Note from Miguel Carreira-Perpiñán

Dimensionality reduction and feature selection

- If we want to train a classifier (or regressor) on a sample $\{(\mathbf{x}_n, y_n)\}_{n=1}^N$ where the number of features D in $\mathbf{x}$ (the dimension of $\mathbf{x}$) is large:
  - Training will be slow.
  - Learning a good classifier will require a large sample.
- It is then convenient to transform each example $\mathbf{x}_n \in \mathbb{R}^D$ into a new example $\mathbf{z}_n = \mathbf{F}(\mathbf{x}_n) \in \mathbb{R}^L$ having lower dimension $L < D$ (as long as we don't lose much information). This would work perfectly if the data points did lie on a manifold of dimension L contained in $\mathbb{R}^D$.
- Two basic ways to do this: $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5)^T \Rightarrow$ for $L = 2$:
  - *Feature selection*: $\mathbf{F}(\mathbf{x}) =$ a subset of $x_1, \dots, x_D$. $\rightarrow \mathbf{F}(\mathbf{x}) = \left(\begin{smallmatrix} x_2 \\ x_5 \end{smallmatrix}\right)$.
    It doesn't modify the features, it simply selects L and discards the rest.
    Ex: best-subset/forward/backward selection.
  - *Dimensionality reduction* (DR): $\mathbf{F}(\mathbf{x}) =$ a l.c. or some other function of all the $x_1, \dots, x_D$. $\rightarrow \mathbf{F}(\mathbf{x}) = \left(\begin{smallmatrix} 1x_1+3x_2-5x_3+5x_4-4x_5 \\ 2x_1+3x_2-1x_3+0x_4+2x_5 \end{smallmatrix}\right)$.
    It constructs L new features and discards the original D features.
    Ex: PCA, LDA...
- If reducing to $L \leq 3$ dimensions, can visualize the dataset and look for patterns (clusters, etc.).
- DR algorithms learn one or more of the following:
  - The *dimensionality reduction or projection mapping* $\mathbf{F}$: $\mathbf{x} \in \mathbb{R}^D \rightarrow \mathbf{z} \in \mathbb{R}^L$.
  - The *reconstruction mapping* $\mathbf{f}$: $\mathbf{z} \in \mathbb{R}^L \rightarrow \mathbf{x} \in \mathbb{R}^D$.
    The image of $\mathbf{f}$ defines a *subspace* or *manifold* of dimension L contained in $\mathbb{R}^D$.
  - The latent projections $\mathbf{z}_1 = \mathbf{F}(\mathbf{x}_1), \dots, \mathbf{z}_N = \mathbf{F}(\mathbf{x}_N) \subset \mathbb{R}^L$ of the training points.

Review of eigenvalues and eigenvectors

For a real symmetric matrix $\mathbf{A}$ of $D \times D$:

- *Eigenvalues* and *eigenvectors* of: $\mathbf{Au} = \lambda\mathbf{u} \Rightarrow \begin{cases} \lambda \in \mathbb{R}: & \text{eigenvalue} \\ \mathbf{u} \in \mathbb{R}^D: & \text{eigenvector.} \end{cases}$
- $\mathbf{A}$ has D eigenvalues $\lambda_1 \geq \cdots \geq \lambda_D$ and D corresponding eigenvectors $\mathbf{u}_1, \dots, \mathbf{u}_D$.
- Eigenvectors of different eigenvalues are orthogonal: $\mathbf{u}_i^T \mathbf{u}_j = 0$ if $i \neq j$.
- $\mathbf{A}$ is $\begin{cases} \textit{nonsingular}: & \text{all } \lambda \neq 0 \\ \textit{positive definite}: & \text{all } \lambda > 0 \quad (\Leftrightarrow \mathbf{x}^T\mathbf{Ax} > 0 \ \forall \mathbf{x} \neq \mathbf{0}) \\ \textit{positive semidefinite}: & \text{all } \lambda \geq 0 \quad (\Leftrightarrow \mathbf{x}^T\mathbf{Ax} \geq 0 \ \forall \mathbf{x} \neq \mathbf{0}). \end{cases}$
- *Spectral theorem*: $\mathbf{A}$ symmetric, real with normalized eigenvectors $\mathbf{u}_1, \dots, \mathbf{u}_D \in \mathbb{R}^D$ associated with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_D \in \mathbb{R} \Rightarrow \mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T = \sum_{i=1}^D \lambda_i \mathbf{u}_i \mathbf{u}_i^T$ where $\mathbf{U} = (\mathbf{u}_1 \dots \mathbf{u}_D)$ is orthogonal and $\mathbf{\Lambda} = \operatorname{diag}(\lambda_1, \dots, \lambda_D)$. In other words, a symmetric real matrix can be diagonalized in terms of its eigenvalues and eigenvectors.
- $\lambda_1 = \max_{\mathbf{x} \neq \mathbf{0}} \frac{\mathbf{x}^T\mathbf{Ax}}{\mathbf{x}^T\mathbf{x}} \quad \Leftrightarrow \max_{\mathbf{x}} \mathbf{x}^T\mathbf{Ax} \text{ s.t. } \|\mathbf{x}\| = 1, \quad$ achieved at $\mathbf{x} = \mathbf{u}_1$.
  $\lambda_2 = \max_{\mathbf{x} \neq \mathbf{0}} \frac{\mathbf{x}^T\mathbf{Ax}}{\mathbf{x}^T\mathbf{x}} \text{ s.t. } \mathbf{x}^T\mathbf{u}_1 = 0 \quad \Leftrightarrow \max_{\mathbf{x}} \mathbf{x}^T\mathbf{Ax} \text{ s.t. } \|\mathbf{x}\| = 1, \ \mathbf{x}^T\mathbf{u}_1 = 0, \quad$ achieved at $\mathbf{x} = \mathbf{u}_2$.
  etc.
- *Covariance matrix* $\mathbf{\Sigma} = \frac{1}{N} \sum_{n=1}^N (\mathbf{x}_n - \boldsymbol{\mu})(\mathbf{x}_n - \boldsymbol{\mu})^T$ positive definite (unless zero variance along some dim.). Mahalanobis distance $(\mathbf{x} - \boldsymbol{\mu})^T \mathbf{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) = 1 \Rightarrow$ ellipsoid with axes $= \sqrt{\lambda_1}, \dots, \sqrt{\lambda_D}$ (stdev along PCs).
- $\mathbf{w} \in \mathbb{R}^D$: $\operatorname{var}\{\mathbf{w}^T\mathbf{x}_1, \dots, \mathbf{w}^T\mathbf{x}_N\} = \mathbf{w}^T\mathbf{\Sigma}\mathbf{w}$. In general for $\mathbf{W}_{D \times L}$: $\operatorname{cov}\{\mathbf{W}^T\mathbf{x}_1, \dots, \mathbf{W}^T\mathbf{x}_N\} = \mathbf{W}^T\mathbf{\Sigma}\mathbf{W}$.

Feature selection: forward selection

- Problem: given a sample $\{(\mathbf{x}_n, y_n)\}_{n=1}^N$ with $\mathbf{x}_n \in \mathbb{R}^D$, determine the best subset of the D features such that the number of selected features L is as small as possible and the classification accuracy (using a given classifier, e.g. a linear SVM) is as high as possible.
  Using all D features will always give the highest accuracy on the training set but not necessarily on the validation set.

- Useful when some features are unnecessary (e.g. irrelevant for classification or pure noise) or redundant (so we don't need them all).
  Useful with e.g. microarray data. Not useful with e.g. image pixels.

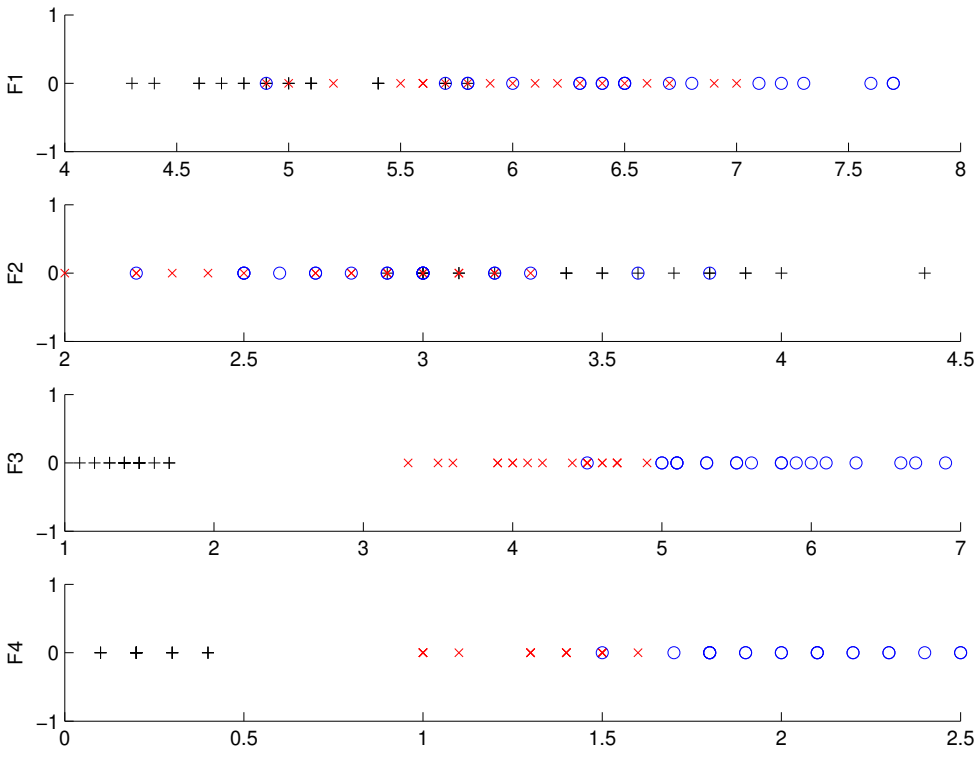
- *Best-subset selection*: for each subset of features, train a classifier and evaluate it on a validation set. Pick the subset having highest accuracy and up to L features.
  Combinatorial optimization: 2^D possible subsets of D features[?]. Ex: $D = 3$: $\{\varnothing, \{x_1\}, \{x_2\}, \{x_3\}, \{x_1, x_2\}, \{x_1, x_3\}, \{x_2, x_3\}, \{x_1, x_2, x_3\}\}$.
  Brute-force search only possible for small $D \Rightarrow$ approximate search.

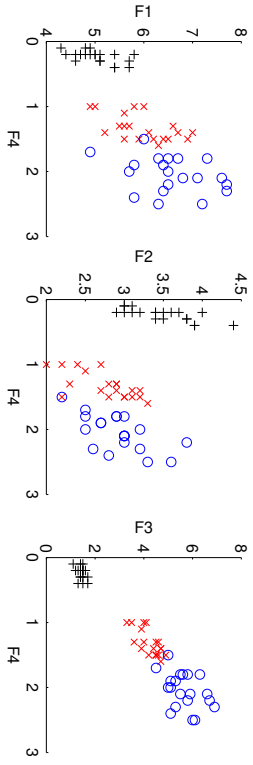
- *Forward selection*: starting with an empty subset $\mathcal{F}$, sequentially add one new feature at a time. We add the feature $d \in \{1, \dots, D\}$ such that the classifier trained on $\mathcal{F} \cup \{d\}$ has highest classification accuracy in the validation set. Stop when the accuracy improves little, or when we reach L features. *Backward selection*: same thing but start with $\mathcal{F} = \{1, \dots, D\}$ and remove one feature at a time.
  It is a greedy algorithm that is not guaranteed to find an optimal subset, but gives good results. It trains $\Theta(L^2)$ classifiers if we try up to L features, so it is convenient when we expect the optimal subset to contain few features.

- *Lasso* (for regression): $\min_{\mathbf{w}} \sum_{n=1}^N (y_n - \mathbf{w}^T \mathbf{x}_n)^2 + \lambda \|\mathbf{w}\|_1$ (where $\lambda \geq 0$ is set by cross-validation). The ℓ_1 norm $\|\mathbf{w}\|_1 = |w_1| + \cdots + |w_D|$ makes many w_d be exactly zero if λ is large enough.

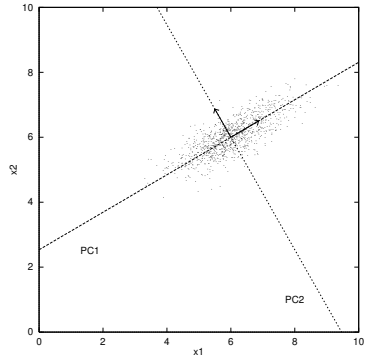
- These feature selection algorithms are *supervised*: they use the labels y_n when training the classifier. The features selected depend on the classifier we use. There are also unsupervised algorithms.

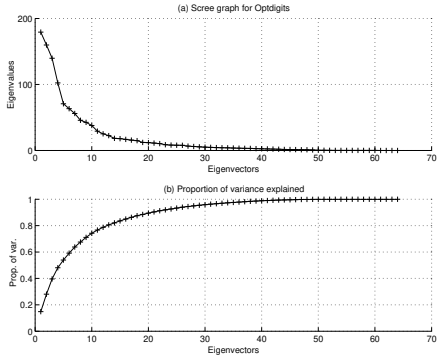
Ex: forward selection on the Iris dataset ($D = 4$ features, $K = 3$ classes). Result: features {F4,F3}.

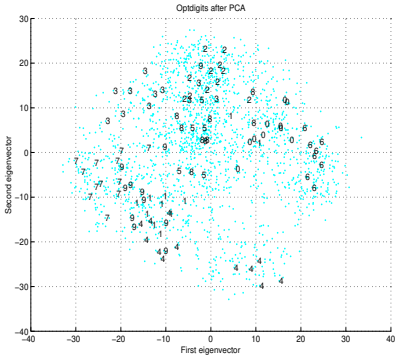




Dimensionality reduction: principal component analysis (PCA)



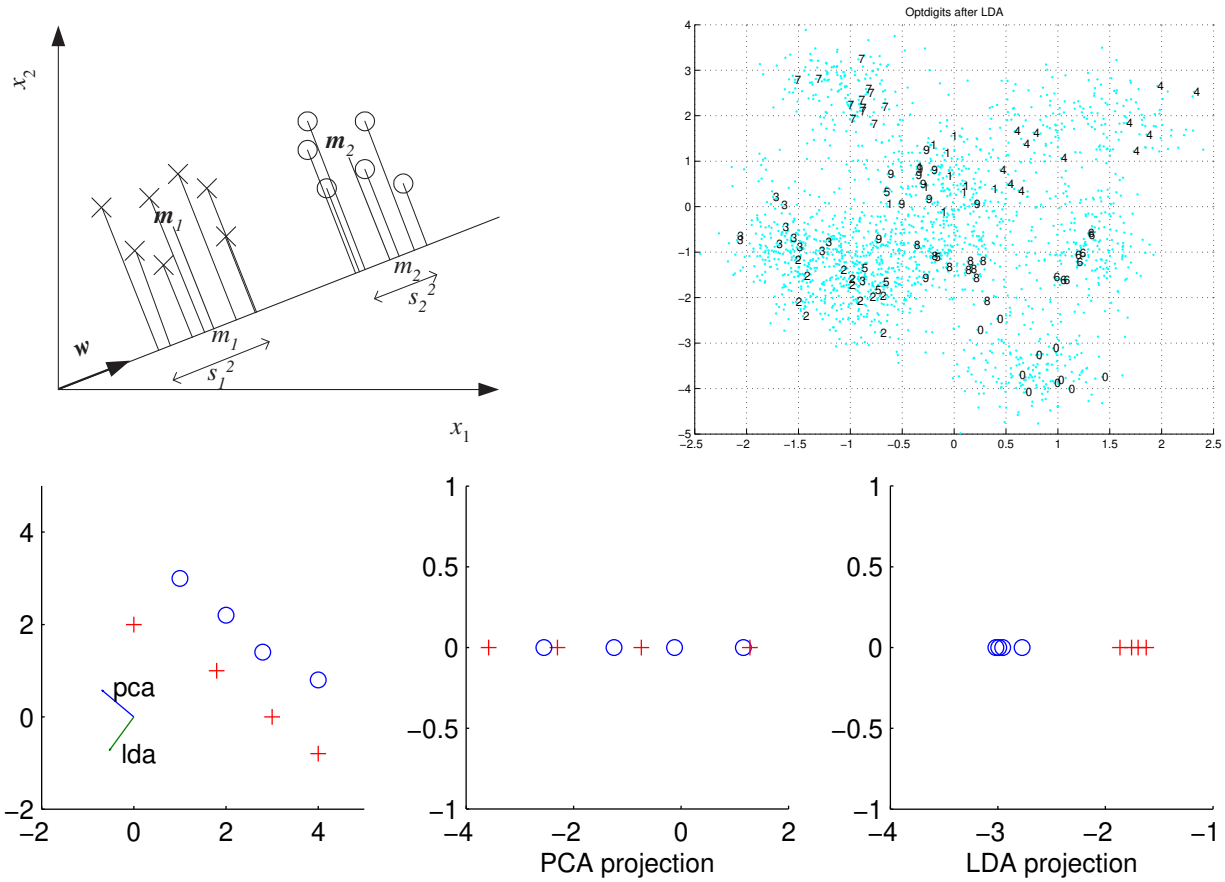




- Aims at preserving most of the signal information.
  Find a low-dimensional space such that when $\mathbf{x}$ is projected there, information loss is minimized.
- Which direction $\mathbf{w} \in \mathbb{R}^D$ shows most variation? $\max_{\mathbf{w}} \mathbf{w}^T \boldsymbol{\Sigma} \mathbf{w}$ s.t. $\|\mathbf{w}\| = 1 \Rightarrow \mathbf{w} = \mathbf{u}_1$[?]. p. 120
- Unsupervised linear DR method: given $\{\mathbf{x}_n\}_{n=1}^N \subset \mathbb{R}^D$ (with mean zero and covariance matrix $\boldsymbol{\Sigma}$ of $D \times D$), when reducing dimension to $L < D$, PCA finds:
  - a *linear projection mapping* $\mathbf{F}$: $\mathbf{x} \in \mathbb{R}^D \to \mathbf{W}^T\mathbf{x} \in \mathbb{R}^L$, and
  - a *linear reconstruction mapping* $\mathbf{f}$: $\mathbf{z} \in \mathbb{R}^L \to \mathbf{W}\mathbf{z} \in \mathbb{R}^D$,

  where $\mathbf{W}_{D\times L}$ has orthonormal columns ($\mathbf{W}^T\mathbf{W} = \mathbf{I}$), that are optimal in two equivalent senses:
  - *Maximum projected variance*: $\max_{\mathbf{W}} \operatorname{tr}\left(\operatorname{cov}\left\{\mathbf{W}^T\mathbf{x}_1, \ldots, \mathbf{W}^T\mathbf{x}_N\right\}\right) \stackrel{?}{=} \operatorname{tr}\left(\mathbf{W}^T\boldsymbol{\Sigma}\mathbf{W}\right)$.
  - *Minimum reconstruction error*: $\min_{\mathbf{W}} \frac{1}{N}\sum_{n=1}^N \left\|\mathbf{x}_n - \mathbf{W}\mathbf{W}^T\mathbf{x}_n\right\|^2 \stackrel{?}{=} -\operatorname{tr}\left(\mathbf{W}^T\boldsymbol{\Sigma}\mathbf{W}\right) + \text{constant}$.
- The covariance in the latent space $\operatorname{cov}\{\mathbf{Z}\} \stackrel{?}{=} \mathbf{W}^T\boldsymbol{\Sigma}\mathbf{W}$ is diagonal[?]: *uncorrelated projections*.
- If the mean of the sample is $\boldsymbol{\mu} = \frac{1}{N}\sum_{n=1}^N \mathbf{x}_n \Rightarrow \mathbf{F}(\mathbf{x}) = \mathbf{W}^T(\mathbf{x} - \boldsymbol{\mu})$ and $\mathbf{f}(\mathbf{z}) = \mathbf{W}\mathbf{z} + \boldsymbol{\mu}$.
- How to compute $\mathbf{W}$, given $\boldsymbol{\Sigma}$? Eigenproblem $\max_{\mathbf{W}} \operatorname{tr}\left(\mathbf{W}^T\boldsymbol{\Sigma}\mathbf{W}\right)$ s.t. $\mathbf{W}^T\mathbf{W} = \mathbf{I}$ whose solution is given by the spectral theorem. Decompose $\boldsymbol{\Sigma} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$ with eigenvectors $\mathbf{U} = (\mathbf{u}_1 \ldots \mathbf{u}_D)$ and eigenvalues $\boldsymbol{\Lambda} = \operatorname{diag}(\lambda_1, \ldots, \lambda_D)$, sorted decreasingly. Then $\mathbf{W} = \mathbf{U}_{1:L} = (\mathbf{u}_1, \ldots, \mathbf{u}_L)$, i.e., the *eigenvectors associated with the largest L eigenvalues of the covariance matrix $\boldsymbol{\Sigma}$*.
- Total variance of the data: $\lambda_1 + \cdots + \lambda_D = \operatorname{tr}(\boldsymbol{\Sigma}) = \sigma_1^2 + \cdots + \sigma_D^2$.
  Variance "explained" by the latent space: $\lambda_1 + \cdots + \lambda_L = \operatorname{tr}\left(\mathbf{W}^T\boldsymbol{\Sigma}\mathbf{W}\right)$.
  We can use the proportion $\frac{\lambda_1+\cdots+\lambda_L}{\lambda_1+\cdots+\lambda_D} \in [0, 1]$ of explained variance to determine a good value for L (e.g. 90% of the variance, which usually will be achieved with $L \ll D$).
- In practice with high-dimensional data (e.g. images), a few principal components explain most of the variance if there are correlations among the features.
- Useful as a preprocessing step for classification/regression: $\begin{cases}\text{reduce the number of features}\\ \text{partly remove noise.}\end{cases}$
- Basic disadvantage: it fails with nonlinear manifolds.
- Related linear DR methods:
  - *Factor analysis*: essentially, a probabilistic version of PCA.
  - *Canonical correlation analysis (CCA)*: projects two sets of features $\mathbf{x}$, $\mathbf{y}$ onto a common latent space $\mathbf{z}$.
- Related nonlinear DR methods: *autoencoders* (based on neural nets), etc.

Dimensionality reduction: linear discriminant analysis (LDA)

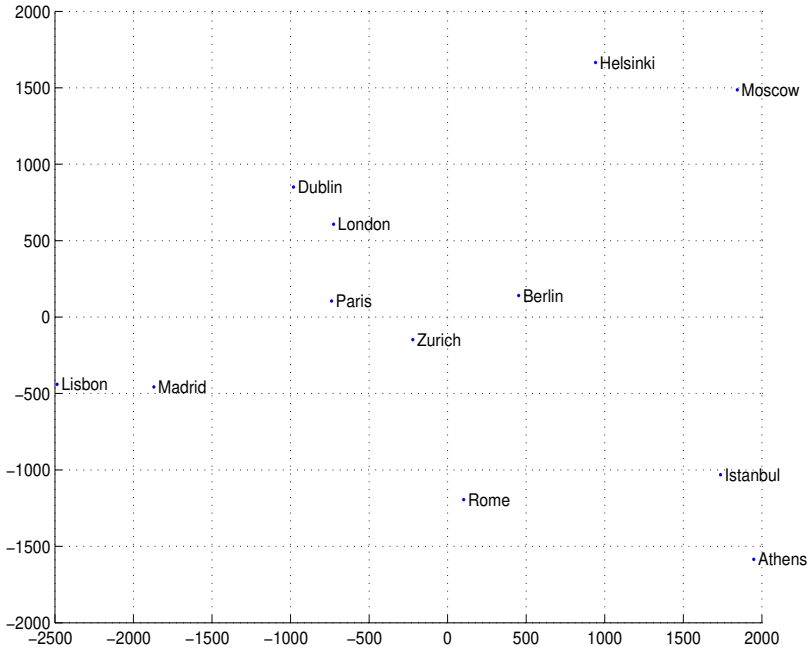


- Aims at preserving most of the signal information *that is useful to discriminate among the classes.*
  Find a low-dimensional space such that when $\mathbf{x}$ is projected there, classes are well separated.
- Supervised linear DR method: given $\{(\mathbf{x}_n, y_n)\}_{n=1}^N$ where $\mathbf{x}_n \in \mathbb{R}^D$ is a high-dimensional feature vector and $y_n \in \{1, \dots, K\}$ a class label, when reducing dimension to $L < D$, LDA finds a linear projection mapping $\mathbf{F}$: $\mathbf{x} \in \mathbb{R}^D \to \mathbf{W}^T\mathbf{x} \in \mathbb{R}^L$ with $\mathbf{W}_{D\times L}$ that is optimal in *maximally separating the classes from each other while maximally compressing each class.*
  Unlike PCA, LDA does not find a reconstruction mapping $\mathbf{f}$: $\mathbf{z} \in \mathbb{R}^L \to \mathbf{W}\mathbf{z} \in \mathbb{R}^D$. It only finds the projection mapping $\mathbf{F}$.
- Define:
  - Number of points in class k: N_k. Mean of class k: $\boldsymbol{\mu}_k = \frac{1}{N_k}\sum_{y_n=k}\mathbf{x}_n$.
  - *Within-class scatter matrix* for class k: $\mathbf{S}_k = \sum_{y_n=k}(\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T$.
  - *Total within-class scatter matrix* $\mathbf{S}_W = \sum_{k=1}^K \mathbf{S}_k$.
  - *Between-class scatter matrix* $\mathbf{S}_B = \sum_{k=1}^K N_k(\boldsymbol{\mu}_k - \boldsymbol{\mu})(\boldsymbol{\mu}_k - \boldsymbol{\mu})^T$ where $\boldsymbol{\mu} = \frac{1}{K}\sum_{k=1}^K \boldsymbol{\mu}_k$.
- In the latent space, the between-class and within-class scatter matrices are $\mathbf{W}^T\mathbf{S}_B\mathbf{W}$ and $\mathbf{W}^T\mathbf{S}_W\mathbf{W}$ (of $L \times L$).
- *Fisher discriminant*: $\max_{\mathbf{W}} J(\mathbf{W}) = \dfrac{|\mathbf{W}^T\mathbf{S}_B\mathbf{W}|}{|\mathbf{W}^T\mathbf{S}_W\mathbf{W}|} = \dfrac{\text{between-class scatter}}{\text{within-class scatter}}$.
- This is an eigenproblem whose solution is $\mathbf{W} = (\mathbf{u}_1, \dots, \mathbf{u}_L) =$ *eigenvectors associated with the largest* L *eigenvalues of* $\mathbf{S}_W^{-1}\mathbf{S}_B$.
  $\text{rank}(\mathbf{S}_B) \overset{?}{\leq} K-1 \Rightarrow \text{rank}(\mathbf{S}_W^{-1}\mathbf{S}_B) \leq K-1$. So we can only use values of L that satisfy $1 \leq L \leq K-1$.
  $\mathbf{S}_W$ must be invertible (if it is not, apply PCA to the data and eliminate directions with zero variance).

Dimensionality reduction: multidimensional scaling (MDS)

True distances along earth surface

Estimated positions on a 2D map



- Aims at preserving distances or similarities.
  Place N points in a low-dimensional map (of dimension L) such that their distances are well preserved.

- Unsupervised DR method: given the matrix of squared Euclidean distances $d_{nm}^2 = \|\mathbf{x}_n - \mathbf{x}_m\|^2$ between N data points, MDS finds points $\mathbf{z}_1, \dots, \mathbf{z}_N \in \mathbb{R}^L$ that approximate those distances:

$$\min_{\mathbf{Z}} \sum_{n,m=1}^{N} \left(d_{nm}^2 - \|\mathbf{z}_n - \mathbf{z}_m\|^2\right)^2.$$

- MDS does not use as training data the actual feature vectors $\mathbf{x}_n \in \mathbb{R}^D$, only the pairwise distances d_{nm}. Hence, it is applicable even when the "distances" are computed between objects that are not represented by features. Ex: perceptual distance between two different colors according to a subject.

- If $d_{nm}^2 = \|\mathbf{x}_n - \mathbf{x}_m\|^2$ where $\mathbf{x}_n \in \mathbb{R}^D$ and $D \geq L$, then MDS is equivalent to PCA on $\{\mathbf{x}_n\}_{n=1}^N$. p. 137

- MDS does not produce a projection or reconstruction mapping, only the actual L-dimensional projections $\mathbf{z}_1, \dots, \mathbf{z}_N \in \mathbb{R}^L$ for the N training points $\mathbf{x}_1, \dots, \mathbf{x}_N \in \mathbb{R}^D$.

- How to learn a projection mapping $\mathbf{F}$: $\mathbf{x} \in \mathbb{R}^D \to \mathbf{z} \in \mathbb{R}^L$ with parameters $\boldsymbol{\Theta}$?

  Direct fit: find the projections $\mathbf{z}_1, \dots, \mathbf{z}_N$ by MDS and then solve a nonlinear regression

$$\min_{\boldsymbol{\Theta}} \sum_{n=1}^{N} (\mathbf{z}_n - \mathbf{F}(\mathbf{x}_n; \boldsymbol{\Theta}))^2$$

  Parametric embedding: requires nonlinear optimization

$$\min_{\boldsymbol{\Theta}} \sum_{n,m=1}^{N} \left(d_{nm}^2 - \|\mathbf{F}(\mathbf{x}_n; \boldsymbol{\Theta}) - \mathbf{F}(\mathbf{x}_m; \boldsymbol{\Theta})\|^2\right)^2.$$

- Generalizations of MDS:
  - *Spectral methods*: Isomap, Locally Linear Embedding, Laplacian eigenmaps...
    Require solving an eigenproblem.
    Isomap: define d_{nm} = geodesic distances (approximated by shortest paths in a nearest-neighbor graph of the sample).
  - *Nonlinear embeddings*: elastic embedding, t-SNE...
    Require solving a nonlinear optimization.