.
At the same time when feature space is high-dimensional we need adequately large training set.
(How many samples are needed in the training set to compute multivariate *d*-dimensional normal distributions for *c* classes?)
Computational complexity also grows radically with the number of dimensions *d*.
Conclusion: it would be good to reduce dimensionality.

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Primary Component Analysis - PCA

Let us assume, that we have n samples $x_1 \cdots x_n$ in d-dimensional space, which we want to represent by a single vector x_0. We define squared-error criterion function: $J_0(x_0) = \sum_{k=1}^{n} \left\| x_0 - x_k \right\|^2$. It is simple to find x_0 value, for which $J_0(x_0)$ is minimized: $x_0 = m = \frac{1}{n}\sum_{k=1}^{n} x_k$.

x_0 is 0-dimensional representation of the samples set.
In one-dimensional case, our set will be represented by the line (strictly speaking by the projections of samples onto the line) given by equation: $\mathbf{x} = \mathbf{m} + a\mathbf{e}$.

For any sample x_k its representation can be written as: $\mathbf{m} + a_k\mathbf{e}$.
(How can we show that line goes through $\mathbf{m}$?)

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Primary Component Analysis

Criterion function for one-dimensional representation:

$$J_1(\mathbf{e}) = \sum_{k=1}^{n} \left\| (\mathbf{m} + a_k \mathbf{e}) - x_k \right\|^2$$

$a_k = \mathbf{e}^T (\mathbf{x}_k - \mathbf{m})$ is the distance of projection of sample x_k from $\mathbf{m}$.

To minimize criterion function $J_1(\mathbf{e})$, i.e. to find the optimal direction *e* of the line, we will use the so-called *scatter matrix*:

$$\mathbf{S} = \sum_{k=1}^{n} (\mathbf{x}_k - \mathbf{m})(\mathbf{x}_k - \mathbf{m})^T$$

(Note that S is the sample covariance matrix times *n-1*).

Now we can write:

$$J_1(\mathbf{e}) = -\sum_{k=1}^{n} a_k^{\ 2} + \sum_{k=1}^{n} \left\| \mathbf{x}_k - \mathbf{m} \right\|^2 = -\mathbf{e}^T \mathbf{S} \mathbf{e} + \sum_{k=1}^{n} \left\| \mathbf{x}_k - \mathbf{m} \right\|^2$$

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Lagrange multipliers

We can replace computing minimum of $J_1(\mathbf{e})$ with finding maximum of $\mathbf{e}^T\mathbf{S}\mathbf{e}$. We use the method of Lagrange multipliers to find extremum of the function $f(\mathbf{x})$ subject to constraint $g(\mathbf{x})=0$. (Functions f and g must have continuous partial derivatives, g additionally non-zero gradient on the curve $g(\mathbf{x})=0$).

We create Lagrangian letting λ be undetermined Lagrange multiplier:

$$L(\mathbf{x},\lambda)=f(\mathbf{x})+\lambda g(\mathbf{x})$$

Differentiating, we can write necessary condition in convenient form:

$$\frac{\partial L(\mathbf{x},\lambda)}{\partial \mathbf{x}}=\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}+\lambda\frac{\partial g(\mathbf{x})}{\partial \mathbf{x}}=0$$

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Primary Component Analysis

We want to maximize $\mathbf{e}^T\mathbf{S}\mathbf{e}$ subject to constraint $\mathbf{e}^T\mathbf{e}-1=0$.

$$L(\mathbf{e},\lambda)=\mathbf{e}^T\mathbf{S}\mathbf{e}-\lambda\left(\mathbf{e}^T\mathbf{e}-1\right)$$

Differentiating with respect to $\mathbf{e}$ we can write:

$$\frac{\partial L(\mathbf{e},\lambda)}{\partial\mathbf{e}}=2\mathbf{S}\mathbf{e}-2\lambda\mathbf{e}$$

We search for zero gradient, that is

$$\mathbf{S}\mathbf{e}=\lambda\mathbf{e}$$

We have obtained equation, that is solved by eigenvectors and eigenvalues of scatter matrix $\mathbf{S}$. We have thus d maxima, of which we must select one. Since $\mathbf{e}^T\mathbf{S}\mathbf{e}=\mathbf{e}^T\lambda\mathbf{e}=\lambda\mathbf{e}^T\mathbf{e}=\lambda$ we select eigenvector corresponding to the largest eigenvalue of scatter matrix $\mathbf{S}$.

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Primary Component Analysis

Extension to the greater number of dimensions (d_S) is straightforward. Instead of line equation we introduce (hyper-) plane equation:

$$\mathbf{x} = \mathbf{m} + \sum_{i=1}^{d_S} a_i \mathbf{e}_i$$

Coefficients a_i are called *principal components*.
Criterion function:

$$J_{d_S} = \sum_{k=1}^{d_S} \left\| \mathbf{m} + \sum_{i=1}^{d_S} a_{ki} \mathbf{e}_i - \mathbf{x}_k \right\|^2$$

is maximized for d_S eigenvectors corresponding to d_S largest eigenvalues of scatter matrix $\mathbf{S}$.

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

PCA – matrix implementation

1. Collect n-dimensional training set $X = \{x_i\}, i = 1,2,\ldots,m$
2. Remove the mean from the set $x_i = x_i - \bar{x}$
3. Compute covariance matrix

   $C_{ij} = x_i x_j$ (because mean is already removed)
4. Compute eigenvalues and eigenvectors of matrix ***C***.

$$\mathbf{C}\boldsymbol{\alpha} = \boldsymbol{\lambda}\boldsymbol{\alpha}$$

5. Sort eigenvalues (and connected eigenvectors) so that

   $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$
6. Select first $d \leq n$ eigenvectors and transform training set to the new coordinate system.

How does it look in practice?

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

PCA – matrix implementation

1. Let’s assume that we have training set in a matrix and each row is a feature vector of an individual object (`ts`).
2. Mean value removal:
   ```
   mu = mean(ts)
   ts = ts - repmat( mean(ts), size(ts,1), 1)
   ```
3. Covariance matrix `cmx`:
   ```
   cmx = cov(ts)
   ```
4. Eigenvalues and eigenvectors:
   ```
   [evec eval] = eig(cmx)
   eval = sum(eval)
   ```
   function `eig` returns eigenvalues in a diagonal matrix

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

PCA – matrix implementation

5. Sorting eigenvalues and eigenvectors:
```
[eval evid] = sort(eval)
eval = eval(size(eval,2):-1:1)
evec = evec(:, evid(size(eval,2):-1:1)
```
or
```
[eval evid] = sort(eval, "descend");
evec = evec(:, evid(1:size(eval,2)));
```
6. Transformation matrix `trmx`:
```
trmx = evec(:, 1:comp_count)
tspca = ts * trmx
```

Now we have to answer very important question:
What value assign to `comp_count`?

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Selecting number of primary components

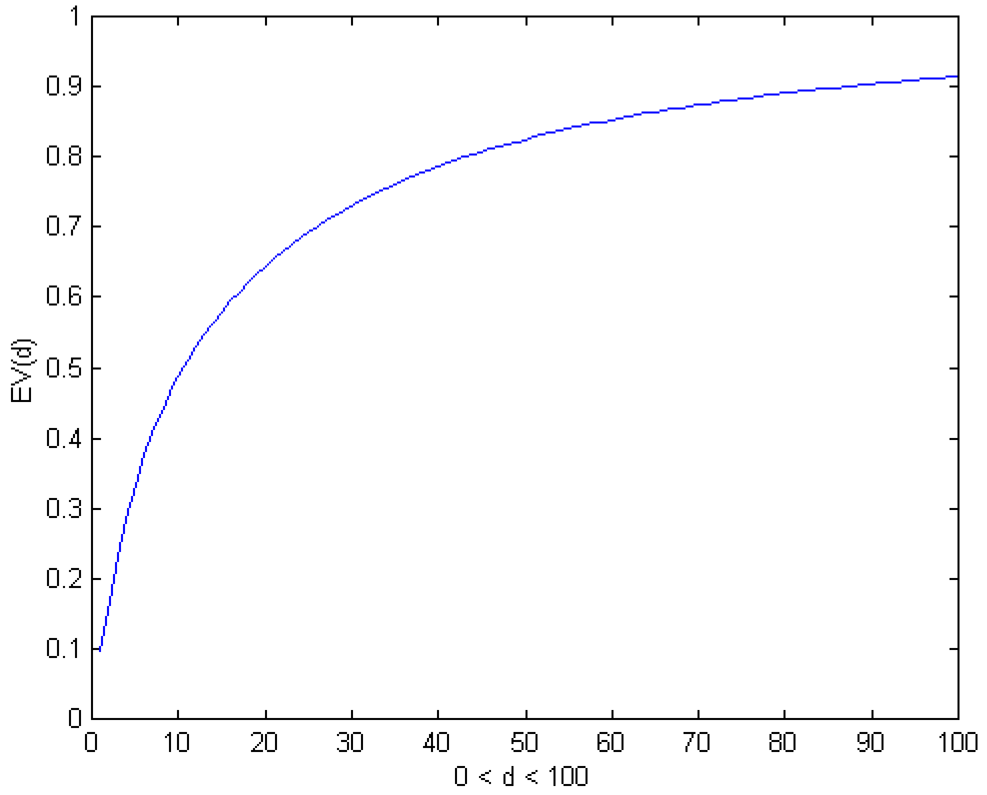
Quality of the training set approximation can be measured as a degree of „explained” set variation. For *n*-dimensional set members, and *d* primary components we have:

$$EV_d = \frac{\sum_{i=1}^{d} \lambda_i}{trace(\mathbf{C})} = \frac{\sum_{i=1}^{d} \lambda_i}{\sum_{j=1}^{n} \lambda_j}$$

EV_d can be used to decide how many of the first primary components use in classifier. We should use „elbow rule” while looking at the EV_d diagram – we can see how good d components approximate training set. What is the dependency between explained variation and classification quality, is an open question.

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Selecting number of primary components



EV$_d$ diagram for the digits from the MNIST database. Elbow rule does not work here. How many component should be used?
In practice we have to check on the testing set.

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

PCA – classification

We have feature vector `vx` of an object to be classified. We should repeat operations that were performed for the training set: removal of the mean value and transformation to the new coordinate system:

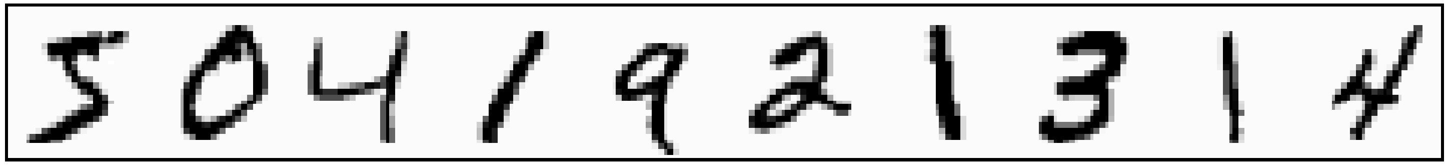
```
vxpca = (vx - mu) * trmx
```

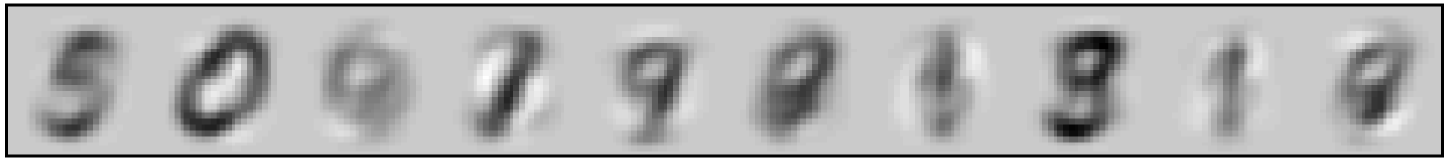
Coordinates in this new coordinate system are considered new features describing objects. We can use any classification method on this new set of features.

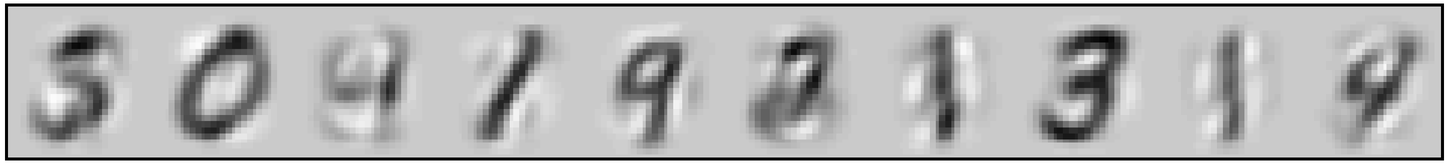
In handwritten digit (MNIST database) classification example 1-NN classifier is used, with training set containing 60000 digits. Test set contained 10000 digits.

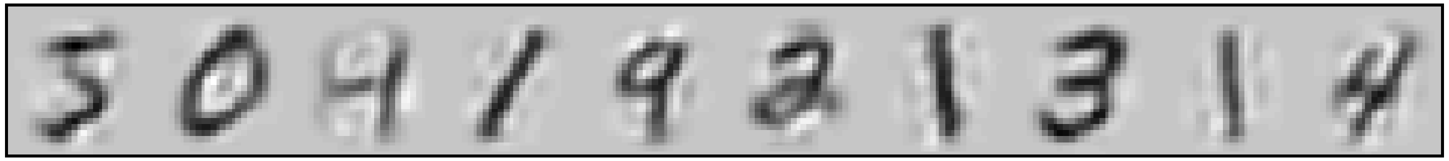
Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

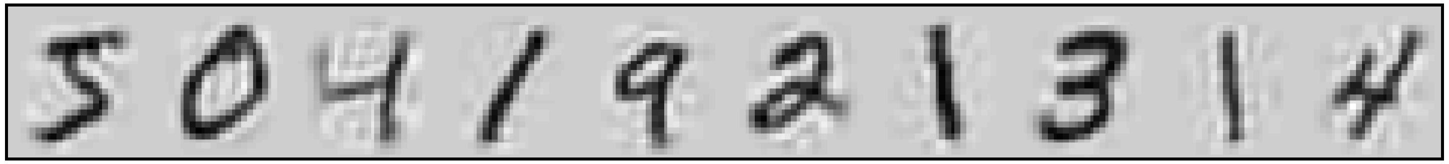
PCA – MNIST results

	Original digits

	N=10
	EV_d=0.49
	q=91.4%

	N=20
	EV_d=0.64
	q=96.4%

	N=40
	EV_d=0.79
	q=97.2%

	N=80
	EV_d=0.89
	q=97.3%

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Fisher’s Linear Discriminant

PCA generally finds components that are useful for representing data. These components may be less useful for discriminating data in different classes.
The problem of finding projections efficient for discriminating is the area of *discriminant analysis*.
We begin with the problem of projecting data from d dimensions onto a line going through the beginning of coordinate system and direction $\mathbf{w}$. Suppose that we have n samples $x_1 \cdots x_n$ in d-dimensional space: n_1 samples belonging to class ω_1 constituting subset D_1 and n_2 samples belonging ω_2 constituting subset D_2.
Sample $\mathbf{x}$ projection onto the line $\mathbf{w}$ will be given by $y = \mathbf{w}^T \mathbf{x}$.

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Fisher's Linear Discriminant

A measure of the separation between projected points of different classes is the difference of sample means. In d-dimensional space means are in the form: $\mathbf{m}_i = \frac{1}{n_i}\sum_{\mathbf{x}\in D_i}\mathbf{x}$, and after projection:

$$\tilde{m}_i = \frac{1}{n_i}\sum_{y\in Y_i} y = \mathbf{w}^T\mathbf{m}_i \ .$$

Distance between classes' means after projection is

$$\left|\tilde{m}_1 - \tilde{m}_2\right| = \left|\mathbf{w}^T(\mathbf{m}_1 - \mathbf{m}_2)\right| .$$

We define scatter of projected samples labeled ω_i as

$\tilde{s}_i^{\,2} = \sum_{y\in Y_i}(y - \tilde{m}_i)^2$. Sum $\tilde{s}_1^{\,2} + \tilde{s}_2^{\,2}$ is called total *within-class scatter*.

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Fisher’s Linear Discriminant

Fisher’s Linear Discriminant employs criterion function in the form:

$$J(\mathbf{w}) = \frac{\left|\tilde{m}_1 - \tilde{m}_2\right|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$

To find maximum of $J(\mathbf{w})$ we introduce scatter matrices $\mathbf{S}_i$ and $\mathbf{S}_W$ defined as: $\mathbf{S}_i = \sum_{\mathbf{x}\in D_i} (\mathbf{x}-\mathbf{m}_i)(\mathbf{x}-\mathbf{m}_i)^T$ and $\mathbf{S}_W = \mathbf{S}_1 + \mathbf{S}_2$.

Scatters values can be written in the form: $\tilde{s}_i^2 = \mathbf{w}^T \mathbf{S}_i \mathbf{w}$,
$\tilde{s}_1^2 + \tilde{s}_2^2 = \mathbf{w}^T \mathbf{S}_W \mathbf{w}$.

The separation of the projected means can be written as
$\left|\tilde{m}_1 - \tilde{m}_2\right|^2 = \mathbf{w}^T \mathbf{S}_B \mathbf{w}$, where $\mathbf{S}_B = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T$ is *between-class scatter* matrix.

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Fisher’s Linear Discriminant

Using scatter matrices $\mathbf{S}_B$ and $\mathbf{S}_W$ we can write criterion function in the form: $J(\mathbf{w}) = \dfrac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}}$.

$J(\mathbf{w})$ is maximized by $\mathbf{w}$ satisfying generalized eigenvalue problem:

$$\mathbf{S}_B \mathbf{w} = \lambda \mathbf{S}_W \mathbf{w}.$$

If $\mathbf{S}_W^{-1}$ exists, we can return to conventional eigenvalue problem formulated as: $\mathbf{S}_W^{-1} \mathbf{S}_B \mathbf{w} = \lambda \mathbf{w}$.

Since the length of $\mathbf{w}$ is unimportant and $\mathbf{S}_B \mathbf{w}$ is the vector in the direction $\mathbf{m}_1 - \mathbf{m}_2$. we can write solution immediately:

$\mathbf{w} = \mathbf{S}_W^{-1} (\mathbf{m}_1 - \mathbf{m}_2)$. Now we have to select threshold value w_0 separating classes.

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Multiple discriminant analysis - MDA

Fisher solution can be extended to more than two classes. In this case scatter matrices should be computed in different manner. For *n* classes between-class scatter matrix is computed according to:

$$S_B = \sum_{i=1}^{n} n_i \left(m_i - m\right)^T \left(m_i - m\right)$$

Within-class scatter matrix:

$$S_W = \sum_{i=1}^{n} S_{Wi} = \sum_{i=1}^{n} \sum_{y_k \in Y_i} \left(y_k - m_i\right)^T \left(y_k - m_i\right)$$

Further processing is analogous: we search for eigenvalues and eigenvectors in generalized form: $\mathbf{S}_B \mathbf{w} = \lambda \mathbf{S}_W \mathbf{w}$.
Note, that in case where S_W matrix is not singular, we can multiply both sides by inverse of S_W and get simple and well known eigenproblem again: $\mathbf{S}_B \mathbf{S}_W^{-1} \mathbf{w} = \lambda \mathbf{w}$

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

PCA, FLD, MDA - summary

Although FLD and MDA seem to have principal advantage over PCA, because in these methods we use information about class membership, in practice their application is limited by small number of dimensions of the resulting space. Due to a derivation of between-class matrix $\mathbf{S}_B$ (it's constructed on *n* points – mean values of samples in classes) solution will have at most *n-1* non-zero eigenvalues, which can be too "tight" space for complicated classification problems.
Next problem arises when means of classes overlap (it is also signal that features at our disposal are poor): in this case $J(\mathbf{w})=0$, so FLD and MDA cannot optimize anything (PCA can be useful even in this case).

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds