IN 5520 – Repetition

Anne Solberg (anne@ifi.uio.no)

3.12.20

- Gaussian classifiers – briefly
- Support vector machines
- Feature selection and transforms

Approaching a classification problem

- Collect and label data
- Get to know the data: exploratory data analysis
- Choose features
- Consider preprocessing/normalization
- Choose classifier
  - Estimate classifier parameters on training data
- Estimate hyperparameters on validation data
  - Alternative: cross-validation on the training data set
- Compute the accuracy on test data

The conditional density p($\mathbf{x}|\ \omega_s$)

- Any probability density function can be used to model p($\mathbf{x}|\ \omega_s$)
- A common model is the multivariate Gaussian density.
- The multivariate Gaussian density:

$$p(\mathbf{x} \mid \omega_s) = \frac{1}{(2\pi)^{n/2} |\mathbf{\Sigma}_s|^{1/2}} \exp\left[-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_s)^t \mathbf{\Sigma}_s^{-1} (\mathbf{x} - \boldsymbol{\mu}_s) \right]$$

- If we have d features, μ_s is a vector of length d and and Σ_s a d$\times$d matrix (depends on class *s*)

$$\boldsymbol{\mu}_S = \begin{bmatrix} \mu_{1s} \\ \mu_{2s} \\ \\ \\ \mu_{ns} \end{bmatrix} \qquad \mathbf{\Sigma}_S = \begin{bmatrix} \sigma_{11} & \sigma_{12} & . & . & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} & . & . & . \\ \sigma_{31} & \sigma_{11} & . & . & . \\ . & . & . & . & . \\ \sigma_{n1} & \sigma_{n2} & . & \sigma_{nn-1} & \sigma_{nn} \end{bmatrix}$$

Symmetric d$\times$d matrix
σ_{ii} is the variance of feature i
σ_{ij} is the covariance between feature i and feature j
Symmetric because $\sigma_{ij} = \sigma_{ji}$

- $|\Sigma_s|$ is the determinant of the matrix Σ_s, and Σ_s^{-1} is the inverse

Discriminant functions for the normal density

- We saw last lecture that the minimum-error-rate classification can be computed using the discriminant functions

$$g_i(\mathbf{x}) = \ln p(\mathbf{x} \mid \omega_i) + \ln P(\omega_i)$$

- With a multivariate Gaussian we get:

$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^t \Sigma_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i) - \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln|\Sigma_i| + \ln P(\omega_i)$$

- Let ut look at this expression for some special cases:

Case 1: $\Sigma_j = \sigma^2 I$

- The discriminant functions simplifies to **linear** functions using such a shape on the probability distributions

$$g_j(\mathbf{x}) = -\frac{1}{2(\sigma^2 I)}(\mathbf{x}-\boldsymbol{\mu}_j)^T(\mathbf{x}-\boldsymbol{\mu}_j) - \frac{d}{2}\ln(2\pi) - \frac{1}{2}\ln\left|\sigma^2 I\right| + \ln P(\omega_j)$$

$$= -\frac{1}{2(\sigma^2 I)}(\mathbf{x}^T\mathbf{x} - 2\boldsymbol{\mu}_j^T\mathbf{x} + \boldsymbol{\mu}_j^T\boldsymbol{\mu}_j) - \frac{d}{2}\ln(2\pi) - \frac{1}{2}\ln\left|\sigma^2 I\right| + \ln P(\omega_j)$$

Common for all classes, no need to compute these terms

Since $\boldsymbol{x}^T\boldsymbol{x}$ is common for all classes, an equivalent $g_j(\boldsymbol{x})$ is a linear function of $\boldsymbol{x}$: .

$$\frac{1}{(\sigma^2)}\boldsymbol{\mu}_j^T\mathbf{x} - \frac{1}{2(\sigma^2)}\boldsymbol{\mu}_j^T\boldsymbol{\mu}_j + \ln P(\omega_j)$$

Case 2: Common covariance, $\Sigma_j = \Sigma$

- An equivalent formulation of the discriminant functions is

$$g_i(\mathbf{x}) = \mathbf{w}_i^t \mathbf{x} + wi_0$$
$$\text{where } \mathbf{w}_i = \mathbf{\Sigma}^{-1}\boldsymbol{\mu}_i$$
$$\text{and } wi_0 = -\frac{1}{2}\boldsymbol{\mu}_i^t \mathbf{\Sigma}^{-1}\boldsymbol{\mu}_i + \ln P(\omega_i)$$

- The decision boundaries are again hyperplanes.
- The decision boundary has the equation:

$$\mathbf{w}^T(\mathbf{x} - \mathbf{x}_0) = 0$$
$$\mathbf{w} = \Sigma^{-1}(\mu_i - \mu_j)$$
$$x_0 = \frac{1}{2}(\mu_i + \mu_j) - \frac{\ln\left[(P(\omega_i)/(P(\omega_j)\right]}{(\mu_i - \mu_j)^T \Sigma^{-1}(\mu_i - \mu_j)}(\mu_i - \mu_j)$$

- Because $\mathbf{w}_i = \mathbf{\Sigma}^{-1}(\mu_i - \mu_j)$ is not in the direction of $(\mu_i - \mu_j)$, the hyperplane will not be orthogonal to the line between the means.

Case 3:, Σ_j=arbitrary

- The discriminant functions will be quadratic:

$$g_i(\mathbf{x}) = \mathbf{x}^t \mathbf{W}_i \mathbf{x} + \mathbf{w}_i^t \mathbf{x} + wi_0$$

$$\text{where } \mathbf{W}_i = -\frac{1}{2}\mathbf{\Sigma}_i^{-1}, \qquad \mathbf{w}_i = \mathbf{\Sigma}_i^{-1}\boldsymbol{\mu}_i$$

$$\text{and } wi_0 = -\frac{1}{2}\boldsymbol{\mu}_i{}^t \mathbf{\Sigma}_i^{-1}\boldsymbol{\mu}_i - \frac{1}{2}\ln\left|\mathbf{\Sigma}_i\right| + \ln P(\omega_i)$$

- The decision surfaces are hyperquadrics and can assume any of the general forms:
  - hyperplanes
  - hypershperes
  - pairs of hyperplanes
  - hyperellisoids,
  - Hyperparaboloids,..
- The next slides show examples of this.
- In this general case we cannot intuitively draw the decision boundaries just by looking at the mean and covariance.

Support Vector Machine classifiers

Cost function – nonseparable case

- The cost function to minimize is now

$$J(w, w_0, \xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} I(\xi_i)$$

$$\text{where} \qquad \mathrm{I}(\xi_i) = \begin{cases} 1 & \xi_\mathrm{i} > 0 \\ 0 & \xi_\mathrm{i} = 0 \end{cases}$$

and ξ is the vector of parameters ξ_i.

- C is a parameter that controls how much misclassified training samples is weighted.
- We skip the mathematics and present the alternative dual formulation:

$$\max_{\lambda}\left(\sum_{i=1}^{N}\lambda_i - \frac{1}{2}\sum_{i,j}\lambda_i\lambda_j y_i y_j x_i^T x_j\right)$$

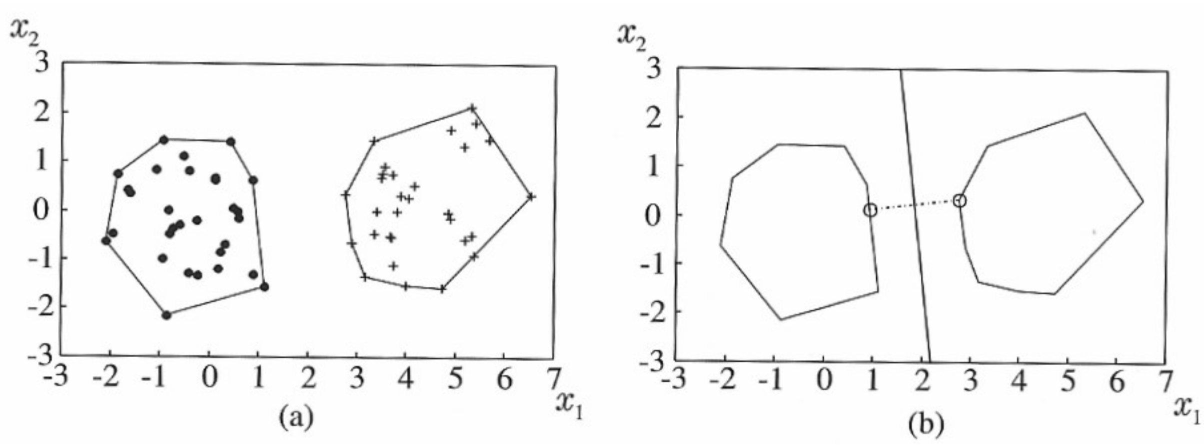
$$\text{subject to} \quad \sum_{\mathrm{i}=1}^{\mathrm{N}}\lambda_\mathrm{i} y_i = 0 \quad \text{and} \quad 0 \le \lambda_i \le C \ \ \forall i$$

- All points between the two hyperplanes (ξ_i>0) can be shown to have λ_i=C.

SVM: A geometric view

- SVMs can be related to the convex hull of the different classes. Consider a class that contains training samples $X=\{x_1,...x_N\}$.
- The convex hull of the set of points in X is given by all convex combinations of the N elements in X.

  - A region R is convex if and only if for any two points x_1,x_2 in R, the whole line segment between x_1 and x_2 is inside the R.
  - The convex hull of a region is the smalles convex region H which satisfies the conditions $R \subseteq H$.



- The convex hull for a class is the smallest convex set that contains all the points in the class (X).
- Searching for the hyperplane with the highest margin is equivalent to searching for the two nearest points in the two convex sets.
  - This can be proved, but we just take the result as an aid to get a better visual interpretation of the SVM hyperplane.

SVMs and kernels

- Note that in both the optimization problem and the evaluation function, g(**x**), the samples come into play as inner products only

$$\max_{\lambda}\left(\sum_{i=1}^{N}\lambda_i - \frac{1}{2}\sum_{i,j}\lambda_i\lambda_j y_i y_j x_i^T x_j\right)$$
$$\text{s.t.}\quad 0 \le \lambda_i \le C \quad \forall i$$
$$\sum_{i=1}^{N}\lambda_i y_i = 0$$

$$g(x) = w^T x + w_0 = \sum_{i=1}^{N_s}\lambda_i y_i x^T x_i + w_0$$

Called «kernel»

- If we have a function evaluating inner products, K($\mathbf{x}_i$,$\mathbf{x}_j$), we can ignore the samples themselves
- Let's say we have K($\mathbf{x}_i$,$\mathbf{x}_j$) evaluating inner products in a higher dimensional space:

  -> no need to do the mapping of our samples explicitly!

Useful kernels for classification

- Polynomial kernels

$$K(x,z) = \left(x^T z - 1\right)^q, \quad q > 0$$

- Radial basis function kernels (very commonly used!)

$$K(x,z) = \exp\left(-\frac{\|x-z\|^2}{\sigma^2}\right)$$

Note the we need to set the σ parameter

The «support» of each point is controlled by σ.

- Hyperbolic tangent kernels (often with β=2 and γ=1)

The inner product is related to the similarity of the two samples.

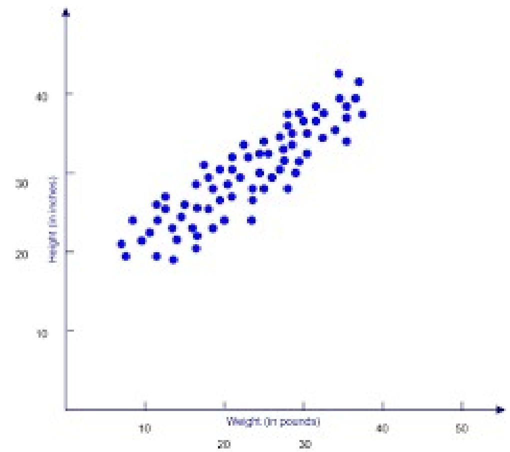
$$K(x,z) = \tanh\left(\beta x^T \, z + \gamma\right)$$

The kernels give inner-product evaluations in the, possibly infinite-dimensional, transformed space.

The kernel inputs need not be numeric, e.g. kernels for text strings are possible.

Idea behind (Principal Component Transform)

- Find a projection **y**=A^T**x** of the feature vector **x**
- Three interpretations of PCA:
  - Find the projection that maximize the variance along the selected projection
  - Minimize the reconstruction error (squared distance between original and transformed data)
  - Find a transform that gives uncorrelated features



Criterion function

- Goal: Find transform minimizing representation error

- We start with a single weight-vector, **w**, giving us a single feature, y_1

- Let $J(\mathbf{w}) = \mathbf{w}^T\mathbf{R}\mathbf{w} = \sigma_w^2$

Maximizing this is equivalent to minimizing representation error

- Now, let's find
  - $\max_{\boldsymbol{w}} J(\boldsymbol{w})$
  - $s.t. \|\boldsymbol{w}\| = 1$

- Transform this problem into a unconstrained problem with a Lagrange multiplier (we skip details here)

Maximizing variance of y_1

$$\mathcal{L}(\mathbf{w},\lambda) \equiv \sigma_{\mathbf{w}}^2 - \lambda(\mathbf{w}^T\mathbf{w} - 1)$$

$$\frac{\partial L}{\partial \lambda} = \mathbf{w}^T\mathbf{w} - 1$$

$$\frac{\partial L}{\partial \mathbf{w}} = 2R\mathbf{w} - 2\lambda\mathbf{w}$$

•Lagrangian function for maximizing σ^2_w with the constraint $\mathbf{w}^T\mathbf{w}$=1

$\Downarrow$ •Equating zero

$$\mathbf{w}^T\mathbf{w} = 1$$

$$R\mathbf{w} = \lambda\mathbf{w}$$

The maximizing **w** is an eigenvector of R!

And $\sigma^2_w=\lambda$!

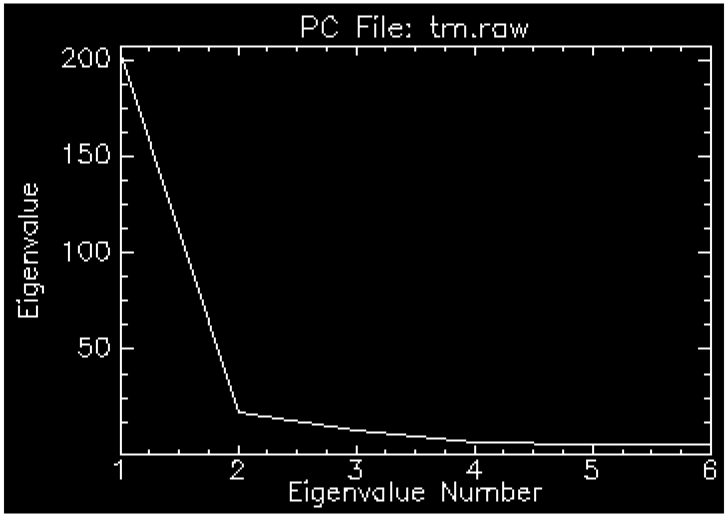
Principal component transform (PCA)

- Place the *m* «principle» eigenvectors (the ones with the largest eigenvalues) along the columns of A
  - They are given as the eigenvectors of the covariance matrix R
- Then the transform $\mathbf{y} = \mathbf{A}^T\mathbf{x}$ gives you the *m* first principle components

Example cont: Inspecting the eigenvalues

- The mean-square representation error we get with m of the N PCA-components is given as

$$E\left[\|x-\hat{x}\|^2\right]=\sum_{i=1}^{N-1}\lambda_i-\sum_{i=1}^{m}\lambda_i=\sum_{i=m}^{N-1}\lambda_i$$



- Plotting λ s will give indications on how many features are needed for representation