Support vector machines (SVMs)

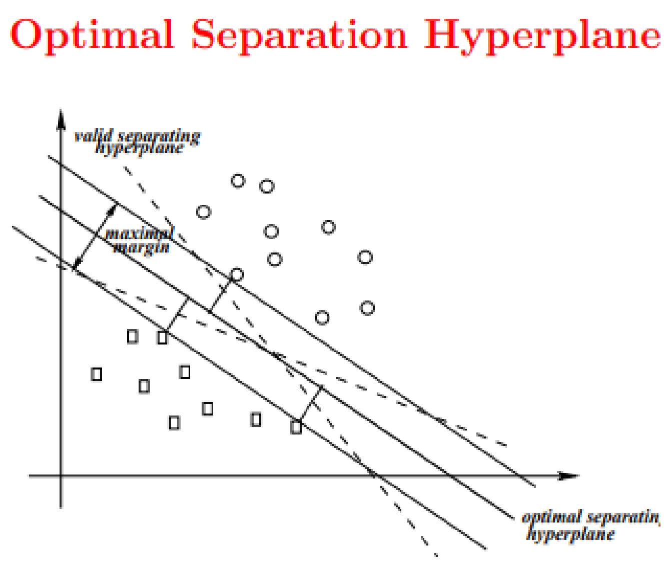
SUMMARY

Performance of supervised learning models

- **predictive performance**
  - performance metrics (AUC,….,MAPE,)
- **computational complexity**
  - for the **training**, and **inference/testing** stages
    - time, space
      - *asymptotic analysis* (classes of complexity: O, θ,…)
      - *empirical analysis*
        - exact running time/required memory on specific data sets

1. Support Vector Machines

- SVMs are a set of related supervised learning methods used for classification and regression
- SVMs are eager inductive learners
- instances are high dimensional real valued vectors (data points in $\Re^d$)
- a SVM constructs a hyperplane or a set of hyperplanes in a high dimensional space, which can be used for classification, regression or other tasks.
  - a good separation is achieved by the hyperplane that has the largest distance to the nearest training data points of any class (the so called *functional margin*)
  - in general, the larger the margin, the lower the generalization error of the classifier ⇒ ***large margin classifiers***



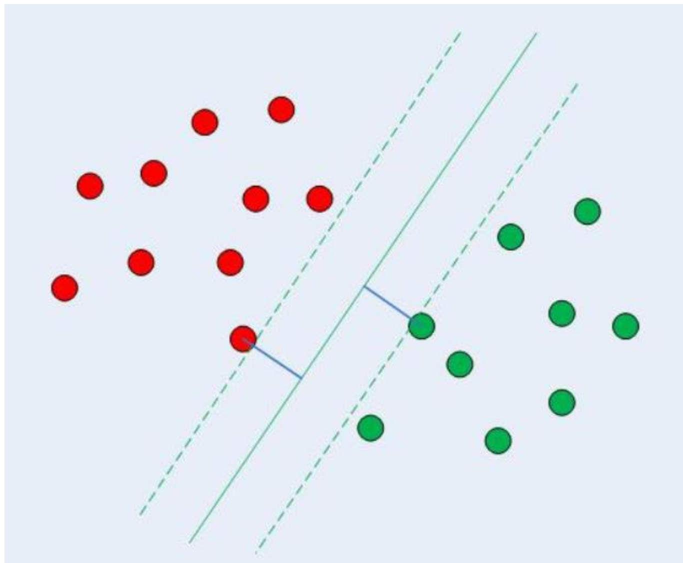
[6]

- SVMs are inherently two-class classifiers
  - for multiple classes (M), two approaches have been proposed
    - **one-against-the-rest**
      - construct a hyperplane between class *k* and the *M*-1 other classes ⇒ *M* SVMs
    - **one-against-one**
      - construct a hyperplane for any two classes ⇒ *M* *(*M*-1)/2 SVMs
- **Extensions of the classical architecture**
  - **Deep SVMs**
    - Different perspectives
      - multiple layers of SVMs - DSVM
      - deep kernel learning architecture with multiple intermediate layers – Totally Deep SVMs
      - DSVMs for regression
- **Applications**
  - pattern recognition
  - classification (facial expression classification, image classification)
    - text categorization
      - e-mail filtering
      - web searching
      - sorting documents by topic
  - predictions (of traffic speed, protein structure prediction, breast cancer diagnosis prediction, time series prediction)
  - object detection, intrusion detection
  - handwritten recognition
  - …
- **Main idea** of SVMs
  - Given a set of data points which belong to either of two classes
    - Find an optimal separating hyperplane

- leaving the largest possible fraction of points of the same class on the same side

and

- maximizing the distance of either class from the hyperplane



- Minimize the risk of misclassifying the training samples and the unseen test samples

- **<u>Approach</u>**: Formulate a ***constrained optimisation problem***, then solve it using ***constrained [quadratic programming](#)***
- [SMO](#) (Sequential Minimal Optimisation) algorithm, developed by John Platt, in 1986

- **Characteristics**
  - SVMs implement automatic complexity control to avoid overfitting
  - even if an SVM has a lot of hyperparameters, large margins make them simple classifiers
  - intuition regarding the large margin:
    - points near the decision surface may represent very uncertain classification decisions
      - there is almost 50% chance of the classifier going either way
    - a classifier with large margin makes no very uncertain decisions
  - [SVR](#) (Support Vector Regression)

2. Types of SVMs

- Linear SVMs
- Linear SVMs with soft margin
- Nonlinear SVMs

a). Linear SVMs

- the linear case
- data are linearly separable by a hyperplane

- linear classifier
- formulate an optimisation problem
  - **Formalisation**
    - ***S*** a set o data points $x_i \in \Re^d$, $i=1,2\ldots,m$
    - two classes: +, -
      - the label of each instance x_i is $y_i \in \{-1,+1\}$
    - **training data** $<x_i, y_i>$, $i=1,2\ldots,m$
    - impose a **functional margin** at least 1

The set S is linear separable if there are $w \in R^d$ and $w_0 \in R$ such that

$$y_i(w \cdot x_i + w_0) \geq 1 \qquad i = 1, \ldots, m$$

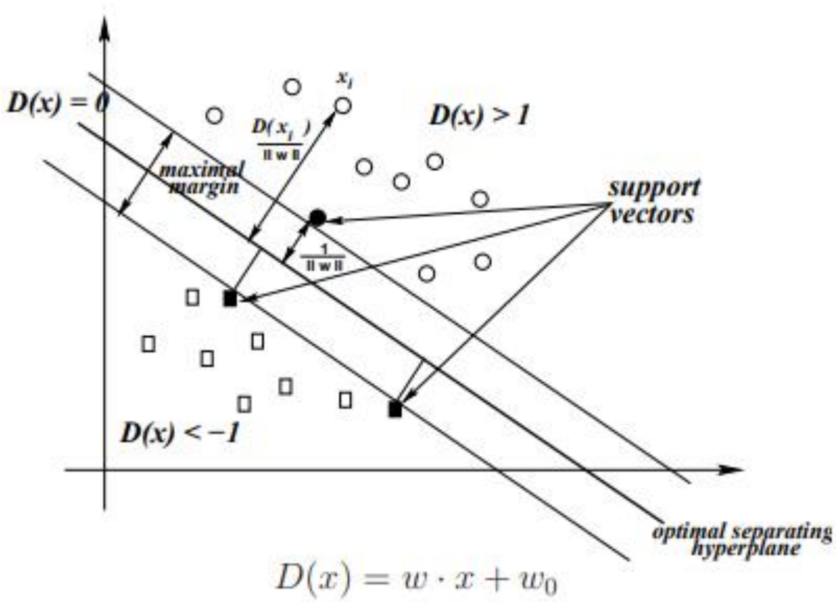
The pair (w, w_0) defines the hyperplane of equation $w \cdot x + w_0 = 0$, named the separating hyperplane.

[6]

The signed distance d_i of a point x_i to the separating hyperplane (w, w_0) is given by $d_i = \frac{w \cdot x_i + w_0}{||w||}$.

It follows that $y_i d_i \geq \frac{1}{||w||}$, therefore $\frac{1}{||w||}$ is the lower bound on the distance between points x_i and the separating hyperplane (w, w_0).

[6]



[6]

- **linear classifier**
  - $f(x) = sgn(w \cdot x + w_0)$
    - if $w \cdot x + w_0 > 0 \Rightarrow$ predict 1, otherwise predict -1

- **Linear SVMs: the primal form**

$$\begin{aligned}&\textit{minimize} \quad \tfrac{1}{2}||w||^2\\ &\textit{subject to} \quad y_i(w \cdot x_i + w_0) \geq 1 \text{ for } i = 1, \ldots, m\end{aligned} \qquad \textbf{(1)}$$

  - **constrained quadratic problem** (QP) with d+1 parameters
  - hypothesis $\rightarrow$ hyperplane
    - $(w_0, w_1, \ldots w_d) \in \Re^{d+1}$
  - if *d* is not very big (10^3) can be solved using **quadratic optimisation methods**
  - for large values of *d* (10^5)
    - Kuhn-Tucker theorem (nonlinear programming)
      - objective function and the associated constraints are convex

        $\Rightarrow$ the *Lagrange multipliers* $(\alpha_i \geq 0, i = 1, \ldots, m)$ are used to transform the problem into an equivalent **dual form**

- **Linear SVMs: the dual form**

$$\begin{aligned}&\textit{maximize} \quad \textstyle\sum_{i=1}^m \alpha_i - \frac{1}{2}\sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \, x_i \cdot x_j\\ &\textit{subject to} \quad \textstyle\sum_{i=1}^m y_i \alpha_i = 0\\ &\qquad\qquad\quad \alpha_i \geq 0, i = 1, \ldots, m\end{aligned}$$

The link between the primal and the dual form:

The optimal solution $(\overline{w}, \overline{w}_0)$ of the primal QP problem is given by

$$\overline{w} = \sum_{i=1}^m \overline{\alpha}_i y_i x_i$$

$$\overline{\alpha}_i (y_i(\overline{w} \cdot x_i + \overline{w}_0) - 1) = 0 \text{ for any } i = 1, \ldots, m$$

where $\overline{\alpha}_i$ are the optimal solutions of the above (dual form) optimisation problem. [6]

b). Linear SVMs with soft margin

- the data set ***S*** is not linearly separable, or one ignores weather or not ***S*** is linearly separable
- extend the optimisation problem **(1)**, by allowing a small number of misclassified points
  - better generalisation of computational efficiency
  - *m* non-negative variables ξ_i

- **primal form**

$$\begin{aligned}&\textit{minimize} \quad \tfrac{1}{2}||w||^2 + C\textstyle\sum_{i=1}^m \xi_i\\ &\textit{subject to} \quad y_i(w \cdot x_i + w_0) \geq 1 - \xi_i \text{ for } i = 1, \ldots, m\\ &\qquad\qquad\quad \xi_i \geq 0 \text{ for } i = 1, \ldots, m\end{aligned}$$

- **dual form**

$$\begin{aligned} &\textit{maximize} && \textstyle\sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i\alpha_j y_i y_j \, x_i \cdot x_j \\ &\textit{subject to} && \textstyle\sum_{i=1}^{m} y_i\alpha_i = 0 \\ & && 0 \le \alpha_i \le C, i = 1, \ldots, m \end{aligned}$$

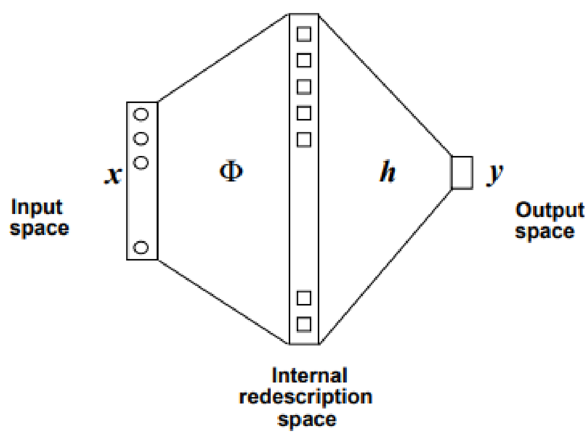
[6]

- introduce a parameter ***C*** – ***misclassification parameter***
  - acts as a regularizing parameter
    - reducing overfitting
  - large ***C*** $\Rightarrow$ minimize the number of misclassified points
  - small ***C*** $\Rightarrow$ maximize the functional margin $\frac{1}{\|w\|}$

c). Nonlinear SVMs

- the data points from the input space $\Re^d$ are mapped into a higher dimensional space $\Re^n$ (n>d) using a function called a map $\phi : \Re^d \rightarrow \Re^n$
  - in a higher dimensional space, it is likely that a linear separator can be constructed
- the training algorithm would depend on the scalar (dot) product $\phi(x_i) \cdot \phi(x_j)$
- constructing (via ϕ) a separating hyperplane with maximum margin in the higher dimensional space yields a nonlinear decision boundary in the input space

General Schema for Nonlinear SVMs



[6]

- nonlinear SVMs make use of the "**kernel trick**"
  - the dot product is computationally expensive
  - a kernel function K: $\Re^d \times \Re^d \rightarrow \Re$ such that $K(x_1, x_2) = \phi(x_1) \cdot \phi(x_2)$, where $\cdot$ is the dot product
  - by using the kernel function, it is possible to compute the separating hyperplane without explicitly constructing the map ϕ

3. Some kernel functions for SVMs

- **Polynomial:** $K(x, x') = (x \cdot x' + c)^q$
- **RBF (radial basis function):** $K(x, x') = e^{-\frac{||x-x'||^2}{2\sigma^2}}$
- **Sigmoide:** $K(x, x') = tanh(\alpha x \cdot x' - b)$

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Example Considering that the dimensionality of the input space is 2 (d=2), show that the polynomial function K previously defined, for c=0 and q=2 is a kernel function.

Proof. Assuming that $x=(x_1, x_2)$ and $y=(y_1, y_2)$, we have that

$K(x, y)=(x \cdot y)^2= x_1^2 y_1^2+2\, x_1 y_1\, x_2\, y_2+ x_2^2 y_2^2$

For proving that K is a kernel function, we must prove that there exists $\phi: \Re^2 \rightarrow \Re^n$ (n>2) such that $(x \cdot y)^2 = \phi(x) \cdot \phi(y)$.

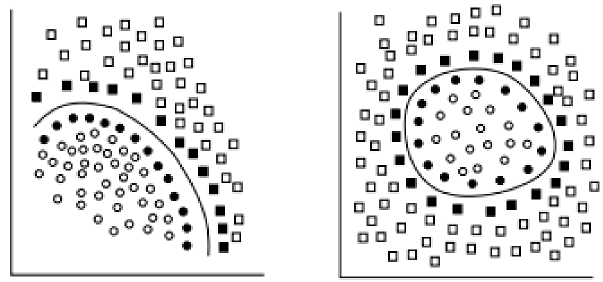
There are three possible mapping functions ϕ which satisfy the previous equality.

a). $\phi: \Re^2 \rightarrow \Re^3$, $\phi(x_1, x_2)=(x_1^2, \sqrt{2} x_1 x_2, x_2^2)$

b). $\phi: \Re^2 \rightarrow \Re^3$, $\phi(x_1, x_2)=1/\sqrt{2}\,(x_1^2 - x_2^2, 2 x_1 x_2, x_1^2+x_2^2)$

c). $\phi: \Re^2 \rightarrow \Re^4$, $\phi(x_1, x_2)=(x_1^2, x_1 x_2, x_1 x_2, x_2^2)$

- decision surfaces induced by a kernel function



(a) *(b)*

Decision surface

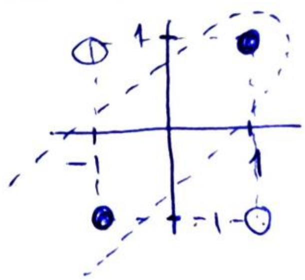
- (a) **by a polynomial classifier, and**
- (b) **by a RBF.**

Support vectors are indicated in dark fill.

[6]

XOR Example

SVM for the XOR problem



A	B	Output	
1	1	1	Class 1
-1	1	-1	Class 2
1	-1	-1	Class 2
-1	-1	1	Class 1

Polynomial Kernel

$$K(u,v) = (u \odot v + 1)^2 = (u_1 \cdot v_1 + u_2 \cdot v_2 + 1)^2 =$$

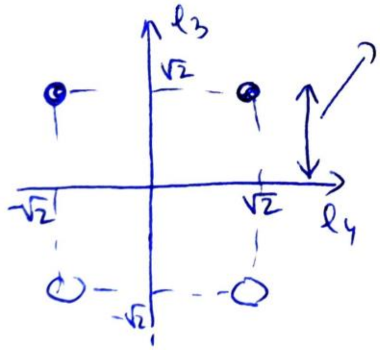
$$u_1^2 v_1^2 + u_2^2 v_2^2 + 2 u_1 u_2 v_1 v_2 + 2 u_1 v_1 + 2 u_2 v_2 + 1$$

$$\Phi(x_1, x_2) = (x_1^2, x_2^2, \sqrt{2}x_1x_2, \sqrt{2}x_1, \sqrt{2}x_2, 1)$$

	x_1	x_2	$l_1 = x_1^2$	$l_2 = x_2^2$	l_3	l_4	l_5	l_6
Class 1	1	1	1	1	$\sqrt{2}$	$\sqrt{2}$	$\sqrt{2}$	1
	-1	-1	1	1	$\sqrt{2}$	$-\sqrt{2}$	$-\sqrt{2}$	1
Class 2	1	-1	1	1	$-\sqrt{2}$	$\sqrt{2}$	$-\sqrt{2}$	1
	-1	1	1	1	$-\sqrt{2}$	$-\sqrt{2}$	$\sqrt{2}$	1

$\Rightarrow$ points are separable on l_3 axis

maximum margin $\sqrt{2}$



$\Rightarrow$ the optimal separating hyperplane $w = [0, 0, 1, 0, 0, 0]$ ($l_3 = 0$)

and

$w_0 = 0$

- **An SVM with a RBF kernel is equivalent with a 2 layered perceptron network**

4. <u>Overfitting</u>

- *C* is a regularization parameter
  - helps in avoiding overfitting
  - theoretically, SVMs should be highly resistant to overfitting
    - in practice, it depends on the careful choice of *C* and other hyperparameters (e.g. the parameters of the kernel)
    - large $C \Rightarrow$ minimize the number of misclassified points
    - small $C \Rightarrow$ maximize the functional margin

5. <u>LIBSVM</u>

- implements the C-SVM algorithm
- for hyperparameters optimization (C and the parameters of the kernel) a grid search procedure is used
  - repeated trials for each parameter across a specified interval using geometric steps
  - for each combination of these parameters, a 10-fold cross-validation is performed during training, the quality of the combination being computed as the average of the accuracy rates estimated for each of the 10 divisions of the data
- the training step is computationally expensive

6. SVM related research topics

- SVR (regression)
- Deep SVMs
- Deep SVMs for regression
- Using SVMs in an ensemble learning
  - Boosted SVMs
  - Bagging SVMs
  - Stacking
    - base model
- Fuzzy SVMs
- Lazy SVMs
- Combining kernels for SVMs
- Hybrid models
  - SVM + ANN (Artificial Neural Networks)
  - SVM + *k*NN (*k*-Nearest Neighbors)
  - SVM + NBC (Naive Bayes Classifier)

[SLIDES]

– Introduction to Support Vector Machines (Ming-Hsuan Yang and Antoine Cornuejols) [6]

[READING]

– Support Vector Machines Explained (T. Fletcher) [1]
– Support Vector Machines (Ch. 23) [2]
– A Practical Guide to Support Vector Classification (Hsu et al.) [3]

Bibliography

[1] Tristan Fletcher, *Support Vector Machines Explained*, 2008, UCL

[2] *Support Vector Machines*, Chapter 23, Cambridge University Press, 2012

[3] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin, *A Practical Guide to Support Vector Classification*, Taiwan, 2009

[4] *The Simplified SMO Algorithm*

[5] John Platt, *Fast Training of Support Vector Machines using Sequential Minimal Optimization*, Microsoft Research, USA