Machine Learning and Pervasive Computing

Stephan Sigg

Georg-August-University Goettingen, Computer Networks

01.06.2015

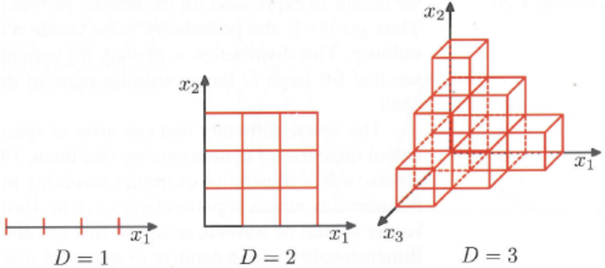
Overview and Structure

13.04.2015	Organisation
13.04.2015	Introduction
20.04.2015	Rule-based learning
27.04.2015	Decision Trees
04.05.2015	A simple Supervised learning algorithm
11.05.2015	–
18.05.2015	Excursion: Avoiding local optima with random search
25.05.2015	–
01.06.2015	High dimensional data
08.06.2015	Artificial Neural Networks
15.06.2015	k-Nearest Neighbour methods
22.06.2015	Probabilistic models
29.06.2015	Topic models
06.07.2015	Unsupervised learning
13.07.2015	Anomaly detection, Online learning, Recom. systems

Outline

Issues related to high dimensional input data

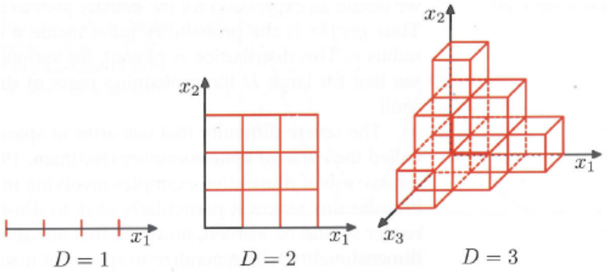
Exponential growth When dividing the space into bins with fixed side-length, the number of bins grows exponentially with dimension



Issues related to high dimensional input data

Exponential growth When dividing the space into bins with fixed side-length, the number of bins grows exponentially with dimension

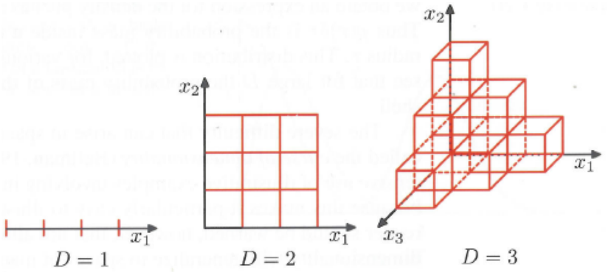
> To capture a distribution underlying some process, sufficient number of samples for all relevant regions in the feature space are required



Issues related to high dimensional input data

Exponential growth When dividing the space into bins with fixed side-length, the number of bins grows exponentially with dimension

Counter-intuitive properties Higher dimensional spaces can have counter-intuitive properties (see example on next slides)



The curse of dimensionality

Example – Volume of a sphere

Consider a sphere of radius $r = 1$ in a D-dimensional space

The curse of dimensionality

Example – Volume of a sphere

Consider a sphere of radius $r = 1$ in a D-dimensional space

What is the fraction of the volume of the sphere that lies between radius $r = 1$ and $r' = 1 - \varepsilon$?

The curse of dimensionality

Example – Volume of a sphere

Consider a sphere of radius $r = 1$ in a D-dimensional space

What is the fraction of the volume of the sphere that lies between radius $r = 1$ and $r' = 1 - \varepsilon$?

We can estimate the volume of a shpere with radius r as

$$V_D(r) = \delta_D r^D$$

for appropriate δ

The curse of dimensionality

Example – Volume of a sphere

We can estimate the volume of a shpere with radius r as

$$V_D(r) = \delta_D r^D$$

for appropriate δ

The curse of dimensionality

Example – Volume of a sphere

We can estimate the volume of a shpere with radius r as

$$V_D(r) = \delta_D r^D$$

for appropriate δ

The required fraction is given by

$$\frac{V_D(1) - V_D(1-\varepsilon)}{V_D(1)} = 1 - (1-\varepsilon)^D$$

The curse of dimensionality

Example – Volume of a sphere

The required fraction is given by

$$\frac{V_D(1) - V_D(1-\varepsilon)}{V_D(1)} = 1 - (1-\varepsilon)^D$$

The curse of dimensionality

Example – Volume of a sphere

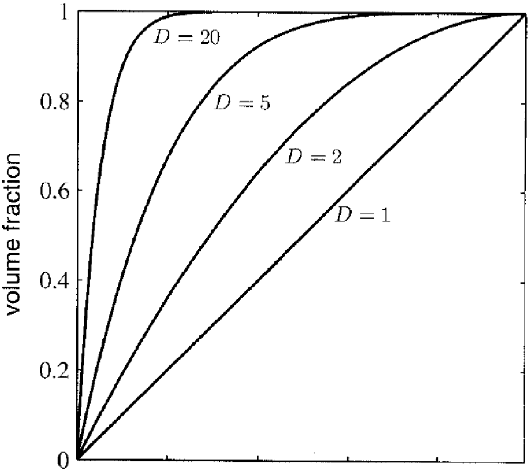
The required fraction is given by

$$\frac{V_D(1) - V_D(1-\varepsilon)}{V_D(1)} = 1 - (1-\varepsilon)^D$$

For large D, this fraction tends to 1

In high dimensional spaces, most of the volume of a sphere is concentrated near the surface

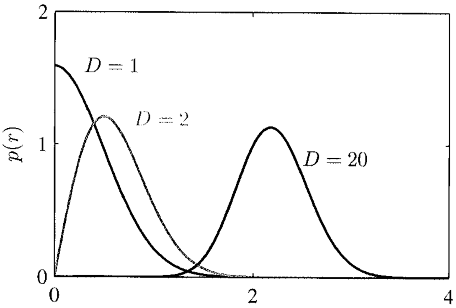
The curse of dimensionality



The curse of dimensionality

Example – Gaussian distribution

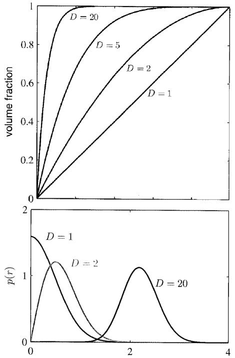
The probability mass of the gaussian distribution is concentrated in a thin shell (here plotted as distance from the origin in a polar coordinate system)

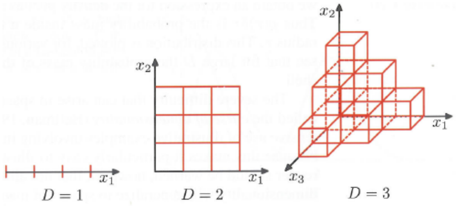


The curse of dimensionality

Discussion

While the curse of dimensionality induces problems, we will investigate effective techniques applicable to high-dimensional spaces





Outline

High dimensional data

Dimensionality reduction

High dimensional data (data with numerous features) not appreciated in general

- $\rightarrow$ slows down classification algorithms
- $\rightarrow$ easier to visualise
- $\rightarrow$ Remove redundant features (e.g. distance travelled $\leftrightarrow$ steps)

High dimensional data

Dimensionality reduction

Principal Component Analysis

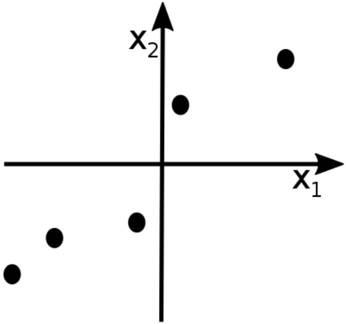
Find lower dimensional surface onto which to project the data

High dimensional data

Dimensionality reduction

Principal Component Analysis

Find lower dimensional surface onto which to project the data

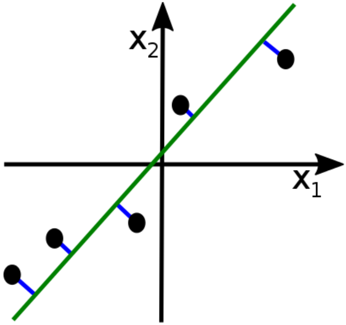


High dimensional data

Dimensionality reduction

Principal Component Analysis

Find lower dimensional surface onto which to project the data

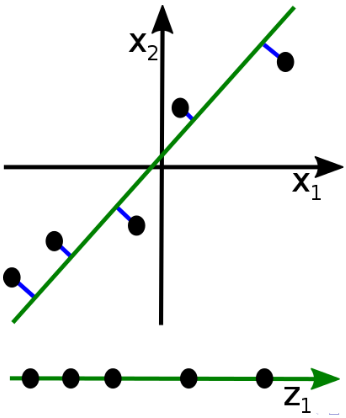


High dimensional data

Dimensionality reduction

Principal Component Analysis

Find lower dimensional surface onto which to project the data



High dimensional data

Dimensionality reduction

PCA finds k vectors $v^{(1)}, \ldots, v^{(k)}$ onto which to project the data such that the projection error is reduced.

High dimensional data

Dimensionality reduction

PCA finds k vectors $v^{(1)}, \ldots, v^{(k)}$ onto which to project the data such that the projection error is reduced.

→ In particular, we find values $z^{(i)}$ to represent the $x^{(i)}$ in this k-dimensional vector space spanned by the $v^{(i)}$

High dimensional data

Dimensionality reduction

1. Compute the covariance matrix from the $x^{(i)}$:

$$\Sigma = \frac{1}{m}\sum_{i=1}^{n} \underbrace{\underbrace{\left(x^{(i)}\right)}_{1\times m\text{-dim.}} \underbrace{\left(x^{(i)}\right)^T}_{m\times 1\text{-dim.}}}_{m\times m\text{-dim.}}$$

High dimensional data

Dimensionality reduction

1. Compute the covariance matrix from the $x^{(i)}$:

$$\Sigma = \frac{1}{m}\sum_{i=1}^{n} \underbrace{\underbrace{\left(x^{(i)}\right)}_{1\times m\text{-dim.}} \underbrace{\left(x^{(i)}\right)^{T}}_{m\times 1\text{-dim.}}}_{m\times m\text{-dim.}}$$

Covariance

A measure of spread of a set of points around their center of mass

High dimensional data

Dimensionality reduction

1. Compute the covariance matrix from the $x^{(i)}$:

$$\Sigma = \frac{1}{m} \sum_{i=1}^{n} \underbrace{\underbrace{\left(x^{(i)}\right)}_{1\times m\text{-dim.}} \underbrace{\left(x^{(i)}\right)^T}_{m\times 1\text{-dim.}}}_{m\times m\text{-dim.}}$$

2. The pricipal components are found by computing the eigenvectors and eigenvalues of Σ (solving equation $(\Sigma - \lambda I_m)u = 0$)

High dimensional data

Dimensionality reduction

When a matrix Σ is multiplied with a vector u', this usually results in a new vector $\Sigma u'$ of different direction than u'.

2. The pricipal components are found by computing the eigenvectors and eigenvalues of Σ (solving equation $(\Sigma - \lambda I_m)u = 0$)

High dimensional data

Dimensionality reduction

When a matrix Σ is multiplied with a vector u', this usually results in a new vector $\Sigma u'$ of different direction than u'.

$\rightarrow$ There are few vectors u, however, which have the same direction ($\Sigma u = \lambda u$).

These are the eigenvectors of Σ and λ are the eigenvalues

2. The pricipal components are found by computing the eigenvectors and eigenvalues of Σ (solving equation $(\Sigma - \lambda I_m)u = 0$)

High dimensional data

Dimensionality reduction

1. Compute the covariance matrix from the $x^{(i)}$:

$$\Sigma = \frac{1}{m} \sum_{i=1}^{n} \underbrace{\underbrace{\left(x^{(i)}\right)}_{1\times m\text{-dim.}} \underbrace{\left(x^{(i)}\right)^T}_{m\times 1\text{-dim.}}}_{m\times m\text{-dim.}}$$

2. The pricipal components are found by computing the eigenvectors and eigenvalues of Σ (solving equation $(\Sigma - \lambda I_m)u = 0$)

Eigenvectors and Eigenvalues

The (orthogonal) eigenvectors are sorted by their eigenvalues with respect to the direction of greatest variance in the data.

High dimensional data

Dimensionality reduction

1. Compute the covariance matrix from the $x^{(i)}$:

$$\Sigma = \frac{1}{m}\sum_{i=1}^{n} \underbrace{\underbrace{\left(x^{(i)}\right)}_{1\times m\text{-dim.}} \underbrace{\left(x^{(i)}\right)^T}_{m\times 1\text{-dim.}}}_{m\times m\text{-dim.}}$$

2. The pricipal components are found by computing the eigenvectors and eigenvalues of Σ (solving equation $(\Sigma - \lambda I_m)u = 0$)
3. Choose the k eigenvectors with largest eigenvalues to represent the projection space U

High dimensional data

Dimensionality reduction

1. Compute the covariance matrix from the $x^{(i)}$:

$$\Sigma = \frac{1}{m}\sum_{i=1}^{n} \underbrace{\underbrace{\left(x^{(i)}\right)}_{1\times m\text{-dim.}} \underbrace{\left(x^{(i)}\right)^T}_{m\times 1\text{-dim.}}}_{m\times m\text{-dim.}}$$

2. The pricipal components are found by computing the eigenvectors and eigenvalues of Σ (solving equation $(\Sigma - \lambda I_m)u = 0$)
3. Choose the k eigenvectors with largest eigenvalues to represent the projection space U
4. These k eigenvectors in U are used to transform the inputs x_i to z_i:

$$z^{(i)} = U^T x^{(i)}$$

High dimensional data

How to choose the number k of dimensions?

We can calculate

$$\frac{\text{Average squared projection error}}{\text{Total variation in the data}} \rightarrow \frac{\sum_{i=1}^{m} ||x^{(i)} - x_{\text{approx}}^{(i)}||^2}{\frac{1}{m}\sum_{i=1}^{m} ||x^{(i)}||^2}$$

as the accuracy of the projection using k principle components as a function of the eigenvalues

$$\frac{\sum_{i=1}^{k} \sqrt{u_i}}{\sum_{j=1}^{m} \sqrt{u_j}} = d$$

High dimensional data

How to choose the number k of dimensions?

We can calculate

$$\frac{\text{Average squared projection error}}{\text{Total variation in the data}} \rightarrow \frac{\sum_{i=1}^{m} ||x^{(i)} - x_{\text{approx}}^{(i)}||^2}{\frac{1}{m}\sum_{i=1}^{m} ||x^{(i)}||^2}$$

as the accuracy of the projection using k principle components as a function of the eigenvalues

$$\frac{\sum_{i=1}^{k} \sqrt{u_i}}{\sum_{j=1}^{m} \sqrt{u_j}} = d$$

We say that $100 \cdot (1-d)\%$ of variance is retained.
(Typically, $d \in [0.01, 0.05]$)

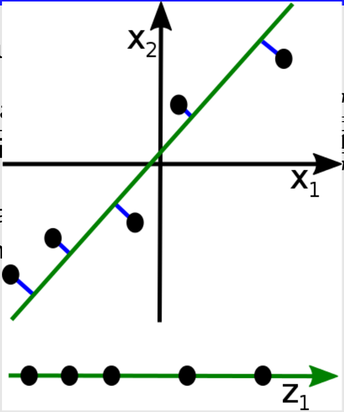
High dimensional data

How to choose the number k of dimensions?

We can calcu

Average squa ... $\sum_{i=1}^{m} ||x^{(i)} - x^{(i)}_{\text{approx}}||^2$

Total vari ... $\frac{1}{m} \sum_{i=1}^{m} ||x^{(i)}||^2$



as the accura ... principle components
as a function

Outline

Latent Semantic Indexing

Motivation

In information retrieval, a common task is to obtain from a large body of documents that subset which best matches a pre-given query

Latent Semantic Indexing

Motivation

In information retrieval, a common task is to obtain from a large body of documents that subset which best matches a pre-given query

$\rightarrow$ Typical feature rpresentations of documents are then term-document matrices:

Latent Semantic Indexing

Motivation

Terms	Documents													
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14
abnormalities	0	0	0	0	0	0	0	1	0	1	0	0	0	0
age	1	0	0	0	0	0	0	0	0	0	0	1	0	0
behavior	0	0	0	0	1	1	0	0	0	0	0	0	0	0
blood	0	0	0	0	0	0	0	1	0	0	1	0	0	0
close	0	0	0	0	0	0	1	0	0	0	1	0	0	0
→ culture	1	1	0	0	0	0	0	1	1	0	0	0	0	0
depressed	1	0	1	1	1	0	0	0	0	0	0	0	0	0
discharge	1	1	0	0	0	1	0	0	0	0	0	0	0	0
disease	0	0	0	0	0	0	0	0	1	0	1	0	0	0
fast	0	0	0	0	0	0	0	0	0	1	0	1	1	1
generation	0	0	0	0	0	0	0	0	1	0	0	0	1	0
oestrogen	0	0	1	1	0	0	0	0	0	0	0	0	0	0
patients	1	1	0	1	0	0	0	1	0	0	0	0	0	0
pressure	0	0	0	0	0	0	0	0	0	0	1	0	0	1
rats	0	0	0	0	0	0	0	0	0	0	0	0	1	1
respect	0	0	0	0	0	0	0	1	0	0	0	1	0	0
rise	0	0	0	1	0	0	0	0	0	0	0	0	0	1
study	1	0	1	0	0	0	0	0	1	0	0	0	0	0

Latent Semantic Indexing

Motivation

In information retrieval, a common task is to obtain from a large body of documents that subset which best matches a pre-given query

→ Typical feature rpresentations of documents are then term-document matrices:

→ These matrices are typically huge but sparse.

Latent Semantic Indexing

Motivation

In information retrieval, a common task is to obtain from a large body of documents that subset which best matches a pre-given query

→ Typical feature rpresentations of documents are then term-document matrices:

→ These matrices are typically huge but sparse.

How to identify those feature dimensions (or combinations thereof) which are most meaningful in such sparse matrices?

Latent Semantic Indexing

Singular Value Decomposition

Any $m \times n$ matrix C can be represented as a singular value decomposition in the form $C = U\Sigma V^T$ where

- U $m \times m$ matrix with orthogonal eigenvectors of CC^T as columns
- V $n \times n$ matrix with orthogonal eigenvectors of C^TC as columns
- Σ Diagonal Matrix with $\Sigma_{ii} = \sqrt{\lambda_i}$; $\Sigma_{ij} = 0, i \neq j$

Latent Semantic Indexing

Singular Value Decomposition

Any $m \times n$ matrix C can be represented as a singular value decomposition in the form $C = U\Sigma V^T$ where

- U $m \times m$ matrix with orthogonal eigenvectors of CC^T as columns
- V $n \times n$ matrix with orthogonal eigenvectors of $C^T C$ as columns
- Σ Diagonal Matrix with $\Sigma_{ii} = \sqrt{\lambda_i}$; $\Sigma_{ij} = 0, i \neq j$

$\rightarrow$ $CC^T = U\Sigma V^T \; V\Sigma U^T = U\Sigma^2 U^T$

- CC^T is a square symmetric real-valued matrix
- Entry (i, j) is a measure of the overlap between the ith and jth terms.
- For term-document incident matrices, it is the number of documents with co-occuring terms i and j.

Latent Semantic Indexing

Singular Value Decomposition

Any $m \times n$ matrix C can be represented as a singular value decomposition in the form $C = U\Sigma V^T$ where

- U $m \times m$ matrix with orthogonal eigenvectors of CC^T as columns
- V $n \times n$ matrix with orthogonal eigenvectors of C^TC as columns
- Σ Diagonal Matrix with $\Sigma_{ii} = \sqrt{\lambda_i}$; $\Sigma_{ij} = 0, i \neq j$

→ $CC^T = U\Sigma V^T \; V\Sigma U^T = U\Sigma^2 U^T$

- CC^T is a square symmetric real-valued matrix
- Entry (i, j) is a measure of the overlap between the ith and jth terms.
- For term-document incident matrices, it is the number of documents with co-occuring terms i and j.

→ Choosing just the first k eigenvectors, the document vectors will be mapped to a lower dimensional representation

It can be shown that this mapping will result in the k-dimensional space with smallest distance to the original space

Latent Semantic Indexing

Example

	d_1	d_2	d_3	d_4	d_5	d_6
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
voyage	1	0	0	1	1	0
trip	0	0	0	1	0	1

U:

Σ:

V^T:

Latent Semantic Indexing

Example

U:

	1	2	3	4	5
ship	−0.44	−0.30	0.57	0.58	0.25
boat	−0.13	−0.33	−0.59	0.00	0.73
ocean	−0.48	−0.51	−0.37	0.00	−0.61
voyage	−0.70	0.35	0.15	−0.58	0.16
trip	−0.26	0.65	−0.41	0.58	−0.09

Σ:

2.16	0.00	0.00	0.00	0.00
0.00	1.59	0.00	0.00	0.00
0.00	0.00	1.28	0.00	0.00
0.00	0.00	0.00	1.00	0.00
0.00	0.00	0.00	0.00	0.39

V^T:

	d_1	d_2	d_3	d_4	d_5	d_6
1	−0.75	−0.28	−0.20	−0.45	−0.33	−0.12
2	−0.29	−0.53	−0.19	0.63	0.22	0.41
3	0.28	−0.75	0.45	−0.20	0.12	−0.33
4	0.00	0.00	0.58	0.00	−0.58	0.58
5	−0.53	0.29	0.63	0.19	0.41	−0.22

Latent Semantic Indexing

Example

Σ:

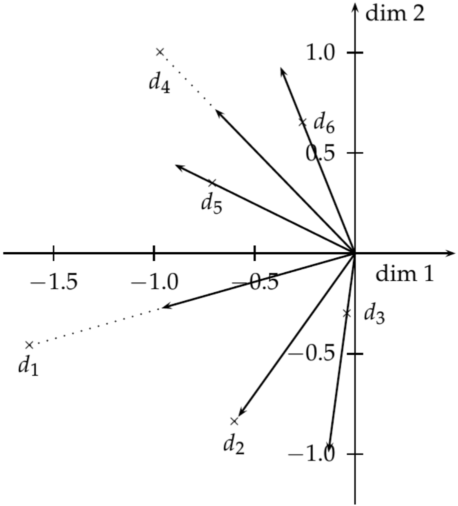
2.16	0.00	0.00	0.00	0.00
0.00	1.59	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00

C_2:

	d_1	d_2	d_3	d_4	d_5	d_6
1	−1.62	−0.60	−0.44	−0.97	−0.70	−0.26
2	−0.46	−0.84	−0.30	1.00	0.35	0.65
3	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00

Find similar queries via the Cosine-similarity

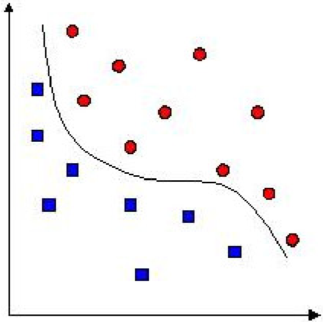
	d_1	d_2	d_3	d_4	d_5	d_6
1	−1.62	−0.60	−0.44	−0.97	−0.70	−0.26
2	−0.46	−0.84	−0.30	1.00	0.35	0.65



Outline

Support vector machines (SVM)

For our previous classifier, we have designed an objective function of sufficient dimension

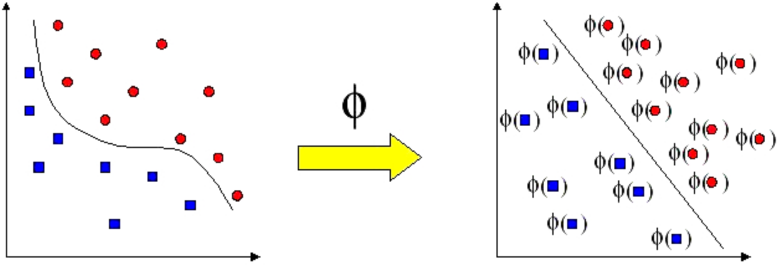


Support vector machines (SVM)

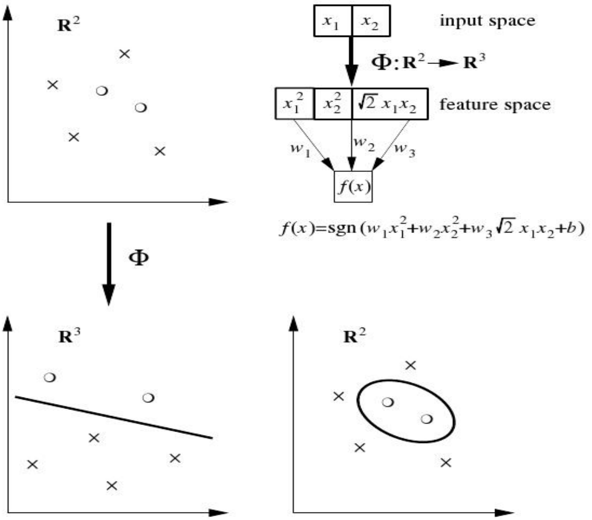
For our previous classifier, we have designed an objective function of sufficient dimension

Alternative to designing complex non-linear functions:

Change dimension of input space so that linear separator is again possible



Support vector machines (SVM)

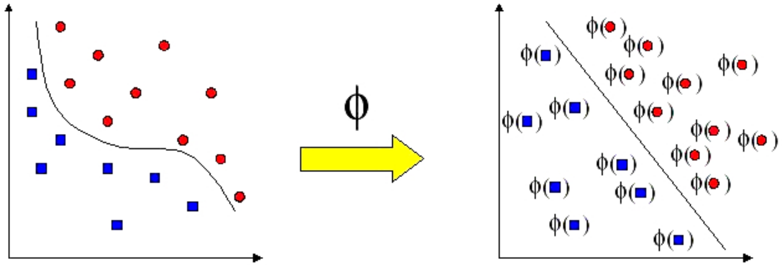


Support vector machines (SVM)

SVM pre-processes data to represent patterns in a high dimension

Dimension often much higher than original feature space

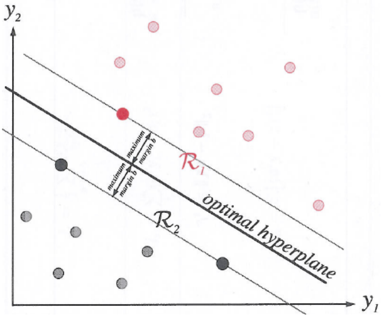
Then, insert hyperplane in order to separate the data



Support vector machines (SVM)

The goal for support vector machines is to find a separating hyperplane with the largest margin to the outer points in all sets

If no such hyperplane exists, map all points into a higher dimensional space until such a plane exists



Support vector machines (SVM)

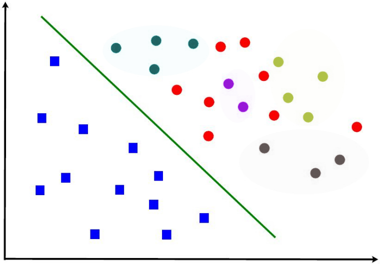
Simple application to several classes by iterative approach:

belongs to class 1 or not?

belongs to class 2 or not?

...

Search for optimum mapping between input space and feature space complicated (no optimum approach known)



Outline

Support vector machines (SVM)

Cost function

Contribution of a single sample to the overall cost:

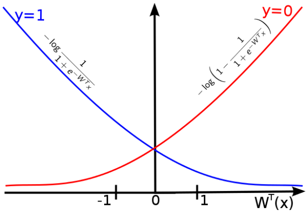
Support vector machines (SVM)

Cost function

Contribution of a single sample to the overall cost:

Logistic regression

$$-y \cdot \log \frac{1}{1+e^{-W^T x}} - (1-y) \cdot \log\left(1 - \frac{1}{1+e^{-W^T x}}\right)$$



Support vector machines (SVM)

Cost function

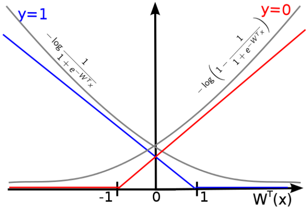
Contribution of a single sample to the overall cost:

Logistic regression

$$-y \cdot \log \frac{1}{1+e^{-W^T x}} - (1-y) \cdot \log \left(1 - \frac{1}{1+e^{-W^T x}}\right)$$

SVM

$$-y \cdot \text{cost}_{y=1}(W^T x) + -(1-y) \cdot \text{cost}_{y=0}(W^T x)$$



Support vector machines (SVM)

Cost function

Logistic regression

$$\min_W \frac{1}{m}\left[\sum_{i=1}^m y_i\left(-\log\frac{1}{1+e^{-W^Tx_i}}\right)+(1-y_i)\left(-\log\left(1-\frac{1}{1+e^{-W^Tx_i}}\right)\right)\right]+\frac{\lambda}{2m}\sum_{j=1}^n w_j^2$$

SVM

$$\min_W C\sum_{i=1}^m\left[y_i\text{cost}_{y=1}(W^Tx_i)+(1-y_i)\text{cost}_{y=0}(W^Tx_i)\right]+\frac{1}{2}\sum_{j=1}^n w_j^2$$

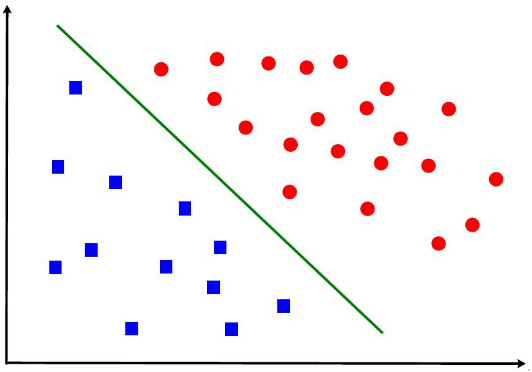
1

[1] C here plays a similar role as $\frac{1}{\lambda}$

Outline

Support vector machines (SVM)

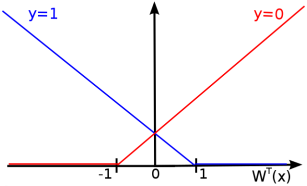
SVM hypothesis

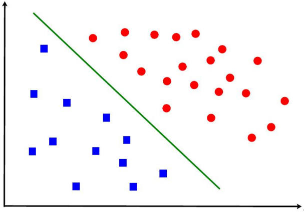


$$\min_{W} C \sum_{i=1}^{m} \left[y_i \text{cost}_{y=1}(W^T x_i) + (1 - y_i) \text{cost}_{y=0}(W^T x_i) \right] + \frac{1}{2} \sum_{j=1}^{n} w_j^2$$

Support vector machines (SVM)

SVM hypothesis

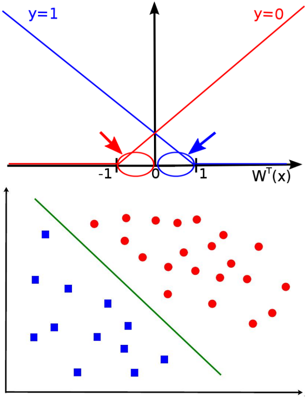




$$\min_{W} C \sum_{i=1}^{m} \left[y_i \mathrm{cost}_{y=1}(W^T x_i) + (1 - y_i)\mathrm{cost}_{y=0}(W^T x_i) \right] + \frac{1}{2} \sum_{j=1}^{n} w_j^2$$

Support vector machines (SVM)

SVM hypothesis

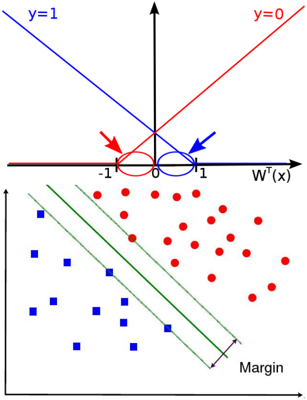


$$W^T x \begin{cases} \geq 0 \\ < 0 \end{cases} \text{ sufficient}$$

$$\min_W C \sum_{i=1}^{m} \left[y_i \text{cost}_{y=1}(W^T x_i) + (1 - y_i) \text{cost}_{y=0}(W^T x_i) \right] + \frac{1}{2} \sum_{j=1}^{n} w_j^2$$

Support vector machines (SVM)

SVM hypothesis

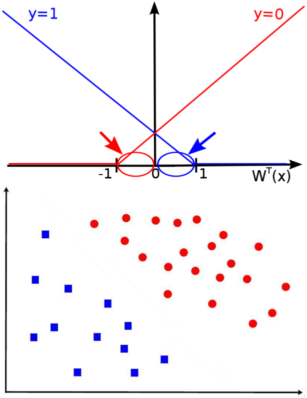


$$W^T x \begin{cases} \geq 1 \\ \leq -1 \end{cases} \Rightarrow \text{confidence}$$

$$\min_W C \sum_{i=1}^{m} \left[y_i \text{cost}_{y=1}(W^T x_i) + (1 - y_i)\text{cost}_{y=0}(W^T x_i) \right] + \frac{1}{2} \sum_{j=1}^{n} w_j^2$$

Support vector machines (SVM)

SVM hypothesis



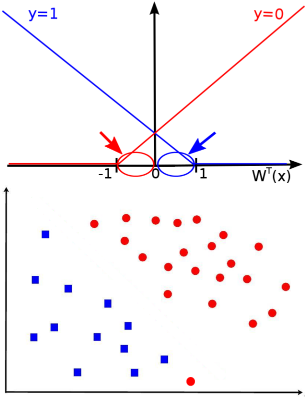
$$W^T x \begin{cases} \geq 1 \\ \leq -1 \end{cases} \Rightarrow \text{confidence}$$

Outliers: Elastic decision boundary

$$\min_W C \sum_{i=1}^{m} \left[y_i \text{cost}_{y=1}(W^T x_i) + (1 - y_i)\text{cost}_{y=0}(W^T x_i) \right] + \frac{1}{2} \sum_{j=1}^{n} w_j^2$$

Support vector machines (SVM)

SVM hypothesis



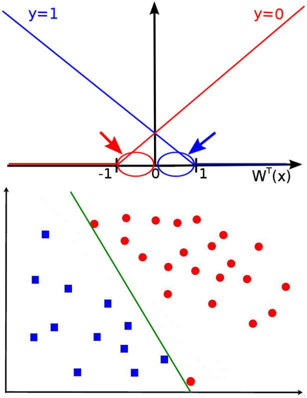
$$W^T x \begin{cases} \geq 1 \\ \leq -1 \end{cases} \Rightarrow \text{confidence}$$

Outliers: Elastic decision boundary

$$\min_W C \sum_{i=1}^{m} \left[y_i \text{cost}_{y=1}(W^T x_i) + (1 - y_i)\text{cost}_{y=0}(W^T x_i) \right] + \frac{1}{2} \sum_{j=1}^{n} w_j^2$$

Support vector machines (SVM)

SVM hypothesis



$$W^T x \begin{cases} \geq & 1 \\ \leq & -1 \end{cases} \Rightarrow \text{confidence}$$

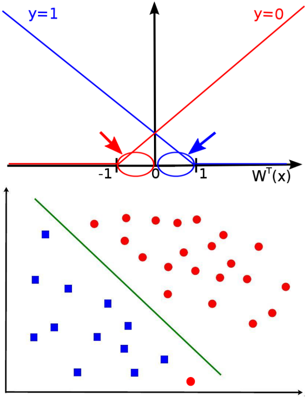
Outliers: Elastic decision boundary

large C stricter boundary at the cost of smaller margin

$$\min_W C \sum_{i=1}^{m} \left[y_i \text{cost}_{y=1}(W^T x_i) + (1 - y_i)\text{cost}_{y=0}(W^T x_i) \right] + \frac{1}{2} \sum_{j=1}^{n} w_j^2$$

Support vector machines (SVM)

SVM hypothesis



$$W^T x \begin{cases} \geq 1 \\ \leq -1 \end{cases} \Rightarrow \text{confidence}$$

Outliers: Elastic decision boundary

small C tolerates outliers

$$\min_W C \sum_{i=1}^{m} \left[y_i \text{cost}_{y=1}(W^T x_i) + (1 - y_i)\text{cost}_{y=0}(W^T x_i) \right] + \frac{1}{2} \sum_{j=1}^{n} w_j^2$$

Support vector machines (SVM)

Large margin classifier

$$\min_{W} C \sum_{i=1}^{m} \left[y_i \text{cost}_{y=1}(W^T x_i) + (1 - y_i) \text{cost}_{y=0}(W^T x_i) \right] + \frac{1}{2} \sum_{j=1}^{n} w_j^2$$

Support vector machines (SVM)

Large margin classifier

$$\min_{W} C \sum_{i=1}^{m} \left[y_i \text{cost}_{y=1}(W^T x_i) + (1 - y_i)\text{cost}_{y=0}(W^T x_i) \right] + \frac{1}{2} \sum_{j=1}^{n} w_j^2$$

Rewrite the SVM optimisation problem as

$$\begin{aligned} \min_{W} \quad & \tfrac{1}{2} \textstyle\sum_{j=1}^{n} w_j^2 \\ s.t. \quad & W^T x_i \geq 1 \quad \text{if } y_i = 1 \\ & W^T x_i \leq -1 \quad \text{if } y_i = 0 \end{aligned}$$

Support vector machines (SVM)

Large margin classifier

$$\min_{W} \quad \frac{1}{2}\sum_{j=1}^{n} w_j^2$$

$$s.t. \quad W^T x_i \geq 1 \text{ if } y_i = 1$$

$$W^T x_i \leq -1 \text{ if } y_i = 0$$

Support vector machines (SVM)

Large margin classifier

$$\min_{W} \quad \frac{1}{2}\sum_{j=1}^{n} w_j^2 = \frac{1}{2}\left(\sqrt{w_1^2 + \cdots + w_n^2}\right)^2$$

$$s.t. \quad W^T x_i \geq 1 \text{ if } y_i = 1$$

$$W^T x_i \leq -1 \text{ if } y_i = 0$$

Support vector machines (SVM)

Large margin classifier

$$\min_{W} \quad \frac{1}{2}\sum_{j=1}^{n} w_j^2 = \frac{1}{2}\left(\sqrt{w_1^2 + \cdots + w_n^2}\right)^2 \quad = \frac{1}{2}||W||^2$$

$$s.t. \qquad W^T x_i \geq 1 \;\; \text{if } y_i = 1$$

$$W^T x_i \leq -1 \text{ if } y_i = 0$$

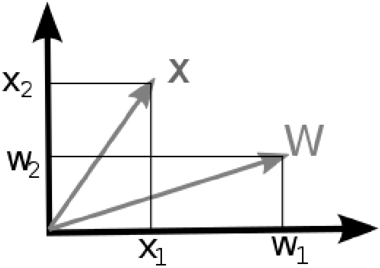
Support vector machines (SVM)

Large margin classifier

$$\min_{W} \quad \frac{1}{2}\sum_{j=1}^{n} w_j^2 = \frac{1}{2}\left(\sqrt{w_1^2 + \cdots + w_n^2}\right)^2 \quad = \frac{1}{2}||W||^2$$

$$s.t. \quad W^T x_i \geq 1 \text{ if } y_i = 1$$

$$W^T x_i \leq -1 \text{ if } y_i = 0$$



$$W^T x = w_1 x_1 + w_2 x_2$$

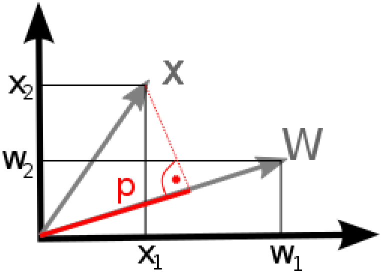
Support vector machines (SVM)

Large margin classifier

$$\min_{W} \quad \frac{1}{2}\sum_{j=1}^{n} w_j^2 = \frac{1}{2}\left(\sqrt{w_1^2 + \cdots + w_n^2}\right)^2 \quad = \frac{1}{2}||W||^2$$

$$s.t. \qquad W^T x_i \geq 1 \;\text{ if } y_i = 1$$

$$W^T x_i \leq -1 \text{ if } y_i = 0$$



$$W^T x = w_1 x_1 + w_2 x_2 = ||W|| \cdot p$$

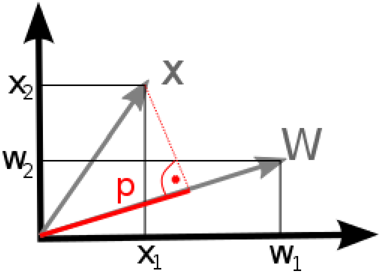
Support vector machines (SVM)

Large margin classifier

$$\min_{W} \quad \frac{1}{2}\sum_{j=1}^{n} w_j^2 = \frac{1}{2}\left(\sqrt{w_1^2 + \cdots + w_n^2}\right)^2 \quad = \frac{1}{2}||W||^2$$

$$s.t. \quad W^T x_i \geq 1 \text{ if } y_i = 1 \quad \rightarrow ||W|| \cdot p_i \geq 1$$

$$W^T x_i \leq -1 \text{ if } y_i = 0 \quad \rightarrow ||W|| \cdot p_i \leq -1$$



$$W^T x = w_1 x_1 + w_2 x_2 = ||W|| \cdot p$$

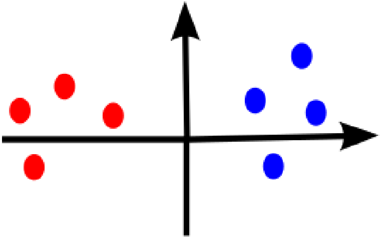
Support vector machines (SVM)

Large margin classifier

$$\min_{W} \quad \frac{1}{2}\sum_{j=1}^{n} w_j^2 = \frac{1}{2}\left(\sqrt{w_1^2+\cdots+w_n^2}\right)^2 \quad = \frac{1}{2}||W||^2$$

$$s.t. \qquad W^T x_i \geq 1 \ \text{ if } y_i = 1 \qquad \rightarrow ||W|| \cdot p_i \geq 1$$

$$W^T x_i \leq -1 \text{ if } y_i = 0 \qquad \rightarrow ||W|| \cdot p_i \leq -1$$



Which decision boundaray is found?

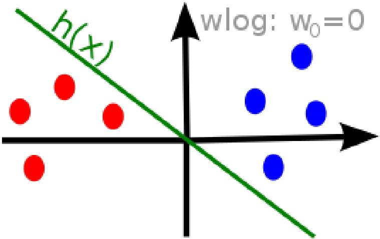
Support vector machines (SVM)

Large margin classifier

$$\min_{W} \quad \frac{1}{2}\sum_{j=1}^{n} w_j^2 = \frac{1}{2}\left(\sqrt{w_1^2+\cdots+w_n^2}\right)^2 \quad = \frac{1}{2}||W||^2$$

$$s.t. \quad W^T x_i \geq 1 \text{ if } y_i = 1 \quad \rightarrow ||W|| \cdot p_i \geq 1$$

$$W^T x_i \leq -1 \text{ if } y_i = 0 \quad \rightarrow ||W|| \cdot p_i \leq -1$$



Which decision boundaray is found?

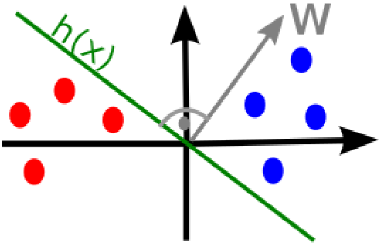
Support vector machines (SVM)

Large margin classifier

$$\min_{W} \quad \frac{1}{2}\sum_{j=1}^{n} w_j^2 = \frac{1}{2}\left(\sqrt{w_1^2 + \cdots + w_n^2}\right)^2 \quad = \frac{1}{2}||W||^2$$

$$s.t. \quad W^T x_i \geq 1 \text{ if } y_i = 1 \quad \rightarrow ||W|| \cdot p_i \geq 1$$

$$W^T x_i \leq -1 \text{ if } y_i = 0 \quad \rightarrow ||W|| \cdot p_i \leq -1$$



Which decision boundaray is found?

$$h(x) = w_1x_1 + w_2x_2$$

$\rightarrow$ W orthogonal to all x with $h(x) = 0$

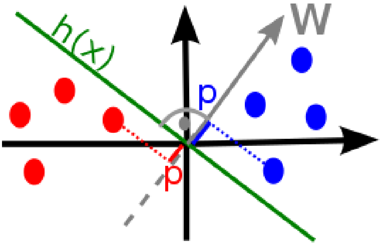
Support vector machines (SVM)

Large margin classifier

$$\min_W \quad \frac{1}{2}\sum_{j=1}^{n} w_j^2 = \frac{1}{2}\left(\sqrt{w_1^2+\cdots+w_n^2}\right)^2 \quad = \frac{1}{2}||W||^2$$

$$s.t. \quad W^T x_i \geq 1 \text{ if } y_i = 1 \quad \rightarrow ||W|| \cdot p_i \geq 1$$

$$W^T x_i \leq -1 \text{ if } y_i = 0 \quad \rightarrow ||W|| \cdot p_i \leq -1$$



Which decision boundaray is found?

$h(x) = w_1 x_1 + w_2 x_2$

$\rightarrow$ W orthogonal to all x with $h(x) = 0$

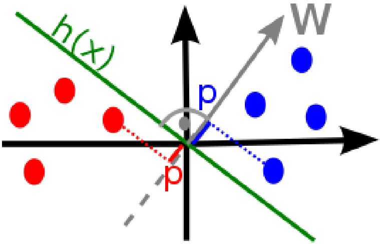
Support vector machines (SVM)

Large margin classifier

$$\min_{W} \quad \frac{1}{2}\sum_{j=1}^{n} w_j^2 = \frac{1}{2}\left(\sqrt{w_1^2 + \cdots + w_n^2}\right)^2 \quad = \frac{1}{2}||W||^2$$

$$s.t. \quad W^T x_i \geq 1 \text{ if } y_i = 1 \quad \rightarrow ||W|| \cdot p_i \geq 1$$

$$W^T x_i \leq -1 \text{ if } y_i = 0 \quad \rightarrow ||W|| \cdot p_i \leq -1$$



Which decision boundaray is found?

$$h(x) = w_1 x_1 + w_2 x_2$$

$\rightarrow$ W orthogonal to all x with $h(x) = 0$

$\Rightarrow$ $\min \frac{1}{2}||W||^2$ and $||W|| \cdot p_i \geq 1$ necessitate larger p_i

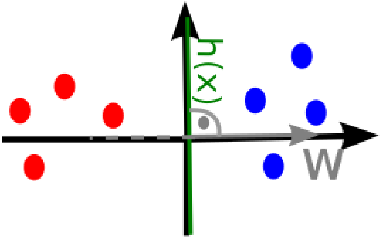
Support vector machines (SVM)

Large margin classifier

$$\min_{W} \quad \frac{1}{2}\sum_{j=1}^{n} w_j^2 = \frac{1}{2}\left(\sqrt{w_1^2 + \cdots + w_n^2}\right)^2 \quad = \frac{1}{2}||W||^2$$

$$s.t. \quad W^T x_i \geq 1 \text{ if } y_i = 1 \quad \rightarrow ||W|| \cdot p_i \geq 1$$

$$W^T x_i \leq -1 \text{ if } y_i = 0 \quad \rightarrow ||W|| \cdot p_i \leq -1$$



Which decision boundaray is found?

$$h(x) = w_1 x_1 + w_2 x_2$$

$\rightarrow$ W orthogonal to all x with $h(x) = 0$

$\Rightarrow$ min $\frac{1}{2}||W||^2$ and $||W|| \cdot p_i \geq 1$ necessitate larger p_i

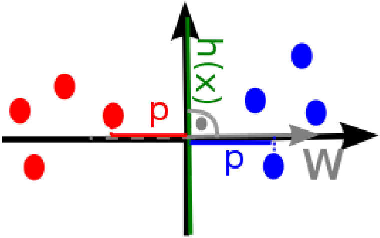
Support vector machines (SVM)

Large margin classifier

$$\min_{W} \quad \frac{1}{2}\sum_{j=1}^{n} w_j^2 = \frac{1}{2}\left(\sqrt{w_1^2 + \cdots + w_n^2}\right)^2 = \frac{1}{2}||W||^2$$

$$s.t. \quad W^T x_i \geq 1 \text{ if } y_i = 1 \quad \rightarrow ||W|| \cdot p_i \geq 1$$

$$W^T x_i \leq -1 \text{ if } y_i = 0 \quad \rightarrow ||W|| \cdot p_i \leq -1$$



Which decision boundaray is found?

$$h(x) = w_1 x_1 + w_2 x_2$$

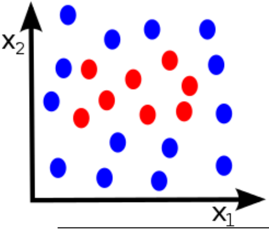
$\rightarrow$ W orthogonal to all x with $h(x) = 0$

$\Rightarrow$ min $\frac{1}{2}||W||^2$ and $||W|| \cdot p_i \geq 1$ necessitate larger p_i

Outline

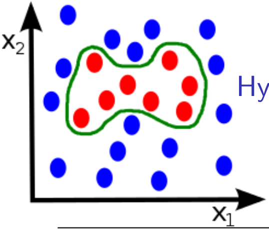
Support vector machines (SVM)

Kernels – Non linear decision boundary



Support vector machines (SVM)

Kernels – Non linear decision boundary

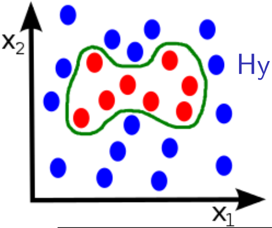


Hypothesis $= 1$ if

$$w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1 x_2 + w_4 x_1^2 + \cdots \geq 0$$

Support vector machines (SVM)

Kernels – Non linear decision boundary



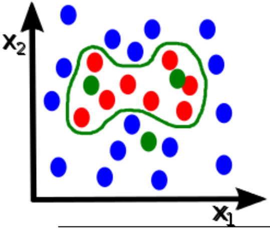
Hypothesis $= 1$ if

$$w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1 x_2 + w_4 x_1^2 + \cdots \geq 0$$

$$\Rightarrow w_0 + w_1 k_1 + w_2 k_2 + w_3 k_3 + \ldots$$

Support vector machines (SVM)

Kernels – Non linear decision boundary

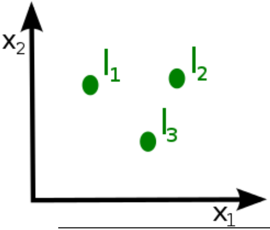


$\Rightarrow w_0 + w_1 k_1 + w_2 k_2 + w_3 k_3 + \ldots$

Kernel Define kernel via landmarks

Support vector machines (SVM)

Kernels – Non linear decision boundary

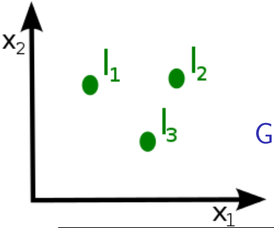


$\Rightarrow \; w_0 + w_1 k_1 + w_2 k_2 + w_3 k_3 + \ldots$

Kernel Define kernel via landmarks

Support vector machines (SVM)

Kernels – Non linear decision boundary

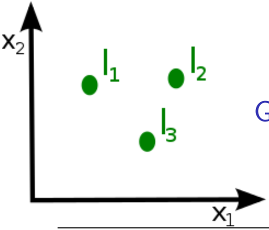


$$\Rightarrow w_0 + w_1 k_1 + w_2 k_2 + w_3 k_3 + \ldots$$

Gaussian: $k(x, l_i) = e^{-\frac{||x - l_i||^2}{2\sigma^2}}$

Support vector machines (SVM)

Kernels – Non linear decision boundary



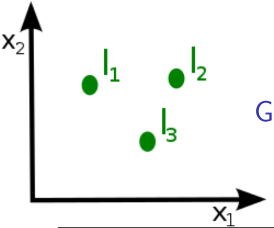
$$\Rightarrow w_0 + w_1 k_1 + w_2 k_2 + w_3 k_3 + \ldots$$

Gaussian: $k(x, l_i) = e^{-\frac{||x - l_i||^2}{2\sigma^2}}$

$x \approx l_i \Rightarrow k(x, l_i) \approx 1$ (0 else)

Support vector machines (SVM)

Kernels – Non linear decision boundary



$$\Rightarrow \quad w_0 + w_1 k_1 + w_2 k_2 + w_3 k_3 + \dots$$

Gaussian: $k(x, l_i) = e^{-\frac{||x-l_i||^2}{2\sigma^2}}$

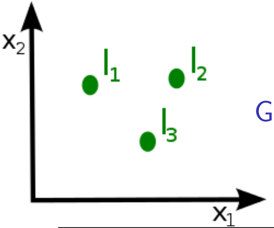
$x \approx l_i \Rightarrow k(x, l_i) \approx 1$ (0 else)

Example: $w_0 = -0.5, w_1 = 1, w_2 = 1, w_3 = 0$

$$h(x) = w_0 + w_1 k(x, l_1) + w_2 k(x, l_2) + w_3 k(x, l_3)$$

Support vector machines (SVM)

Kernels – Non linear decision boundary



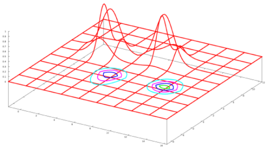
$$\Rightarrow \; w_0 + w_1 k_1 + w_2 k_2 + w_3 k_3 + \ldots$$

Gaussian: $k(x, l_i) = e^{-\frac{||x - l_i||^2}{2\sigma^2}}$

$x \approx l_i \;\Rightarrow k(x, l_i) \approx 1$ (0 else)

Example: $w_0 = -0.5, w_1 = 1, w_2 = 1, w_3 = 0$

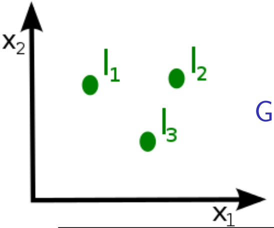
$$h(x) = w_0 + w_1 k(x, l_1) + w_2 k(x, l_2) + w_3 k(x, l_3)$$



$\sigma = 1$

Support vector machines (SVM)

Kernels – Non linear decision boundary



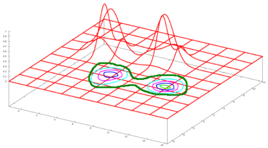
$$\Rightarrow \; w_0 + w_1 k_1 + w_2 k_2 + w_3 k_3 + \ldots$$

Gaussian: $k(x, l_i) = e^{-\frac{||x-l_i||^2}{2\sigma^2}}$

$x \approx l_i \Rightarrow k(x, l_i) \approx 1$ (0 else)

Example: $w_0 = -0.5, w_1 = 1, w_2 = 1, w_3 = 0$

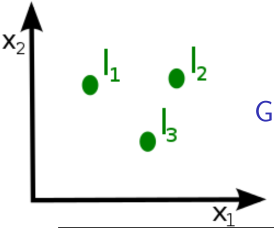
$$h(x) = w_0 + w_1 k(x, l_1) + w_2 k(x, l_2) + w_3 k(x, l_3)$$



$\sigma = 1$

Support vector machines (SVM)

Kernels – Non linear decision boundary



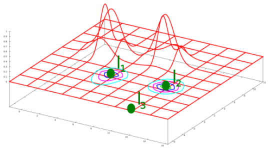
$$\Rightarrow \; w_0 + w_1 k_1 + w_2 k_2 + w_3 k_3 + \dots$$

Gaussian: $k(x, l_i) = e^{-\frac{||x-l_i||^2}{2\sigma^2}}$

$x \approx l_i \Rightarrow k(x, l_i) \approx 1$ (0 else)

Example: $w_0 = -0.5, w_1 = 1, w_2 = 1, w_3 = 0$

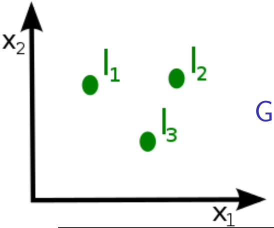
$$h(x) = w_0 + w_1 k(x, l_1) + w_2 k(x, l_2) + w_3 k(x, l_3)$$



$\sigma = 1$

Support vector machines (SVM)

Kernels – Non linear decision boundary



$$\Rightarrow \; w_0 + w_1 k_1 + w_2 k_2 + w_3 k_3 + \ldots$$

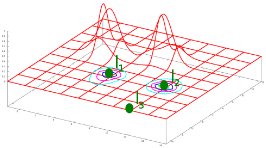
Gaussian: $k(x, l_i) = e^{-\frac{||x - l_i||^2}{2\sigma^2}}$

$x \approx l_i \Rightarrow k(x, l_i) \approx 1$ (0 else)

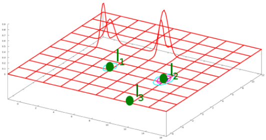
σ controls the width of the Gaussian

Example: $w_0 = -0.5, w_1 = 1, w_2 = 1, w_3 = 0$

$$h(x) = w_0 + w_1 k(x, l_1) + w_2 k(x, l_2) + w_3 k(x, l_3)$$



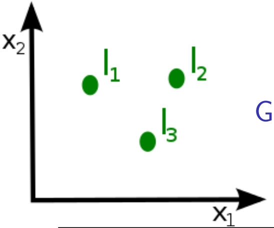
$\sigma = 1$



$\sigma = 0.5$

Support vector machines (SVM)

Kernels – Non linear decision boundary



$$\Rightarrow \; w_0 + w_1 k_1 + w_2 k_2 + w_3 k_3 + \ldots$$

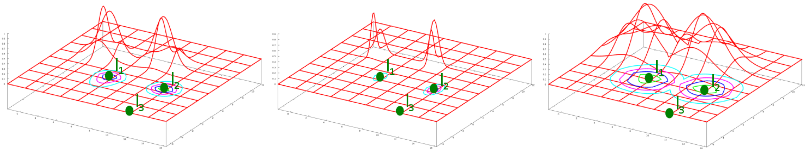
Gaussian: $k(x, l_i) = e^{-\frac{||x-l_i||^2}{2\sigma^2}}$

$x \approx l_i \Rightarrow k(x, l_i) \approx 1$ (0 else)

σ controls the width of the Gaussian

Example: $w_0 = -0.5, w_1 = 1, w_2 = 1, w_3 = 0$

$$h(x) = w_0 + w_1 k(x, l_1) + w_2 k(x, l_2) + w_3 k(x, l_3)$$



$\sigma = 1$ $\quad$ $\sigma = 0.5$ $\quad$ $\sigma = 2$

Support vector machines (SVM)

Kernels – placement of landmarks

Possible choice of initial landmarks: All training-set samples

Training of w_i

$$f_i = \begin{bmatrix} k(x_i, l_1) \\ \vdots \\ k(x_i, l_m) \end{bmatrix}$$

$$\min_W C \sum_{i=1}^{m} y_i \text{cost}_{y_i=1}(W^T f_i) + (1 - y_i) \cdot \text{cost}_{y_i=0}(W^T f_i) + \frac{1}{2} \sum_{j=1}^{m} w_j^2$$

Outline

Questions?

Stephan Sigg
stephan.sigg@cs.uni-goettingen.de

Literature

- C.M. Bishop: Pattern recognition and machine learning, Springer, 2007.
- R.O. Duda, P.E. Hart, D.G. Stork: Pattern Classification, Wiley, 2001.