



UiO : Department of Informatics
University of Oslo

Biologically inspired computing – Lecture 19 October 2015

Support Vector Machines (Marshall Chpt 8)
Ensembles (Marshall Chpt 13)
Dimensionality (Marshall Chpt 6.2)

UiO : Department of Informatics
University of Oslo

This lecture

1. Support vector machines
 - Optimal separation
 - Kernels
2. Ensemble learning
3. Dimensionality reduction
 - Principal component analysis

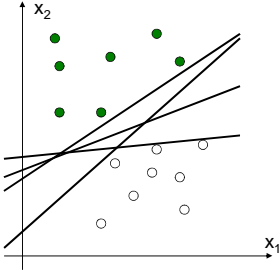
3

UiO : Department of Informatics
University of Oslo

Optimal separation

Linear separators:

Which one is best?



4

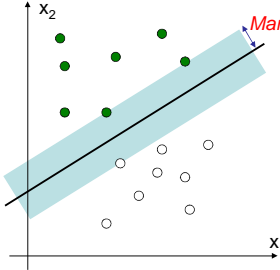
UiO : Department of Informatics
University of Oslo

Optimal separation

Choose the one with the best margin!

Why?

- New data near the training data points will likely be of the same class



5

UiO : Department of Informatics
University of Oslo

Optimal separation

Support vectors

- The training data defining the margin
- The rest of the data can be discarded when we are done learning

6

UiO : Department of Informatics
University of Oslo

Optimal separation

- Distance to hyperplane:

$$w \cdot x_i - b = \begin{cases} > 0 & \text{above plane - class A: } y_i = 1 \\ < 0 & \text{below plane - class B: } y_i = -1 \end{cases}$$
- If we require that $y_i(w \cdot x_i - b) \geq 1$ then the margin is $M = 1/(2|w|)$
 - Maximizing the margin \Leftrightarrow minimizing $w \cdot w$
 - Exact solution can be found, along with a list of support vectors, using quadratic programming
- SVM in 7 minutes (Thales Sehn Körting):
<https://www.youtube.com/watch?v=1NxnPkZM9bc>

7

UiO : Department of Informatics
University of Oslo

Nonlinearity

- How to classify linearly inseparable data?
 - Combine many linear SVMs?
 - Similar to multilayer neural networks
 - But what are the target outputs for the hidden layers?
 - A different idea:
 - Map inputs into a higher-dimensional space
 - Hope that they are linearly separable there.

8

UiO : Department of Informatics
University of Oslo

Increase dimensionality

$\varphi: (x) \rightarrow (x, x^2)$

9

High dimensionality

- SVMs typically map to feature spaces of much higher dimension
 - With enough dimensions, it becomes very likely that the data becomes linearly separable

10

Kernels

- Finding the hyperplane only requires the dot product between vectors, not the actual vectors
 - Calculating $\varphi(x_i) \cdot \varphi(x_j)$ might be much easier than $\varphi(x_i)$
- $K(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j)$ is called the *kernel* of φ
 - Common kernels include
 - None: $K(x_i, x_j) = x_i \cdot x_j$
 - Polynomial: $K(x_i, x_j) = (1 + x_i \cdot x_j)^p$
 - Sigmoid: $K(x_i, x_j) = \tanh(kx_i \cdot x_j - \delta)$
 - Radial basis function: $K(x_i, x_j) = \exp(-(x_i - x_j)^2 / 2\sigma^2)$

11

Overfitting

- Any data set is linearly separable in a feature space of sufficient complexity
- We have to be aware of overfitting: Use cross-validation and early stopping!
 - If there are noisy outliers (esp. mislabeled examples), we need to take stronger measures: soft margin.

12

Soft margins

- Instead of perfectly separating all data, **allow** some **misclassifications**
- Introduce slack variables
 - Optimize tradeoff between maximum margin and misclassification penalty
 - Tradeoff is balanced by **penalty factor C**
- Useful when some error is tolerated, or when there are chances of mislabeled training data

13

UiO Department of Informatics
University of Oslo

Applications

- Classification
 - Multi-class can be achieved via multiple outputs
- Regression
- Object detection & recognition
- Content-based image retrieval
- Text recognition
- Speech recognition
- Biometrics
- Etc.

14

UiO Department of Informatics
University of Oslo

Considerations

- Quite powerful (hard to beat by other algorithms)
 - Must beware of overfitting
- Robust to some noise, if margin is managed properly
- Fast to apply
- Difficult to interpret
- How to pick kernel?
 - Start with Gaussian RBF or polynomial
 - May require domain-specific knowledge
 - Can combine kernels for heterogeneous data
 - Consult experts

15

UiO Department of Informatics
University of Oslo

Ensemble learning

The diagram illustrates the ensemble learning process. At the bottom, three scatter plots show input patterns in a 2D space with axes x_1 and x_2 . The first plot shows a cluster of points. The second plot shows two overlapping shaded regions representing different classifiers' decision boundaries. The third plot shows the combined result. Above the plots, arrows labeled 'Input Pattern' point to two boxes, 'Classifier 1' and 'Classifier N'. From these boxes, arrows point to a central 'Combine' box, which then points to a final 'Decision' box.

16

UiO Department of Informatics
University of Oslo

Ensemble learning

- “Decision by committee”
 - Train multiple classifiers to be slightly different
 - An “ensemble”
 - Make classifications based on the combined results of all of them
- Two common types of training differentiation
 - **Boosting**: change the importance of each training vector (data point)
 - **Bagging**: change the training vectors being used

17

Boosting - AdaBoost

- Iteratively trains classifiers
- Each data point is assigned a weight
 - For the first classifier all the weights are equal
 - For the next classifier the weights of the data points that were misclassified previously is raised
 - This is continued until the combined error of the classifiers trained so far is sufficiently low
- Dependent on the classifier's ability to consider the weights in their training

18

Bagging

- Makes a random sample of the training data for each classifier – *bootstrap samples*
 - Same size as the training data
 - *With replacement*
 - Some data points will occur at least twice!
 - Variance will be reduced
 - Each classifier will have different views of the training data

19

Combining the classifiers

- Which classifiers do we listen to when the ensemble is in disagreement?
 - **Weighted voting** (used in boosting)
 - Some classifiers have greater influence than others
 - **Majority voting** (used in bagging)
 - The most "popular" class is chosen
 - **Mixture of experts**
 - A meta-machine learning algorithm decides which classifiers are most likely to be correct

20

Majority voting

What to do about the disagreement

- Refuse to classify?
- Classify only if more than half agree?
- Return the most common vote?

Depends on the application

21

Dimensionality reduction – Feature extraction

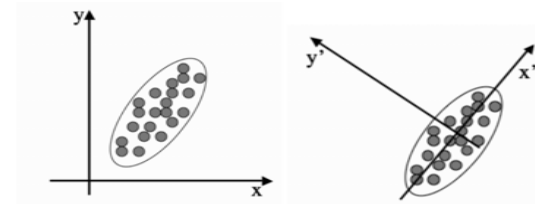
Why reduce dimensionality?

- Reduces time complexity: Less computation
- Reduces space complexity: Less parameters
- Saves the cost of acquiring irrelevant features
- Simpler models are more robust
- Easier to interpret; simpler explanation
- Data visualization (structure, groups, outliers, etc.) if plotted in 2 or 3 dimensions

22

Principal components

- The directions along with the most variation
 - Don't have to correspond to the coordinate axes



23

YouTube introductions

Application examples (Rasmus Bro):

- https://www.youtube.com/watch?annotation_id=annotation_963680&feature=iv&src_vid=K-F19DORO1w&v=UUxIXU_Ob6E
- <https://www.youtube.com/watch?v=26YhtSJ1qc>

24

Principal component analysis

- Rotate the axes to lie along the principal components
- Remove the axes with the least variation
 - Keep a certain number of dimensions
 - Or: keep a certain percentage of the variation

25

Calculating the principal components

- Calculate the covariance matrix of the data
- Calculate the eigenvalues and eigenvectors of the covariance matrix
- Transform the data with the eigenvectors for the largest eigenvalues as the new basis

26

Calculating the covariance matrix

The variance of feature i :

$$\sigma_i^2 = \sigma_{ii} = \frac{1}{N} \sum_{k=1}^N (x_{ki} - \mu_i)^2$$

The covariance between feature i and j :

$$\sigma_{ij} = \frac{1}{N} \sum_{k=1}^N (x_{ki} - \mu_i)(x_{kj} - \mu_j)$$

27

Calculating the covariance matrix

The covariance matrix is composed of the variances and covariances of every combination of feature:

$$\begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \cdots & \sigma_{nn} \end{bmatrix}$$

28

The covariance eigenvectors

The eigenvectors \mathbf{v}_i and eigenvalues λ_i are the n unique values of matrix \mathcal{C} such that

$$\lambda_i \mathbf{v}_i = \mathcal{C} \mathbf{v}_i$$

- The eigenvectors of the covariance matrix describe the directions of the principal components
- The eigenvalues tell us how large part of the total variation in the data that is accounted for by that principal component

29

Notes on PCA

- PCA is a linear transformation
 - Does not directly help with data that is not linearly separable
 - However, may make learning easier because of reduced complexity
- PCA removes some information from the data
 - Might just be noise
 - Might provide helpful nuances that may be of help to some classifiers