EE 649 Pattern Recognition

Dimensionality Reduction

Ulisses Braga-Neto

ECE Department

Texas A&M University

Main Idea

In many instances, it is often necessary to reduce the number of measurements (features) in a data set to make the problem tractable. Examples include:

- Digital signal/image data with thousands/millions of samples/pixels.
- Historical time-series data accumulated over years (e.g. weather, stock market, etc.)
- "Omic" (genomic, proteomic, immunomic, etc.) data with tens of thousands of molecular measurements (RNA, protein, antibody, etc.)

Why Reduce Dimensionality?

Recall that the Bayes error can never increase as the number of features increases. So why is dimensionality reduction necessary or even wanted?

- To improve classification performance. The peaking phenomenon ("curse of dimensionality") implies that the true classification error can and often does increase with more features.
- To reduce computational load, in terms of both execution time and data storage.
- To perform preliminary exploratory data analysis (visualization of high-dimensional data)

Some Heuristics

Dimensionality reduction will generally involve loss of information. One typically wants to reduce the number of features in such a way that this loss is minimized. Some heuristics for this are:

- Features that are functions of other features should be discarded.
- Features that are nearly constant (small-variance) should be discarded.
- Features strongly correlated with Y should be retained.
- Features weakly correlated with Y (i.e., "noisy features") should be discarded.

Class-Separability Criteria

Given the original feature vector $X = (X_1, \ldots, X_p) \in R^p$, dimensionality reduction finds a transformation $T : R^p \to R^d$, where $d < p$, such that the new feature vector is $X' = T(X) = (X'_1, \ldots, X'_d) \in R^d$.

To minimize the loss of information, $X' = T(X)$ should be selected such that a *class-separability criterion* $J(X', Y)$ is maximized. For example

- The Bayes error:

$$J(X', Y) = 1 - \epsilon^*(X', Y) = 1 - E[\min\{\eta(X'), 1 - \eta(X')\}]$$

- The designed classification error:

$$J_{\Psi_n}(X', Y) = 1 - \epsilon_n(X', Y) = 1 - E[|Y - \Psi_n(X'; S_n)|]$$

Note: In practice, estimates of these errors have to be used.

Class-Separability Criteria - II

Additional class-separability criteria:

- F-errors (e.g. asymptotic nearest-neighbor error ϵ_{NN}, Matsushita error ρ, expected conditional entropy $\mathcal{E}$):

$$J(X', Y) = 1 - d_F(X', Y) = 1 - E[F(\eta(X'))]$$

- Mahalanobis distance:

$$J(X', Y) = \sqrt{(\mu_1 - \mu_0)^T \Sigma^{-1} (\mu_1 - \mu_0)}$$

- Scatter-Matrices: (e.g., Fisher's discriminant)

$$J(X', Y) = \frac{w^t S_B w}{w^t S_W w}$$

where $X' = T(X) = w^T X$.

Feature Extraction $\times$ Feature Selection

There are two classes of dimensionality reduction methods:

- *Feature Extraction:* The objective is finding a general $T : R^p \to R^d$ that minimizes the loss of information according to the class-separability criterion.

- *Feature Selection:* This is feature extraction where T is restricted to be an *orthogonal projection* from R^p to R^d, that is, there is a set $A \subset \{1, \ldots, p\}$, with $|A| = d$, such that the components of $X' = T(X) \equiv X^A$ are simply X_i for $i \in A$. Therefore, each original feature is simply either retained or discarded, and the final feature set retains the "physical" meaning of the original features (this may be important in some applications, such as Genomics, and not in others, such as Digital Imaging).

Feature Selection

- Let $J(A) = J(X^A, Y)$ be a class-separability criterion associated with A. The feature selection problem is to find A^* such that

$$A^* = \arg \max_{|A|=d} J(A)$$

- Since this is a finite problem, the optimal solution is guaranteed to be reached by *exhaustive search*: compute $J(A)$ for all possible subsets $A \subset \{1, \ldots, p\}$ of size d and pick the maximum. The number of subsets to be evaluated is clearly:

$$m = \binom{p}{d} = \frac{p!}{d!(p-d)!}$$

This number can be astronomical for even modest p and d (e.g., $p = 100$ and $d = 10$ give $m > 10^{13}$).

Filter vs. Wrapper Feature Selection

- The ultimate objective of feature selection is to provide a feature vector with which to design a classifier via a classification rule.
- If the criterion $J(A)$ is indepedent of the classification rule, the method is said to be a *filter approach*.
- Otherwise, the method is said to be a *wrapper approach*. Usually in this case one has as criterion: $J(X^A, Y) = 1 - \epsilon_n(X^A, Y)$ or, in practice, $J(X^A, Y) = 1 - \hat{\epsilon}_n$, where $\hat{\epsilon}_n$ is an *error estimator* for $\epsilon_n(X^A, Y)$, based on the training data S_n.
- The true performance of the selected feature set and classifier has to be assessed either with knowledge of F_{XY} or by means of a large independent test set.

Bayes Error

- The Bayes error provides a *filter* feature selection criterion, which is very natural, as it gives a lower bound on classification error based on the selected feature set.
- Recall that it also has the monotonicity property:

$$A \subseteq B \;\Rightarrow\; \epsilon^*(X^A, Y) \geq \epsilon^*(X^B, Y)$$

  That is, the Bayes error never decreases as more features are added (we have shown this in class).
- The optimal feature set of size d in this case is called the *Bayes feature set of size* d.
- Even if one can compute the Bayes error exactly, the complexity of exhaustive search cannot be avoided in order to find the Bayes feature set, as we see next.

Cover-Van Campenhout Theorem

(Thm 32.1 DGL) Let $A_1, A_2, \ldots A_{2^p}$ be *any* ordering of all possible subsets of $\{1, \ldots, p\}$, satisfying only the constraint $i < j$ if $A_i \subset A_j$ (hence, $A_1 = \emptyset$ and $A_{2^p} = \{1, \ldots, p\}$). Let $\epsilon^*(A) = \epsilon^*(X^A, Y)$ for short. There is a distribution of (X, Y) such that

$$\epsilon^*(A_1) > \epsilon^*(A_2) > \cdots > \epsilon^*(A_{2^p})$$

Corollary 1: For all possible subsets $A \subset \{1, \ldots, p\}$ of size d any ordering of the Bayes error is possible.

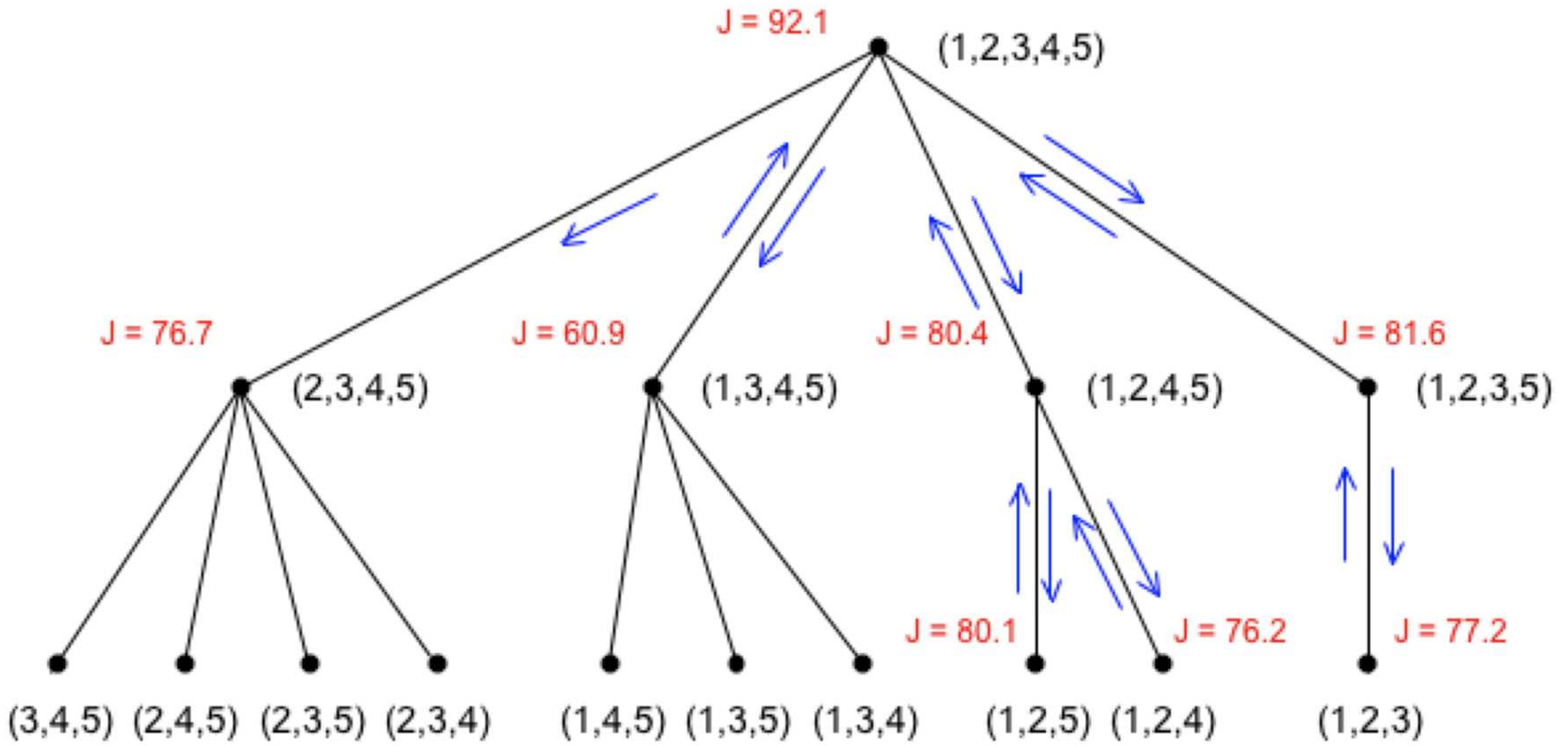
Corollary 2: No algorithm to find the Bayes feature set of size d can be guaranteed to beat the complexity of exhaustive search.

Branch-and-Bound Algorithm

- Consider class-separability criteria that have the monotonicity property: $A \subseteq B \Rightarrow J(A) \leq J(B)$ (e.g., the Bayes error, or any F-error, yields this property).
- For such J, the *branch-and-bound* algorithm is a graph-search technique that can find the optimal feature set doing fewer evaluations than exhaustive search.
- This does not contradict the Cover-Van Campenhout Theorem in the case of the Bayes error, since worst-case performance in that case is still equivalent to exhaustive search. There is also the extra overhead of graph construction and search.
- In practice, error estimates must be used and the monotonicity property does not hold. There is however some evidence that the BAB algorithm is robust for violations of monotonicity.

Branch-and-Bound Algorithm - II

Example (Webb): $p = 5$ and $d = 3$



The BAB algorithm makes evaluates 8 feature sets, whereas exhaustive search evaluates $\binom{5}{3} = 10$ feature sets

Sub-Optimal Search

- A sub-optimal solution in many cases cannot be avoided due to the sheer size of the problem.
- There are a number of fast feature selection algorithms that execute sub-optimal searches.
  - Best Individual d Features
  - Sequential Forward Search
  - Sequential Backward Search
  - Plus-l Take-r Search
  - Generalized Sequential Forward Search
  - Generalized Sequential Backward Search
  - Generalized Plus-l Take-r Sequential Search
  - Floating Search

Best Individual d Features

- This is the simplest method: just compute $J(X_i, Y)$ for each individual original feature X_i, and pick the d features with largest J.
- This is an intuitive heuristic, but it can fail badly, as it ignores multivariate relationships.
- For a simple theoretical counter-example, consider the case where the best d features are equal: $X'_1 = \cdots X'_d$
- This method is nevertheless quite common. It is often based on the correlation of individual features with Y or scores such as the t-score or rank-sum score.
- Surprisingly, in small-sample cases, such simple filter methods can outperform more complex wrapper approaches.

Toussaint's Counter-Example

Even if all p original features are uncorrelated, the result of best-individual method can be very bad.

This surprising fact is shown, in the case of the Bayes error, by the following result.

(Thm 32.2 DGL) Let $p = 3$. There is a distribution of (X, Y) such that X_1, X_2 and X_3 are conditionally-independent given Y and

$$\epsilon^*(\{1\}) < \epsilon^*(\{2\}) < \epsilon^*(\{3\})$$

But such that

$$\epsilon^*(\{1,2\}) > \epsilon^*(\{1,3\}) > \epsilon^*(\{2,3\})$$

Therefore, the best 2 individual features form the worst 2-feature set, and the worst 2 individual features form the best 2-feature set.

Sequential Forward Search

Sequential methods generally outperform best individual feature selection (except in small-sample cases).

- Sequential Forward Search (bottom-up search):
  - Let $X_{(0)} = \emptyset$.
  - Given the current feature set $X_{(k)}$, the criterion $J(X_{(k)} \cup X_i, Y)$ is evaluated for each $X_i \notin X_{(k)}$ and the X_i^* that maximizes this is added to the feature set: $X_{(k+1)} = X_{(k)} \cup X_i^*$.
  - Stop if $k = d$ or if no improvement is possible.

This has the disadvantage that once a feature is added, it is "frozen" in place, i.e. it can never be removed from the working feature set.

Sequential Backward Search

- Sequential Backward Search (top-down search):
  - Let $X_{(0)} = X$.
  - Given the current feature set $X_{(k)}$, the criterion $J(X_{(k)} \setminus X_i, Y)$ is evaluated for each $X_i \in X_{(k)}$ and the X_i^* that minimizes the drop

$$J(X_{(k)}, Y) - J(X_{(k)} \setminus X_i, Y)$$

    is removed from the feature set: $X_{(k+1)} = X_{(k)} \setminus X_i^*$.
  - Stop at $k = d$.

The main disadvantage of this method is that feature sets of high dimensionality have to be considered. If the criterion J involves the classification error (e.g. wrapper feature selection), then this method is impractical for large p.

Generalized Sequential Search

- Generalized Sequential Forward Search: This is a generalization of sequential forward search, where at each stage, all combinations Z_j of r features not in the current feature set $X_{(k)}$ are considered, and the group Z_j^* that maximizes $J(X_{(k)} \cup Z_j, Y)$ is added: $X_{(k+1)} = X_{(k)} \cup Z_j^*$.

- Generalized Sequential Backward Search: This is a generalization of sequential forward search, where at each stage, all combinations Z_j of r features in the current feature set $X_{(k)}$ are considered, and the group Z_j^* that minimizes the drop $J(X_{(k)}, Y) - J(X_{(k)} \setminus Z_j, Y)$ is removed: $X_{(k+1)} = X_{(k)} \setminus Z_j^*$.

Plus-l Take-r Search

- This allows back-tracking in the sequential search.
- If $l > r$ this is a bottom-up search. At each stage, l features are added to the current feature set using SFS and then r features are removed using SBS.
- If $r > l$ this is a top-down search. At each stage, r features are removed from the current feature set using SBS and then l features are added using SFS.
- Generalized Plus-l Take-r Search: This uses GSFS and GSBS instead of SFS and SBS, respectively.

Floating Search

- This can be considered a development of the Plus-l Take-r Search method, where the values of l and r are allowed to vary, i.e., "float," at different stages of the feature selection process.
- The advantage of this method is that one is allowed to backtrack in an "optimal" sense.
- There is a bottom-up version (SFFS) and a top-down version (SFBS).

Feature Extraction

- The objective here is purely finding a $T : R^p \rightarrow R^d$ that minimizes the loss of information according to the class-separability criterion.
- The transformation $X' = T(X)$ introduces "compression," whereby some features may contribute more, less, or be completely eliminated from the final feature set X'.
- Feature extraction is recommended over feature selection if the physical meaning of the original features is not important, since the class-separability obtained is generally larger for feature extraction, at the same feature size d.

"Lossless" Transformations

- If the class-separability criterion is the Bayes error, or indeed any F-error, we have proved in class that if the transformation $X' = T(X)$ is invertible, then the compression is "lossless", i.e. $J(X', Y) = J(X, Y)$.

- Another example of lossless feature extraction, with respect to the Bayes error, is given by $\eta : R^p \rightarrow R$, where $X' = \eta(X) = P(Y = 1|X)$. In this case, a single feature contains all the discriminatory information in (X, Y). But of course, η is usually unknown, or only partially known.

"Lossless" Transformations - II

- Similarly, if the class-conditional densities are Gaussian, we know that the optimal discriminant is given by

$$g(X) = \frac{1}{2}\left[(X-\mu_0)^T\Sigma_0^{-1}(X-\mu_0) - (X-\mu_1)^T\Sigma_1^{-1}(X-\mu_1)\right] + \frac{1}{2}\ln\frac{|\Sigma_0|}{|\Sigma_1|} + \ln\frac{P(Y=1)}{P(Y=0)}$$

Thus, $g : R^p \rightarrow R$, with $X' = g(X)$, is a lossless transformation. Of course, in practice μ_i and Σ_i have to be estimated from the data, so that even if the Gaussian assumption holds, the transformation is not lossless (but the loss would tend to zero as $n \rightarrow \infty$).

Minimum-Risk Classification

- We will show how to characterize lossless transformations if the criterion is the *Bayes Risk*.
- Let us recall the minimum-risk classification set-up. We define a cost function $C : \{0,1\}^2 \rightarrow R$

$$C(i,j) = \lambda_{ij}, \quad i,j = 0,1$$

where l_{ij} is the loss incurred by deciding class i when the true class is j.

- The expected loss of a classifier ψ given $X = x$, also called the conditional risk, is given by

$$\begin{aligned} R(\psi(x) = i | X = x) &= \sum_{j=0}^{1} \lambda_{ij} P(Y = j | X = x) \\ &= \lambda_{i1}\, \eta(x) + \lambda_{i0}(1 - \eta(x)), \;\; i = 0,1 \end{aligned}$$

Minimum-Risk Classification - II

- The risk of ψ is the expected risk

$$R_\psi(X,Y) = E\left[R(\psi(X)|X)\right]$$

- The minimum-cost classifier ψ_C^* is obtained by minimizing the conditional risk at each point $X = x$:

$$\psi_C^*(x) = \begin{cases} 1, & (\lambda_{01} - \lambda_{11})\eta(x) > (\lambda_{10} - \lambda_{00})(1 - \eta(x)) \\ 0, & \text{otw} \end{cases}$$

- The minimal risk or Bayes risk is given by:

$$R_C^*(X,Y) = E\left[R(\psi_C^*(X)|X)\right]$$

Admissible Transformations

- Let the class-separability criterion be the Bayes risk.
- A transformation $X' = T(X)$ is said to be admissible if there is no loss of information for *any* cost function:

$$R^*_C(X', Y) = R^*_C(X, Y), \quad \text{for all } C : \{0, 1\}^2 \to R$$

- It can be shown that each invertible T is admissible, and so is $T(X) = \eta(X)$.
- In a sense, every admissible T must be related to $\eta(X)$:

  (Thm 32.5 DGL) A transformation $T : R^p \to R^d$ is admissible if and only if there is a mapping $G : R^d \to R$ such that

$$\eta(X) = G(T(X)) \quad \text{with probability 1}$$

Sufficient Statistic

- The transformed feature set $X' = T(X)$ is called a *sufficient statistic* if

$$\eta(X, X') = P(Y = 1|X', X) = P(Y = 1|X') = \eta(X')$$

- The following theorem establishes that admissible transformations *are* sufficient statistics:

  (Thm 32.6 DGL) A transformation T is admissible if and only if $X' = T(X)$ is a sufficient statistic.

- So T is lossless if and only if the information in $X' = T(X)$ is sufficient for X.

Some Examples

- If the class-conditional densities are Gaussian with equal a-priori probabilities, then $(\mu_0, \mu_1, \Sigma_0, \Sigma_1)$ is a sufficient statistic (even though the optimal discriminant $g(X)$ would be a more efficient, univariate statistic).
- If only partial information is available about the distribution, the concept of sufficient statistic can be quite useful in practice.
  - Example 1: if $\eta(X) = e^{-c||X||}$, for some unknown $c > 0$, then $||X||$ is a univariate sufficient statistic.
  - Example 2: if $\eta(X_1, X_2, X_3) = g(X_1X_2, X_2X_3)$, for any fixed function $g : R^3 \rightarrow R^2$, then $X' = (X_1X_2, X_2X_3)$ is a bivariate sufficient statistic.

Principal Component Analysis (PCA)

- PCA is based on the previously-mentioned heuristic according to which low-variance features should be avoided. Here, an extra step of feature decorrelation is applied first.
- After the decorrelation step, the first d individual (transformed) features with the largest variance are retained.
- Therefore, PCA uses the best individual features approach, with uncorrelated features.
- The main issue with PCA for classification is that it is *unsupervised*, i.e., the dependence of Y on X is not considered. Variants exist that address this (e.g. using scatter matrices instead of the covariance matrix).

Discrete Karhunen-Loève Transform

- Given X, we can always find a set of p orthonormal eigenvectors $u_1, \ldots, u_p$ for the covariance matrix Σ_X, corresponding to nonnegative eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$.
- Consider the linear transformation given by

$$Z = U^T(X - \mu)$$

where $U = [u_1 \ldots u_p]$ and $\mu = E[X]$.

- Clearly,

$$E[Z] = E\left[U^T(X - \mu)\right] = U^T\left(E[X] - \mu\right) = 0$$

so that the Z_i are all zero-mean, for $i = 1, \ldots, p$.

Discrete Karhunen-Loève Transform-II

- It follows that

$$\Sigma_Z = E[ZZ^T] = E\left[U^T(X-\mu)(X-\mu)^T U\right]$$
$$= U^T E\left[(X-\mu)(X-\mu)^T\right] U = U^T \Sigma_X U = \Lambda$$

  where Λ is the diagonal matrix of eigenvalues $\lambda_1, \ldots, \lambda_p$.

- Therefore,

$$E[Z_i Z_j] = 0, \quad \text{for } i \neq j$$

  that is, the Z_i are uncorrelated, and

$$\sigma_i^2 = \text{Var}(Z_i) = E[Z_i^2] = \lambda_i$$

  so the variance of Z_i is given by the corresponding eigenvalue λ_i.

Discrete Karhunen-Loève Transform-III

- The equations

$$Z_i = u_i^T (X - \mu)\,, \;\; i = 1, \ldots, p$$

subject to

$$\Sigma_X \, u_i = \sigma_i^2 \, u_i \,, \;\; i = 1, \ldots, p$$

define the discrete Karhunen-Loève transform.

- The discrete KL transform produces zero-mean, uncorrelated transformed features. This is similar to the *whitening* transformation mentioned previously:

$$W = \Lambda^{-\frac{1}{2}} U^T (X - \mu)$$

except that the whitening transformation also normalizes all variances to unity.

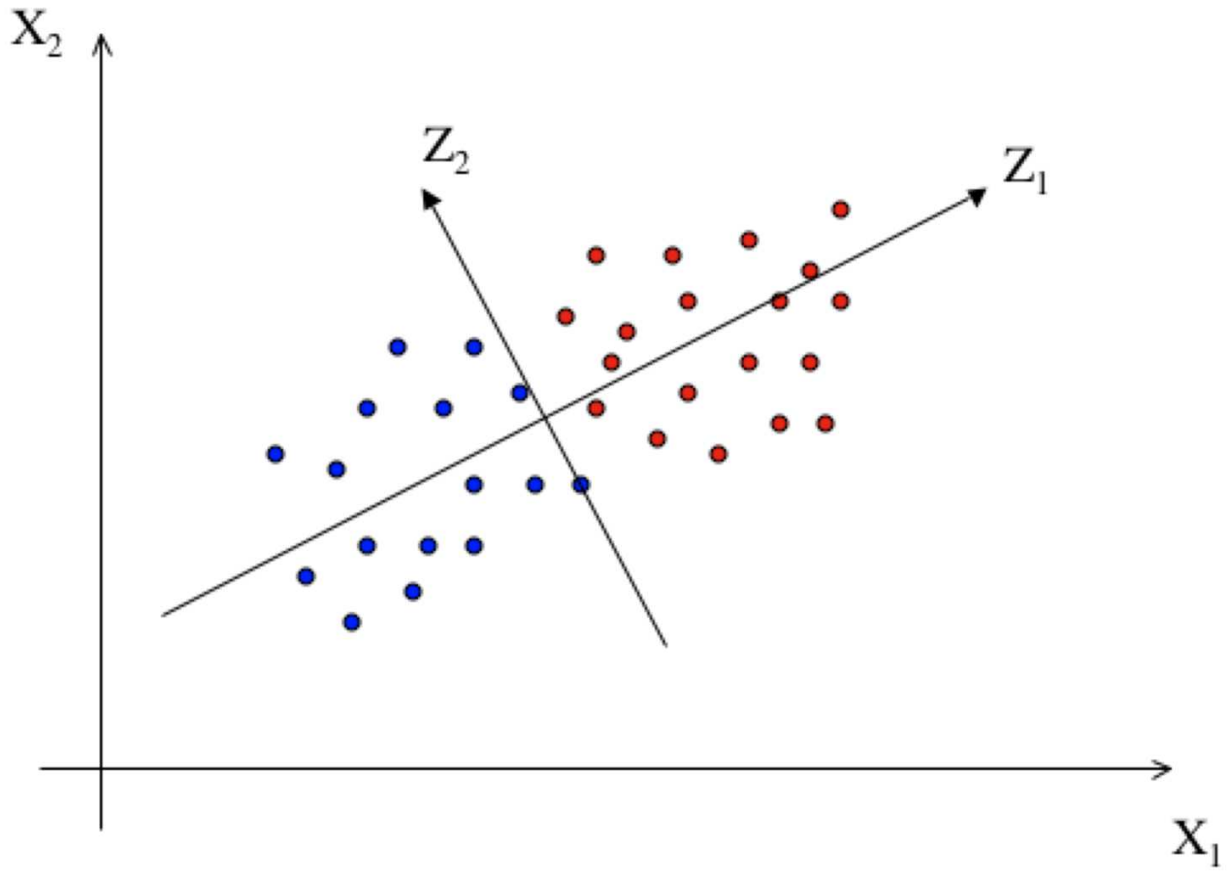
PCA Transform

- The component Z_i is the i-th *principal component*.
- The first PC Z_1 has the maximal variance λ_1, and the eigenvector u_1 points to the direction of maximal variation. The second PC Z_2 has the maximal variance in a direction perpendicular to u_1, while Z_3 has the maximal variance perpendicular to u_1 and u_2, and so on.
- The PCA transform $X' = T(X)$ consists of applying the discrete KL transform and then keeping the first d principal components $X' = (Z_1, \ldots, Z_d)$. In other words

$$X' = A^T(X - \mu)$$

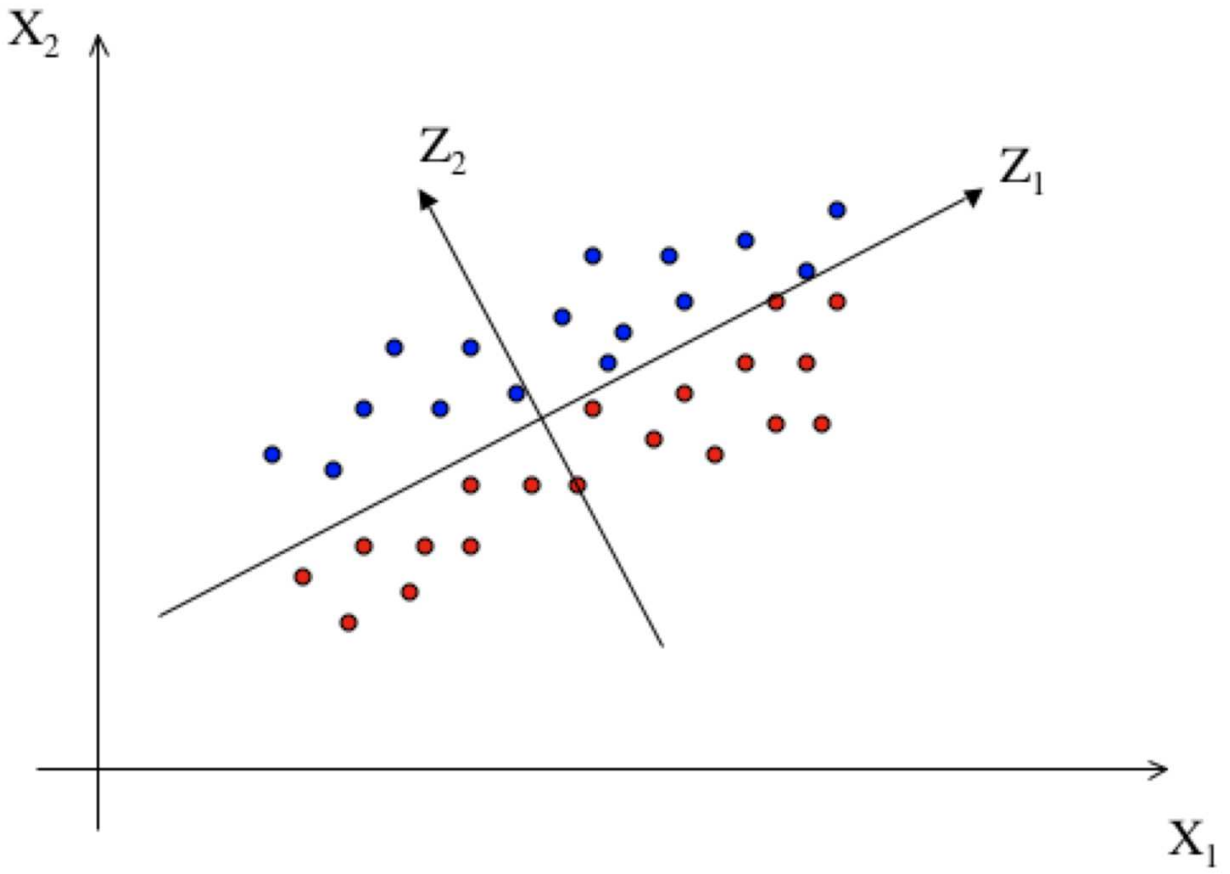
where $A = [u_1 \cdots u_d]$ is a rank-d matrix (therefore PCA is not in general invertible and lossy with respect to the Bayes error criterion).

PCA Example



The first principal component Z_1 alone contains most of the discrimination information.

PCA (Counter-)Example



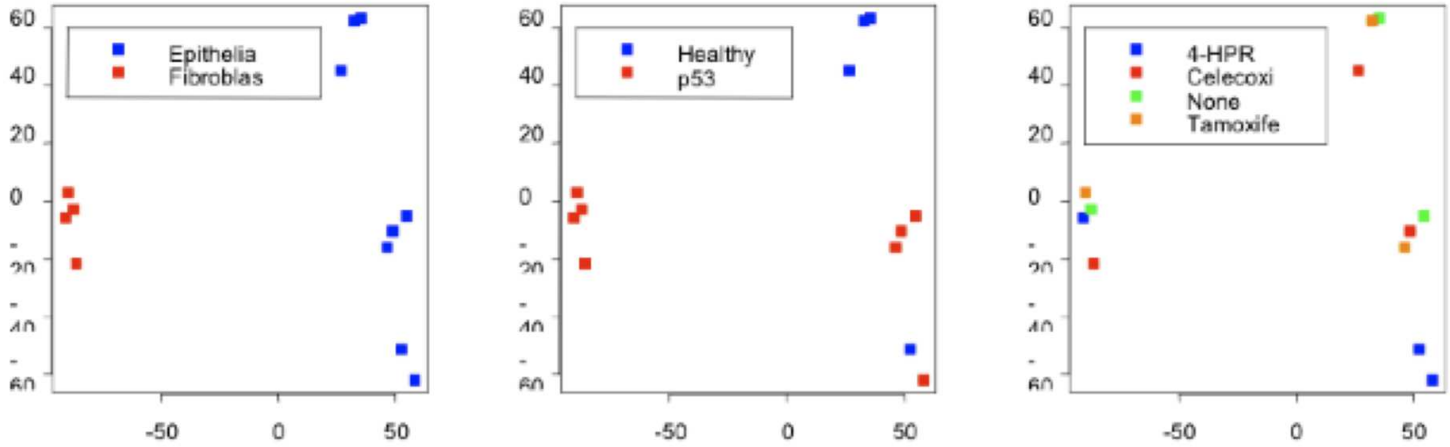
Here, the discrimination information is contained in the second principal component Z_2!

Real-Data 2-D PCA Example

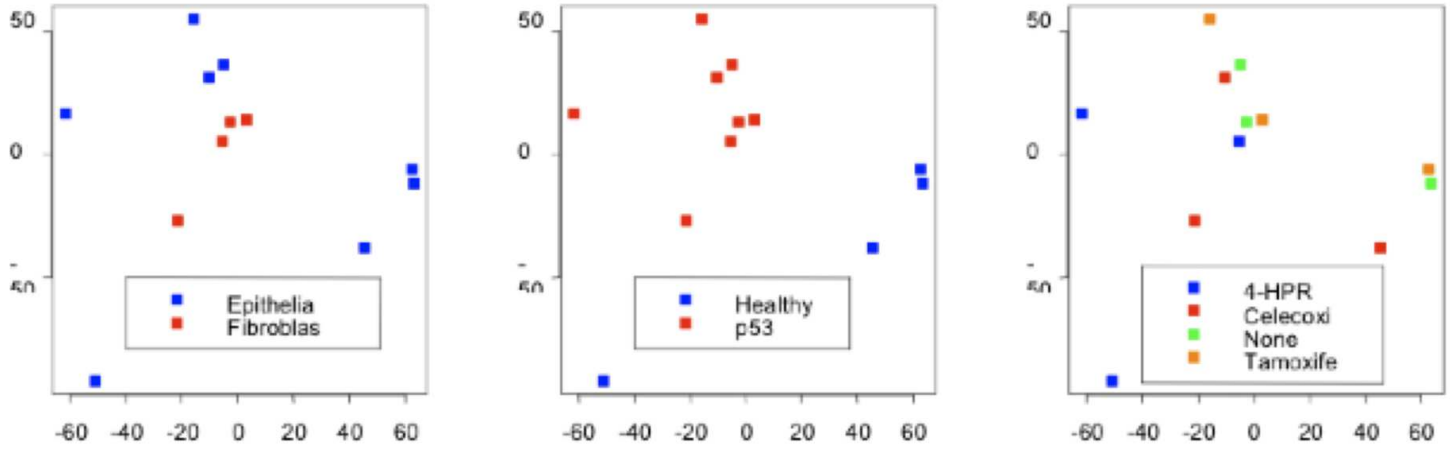
- Cancer chemotherapy study
- Gene expression data with 12 samples.
- Reduction from 12,573 initial genes to 2 features.
- Three groupings:
  - Cell type: Epithelial cells (8) vs. fibroblasts (4)
  - p53 status: “Healthy” patients (4) vs. p53-mutant patients (8)
  - Treatment: 4-HPR (3) vs. Tamoxifen (3) vs. Celecoxib (3) vs. none (3)
- Data produced by Louise Strong’s group, processed by Kevin Coombes – MD Anderson Cancer Center.

Real-Data 2-D PCA Example - II

- First PC (x axis) vs. Second PC (y axis)



- Second PC (x axis) vs. Third PC (y axis)



Multidimensional Scaling

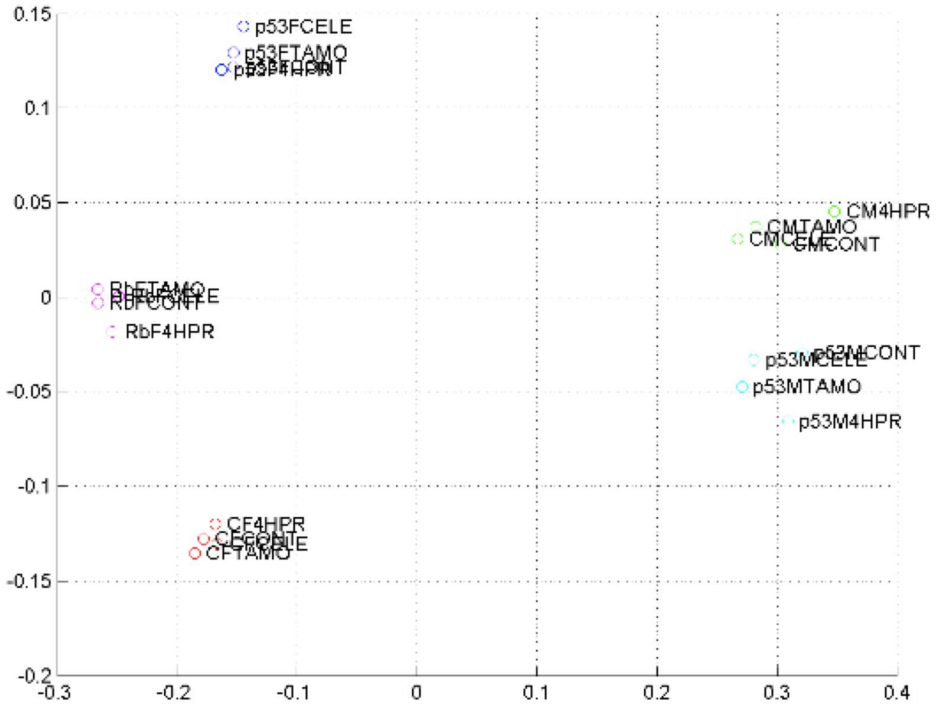
- The main idea is to find points in R^d that best approximate pairwise dissimilarities (e.g., Euclidean distances, $1-$correlation) in the original space R^p.
- If δ_{ij} and d_{ij} are the dissimilarities between original and transformed points, respectively, the goodness of fit can be measured by the *stress* (values < 10% are good):

$$S = \sqrt{\frac{\sum_{i,j}(\delta_{ij}^2 - d_{ij}^2)}{\sum_{i,j} d_{ij}^2}}$$

- This is nonlinear feature extraction, which can be advantageous over linear methods such as PCA.
- The main issues are that it is unsupervised, and it is not simple to express $T(X)$ to apply to a new sample point.

Real-Data 2-D MDS Example

- Data from previous cancer study (with 8 new samples).
- Reduction from 904 initial genes to 2 features.
- Processed by our group. Stress = 4.64%



Real-Data 3-D MDS Example

- Reduction of same data to 3 features. Stress = 1.83%