Machine Learning for Big Data: Support Vector Machine

Lionel Fillatre

fillatre@unice.fr

Topics

- Introduction
- Large Margin SVM
- Compute the Large Margin SVM
- Soft Margin SVM
- Large-Scale Soft Margin SVM
- Multi-class SVM
- Non-linear SVM
- Conclusion

1 Introduction

Classification

- Everyday, all the time, we classify things. Some examples:
  - Is the digit a 9?

  - Will this patient survive?

  - Will this client click on my add?

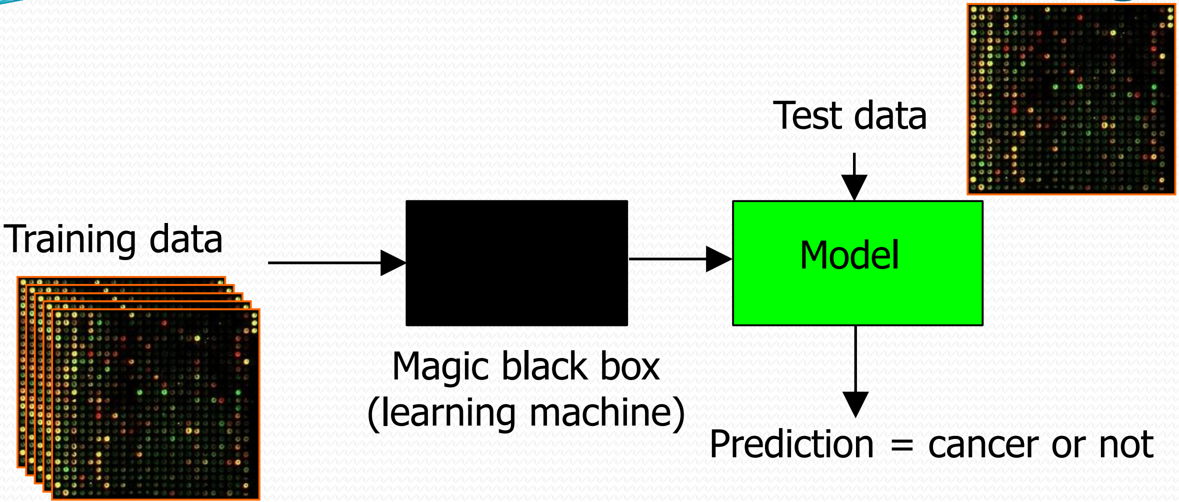
Problems in classifying data

- Often high dimension of data.
- Hard to put up simple rules.
- Amount of data.
- Need automated ways to deal with the data.
- Use computers: data processing, statistical analysis, try to learn patterns from the data (machine Learning).

Support Vector Machine (SVM)

- A classifier derived from statistical learning theory by Vapnik, et al. in 1992
- SVM became famous when, using images as input, it gave accuracy comparable to neural-network with hand-designed features in a handwriting recognition task
- Currently, SVM is widely used in object detection & recognition, content-based image retrieval, text recognition, biometrics, speech recognition, etc.
- Also used for regression (not cover in this course)

Black box view of Machine Learning



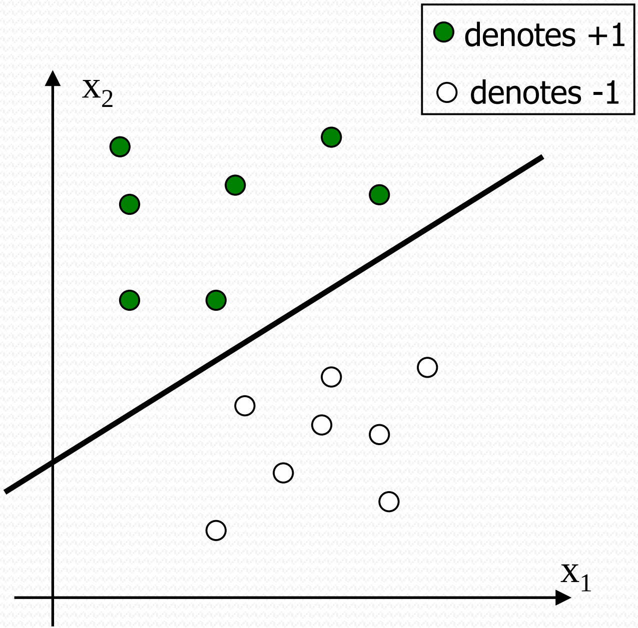
- **Training data:** expression patterns of some cancer + expression data from healty person
- **Test data:** unlabelled data of both classes
- **Model**: the model can distinguish between healty and sick persons. Can be used for prediction.

2 Large Margin SVM

Linear Discriminant Function

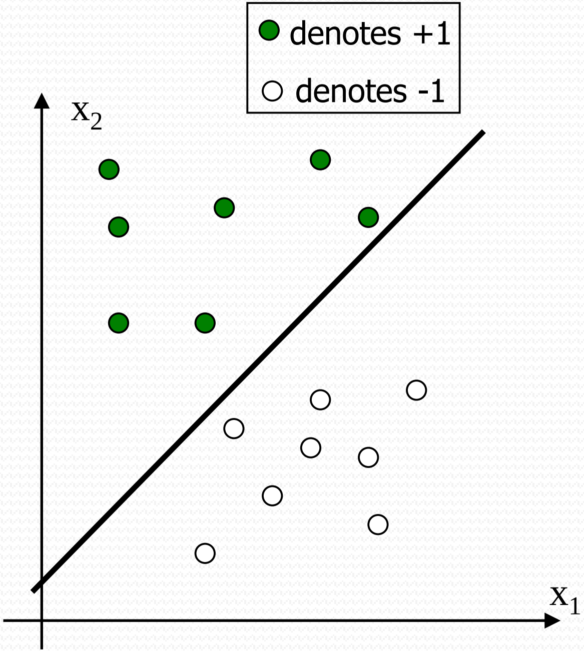
- How would you classify these points using a linear discriminant function in order to minimize the error rate?

- Infinite number of answers!



Linear Discriminant Function

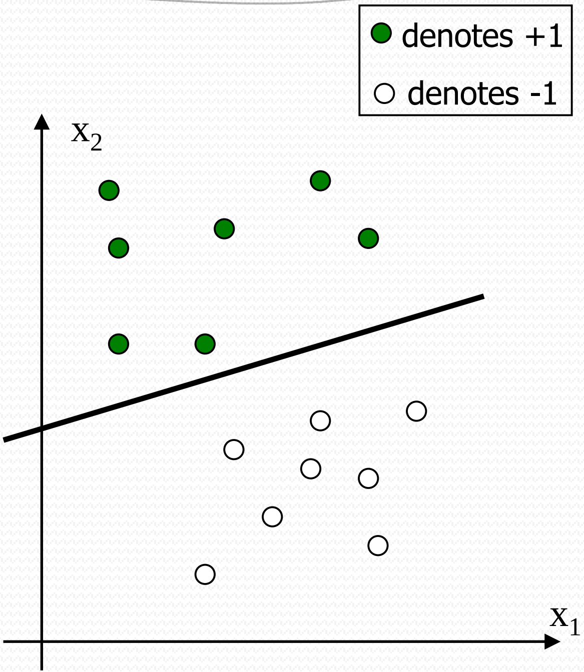
- How would you classify these points using a linear discriminant function in order to minimize the error rate?
- Infinite number of answers!



Linear Discriminant Function

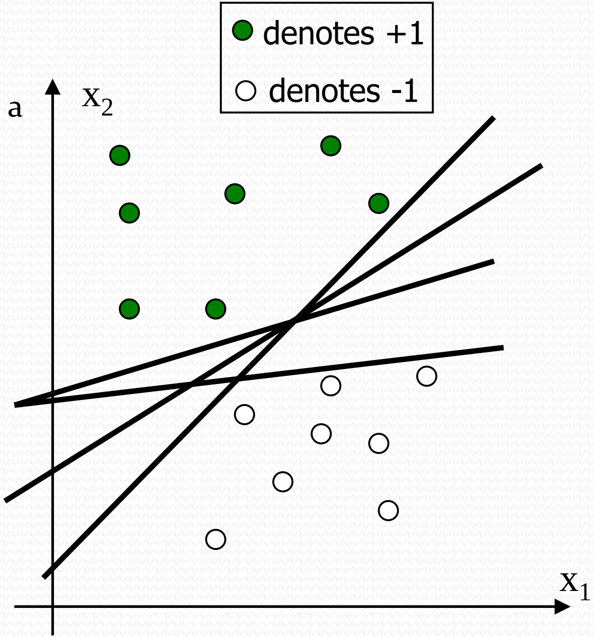
- How would you classify these points using a linear discriminant function in order to minimize the error rate?

- Infinite number of answers!



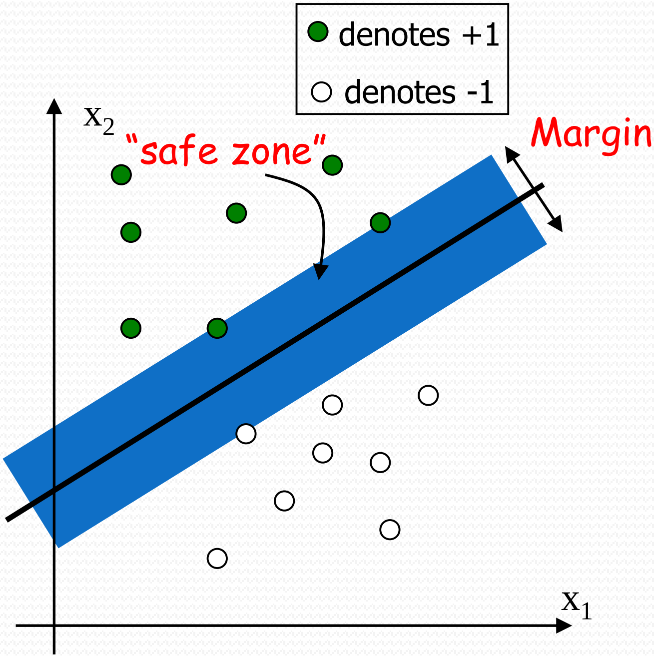
Linear Discriminant Function

- How would you classify these points using a linear discriminant function in order to minimize the error rate?
- Infinite number of answers!
- Which one is the best?



Large Margin Linear Classifier

- The linear discriminant function (classifier) with the maximum margin is the best.
- Margin is defined as the width that the boundary could be increased by before hitting a data point
- Why it is the best?
  - Robust to outliners and thus strong generalization ability



Large Margin Linear Classifier

- Given a set of data points:

$$\{(\mathbf{x}_i, y_i)\},\ i = 1, 2, \cdots, n \quad \text{, where}$$

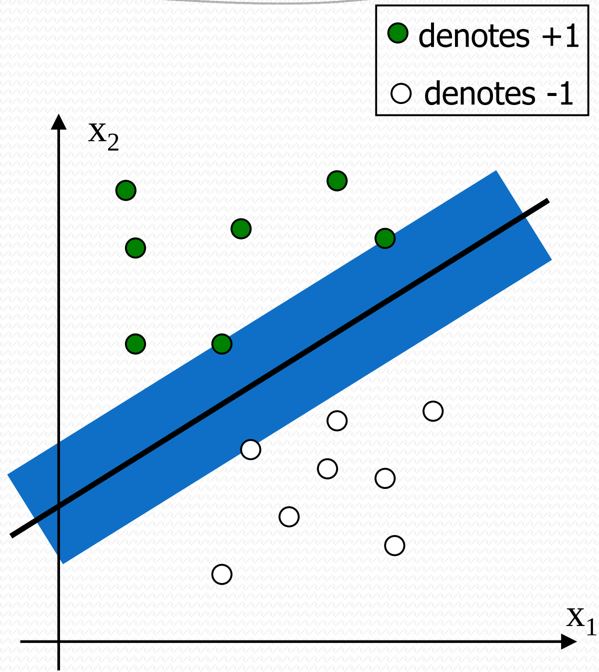
$$\text{For } y_i = +1,\quad \mathbf{w}^T \mathbf{x}_i + b > 0$$

$$\text{For } y_i = -1,\quad \mathbf{w}^T \mathbf{x}_i + b < 0$$

- With a scale transformation on both *w* and *b*, the above is equivalent to

$$\text{For } y_i = +1,\quad \mathbf{w}^T \mathbf{x}_i + b \geq 1$$

$$\text{For } y_i = -1,\quad \mathbf{w}^T \mathbf{x}_i + b \leq -1$$



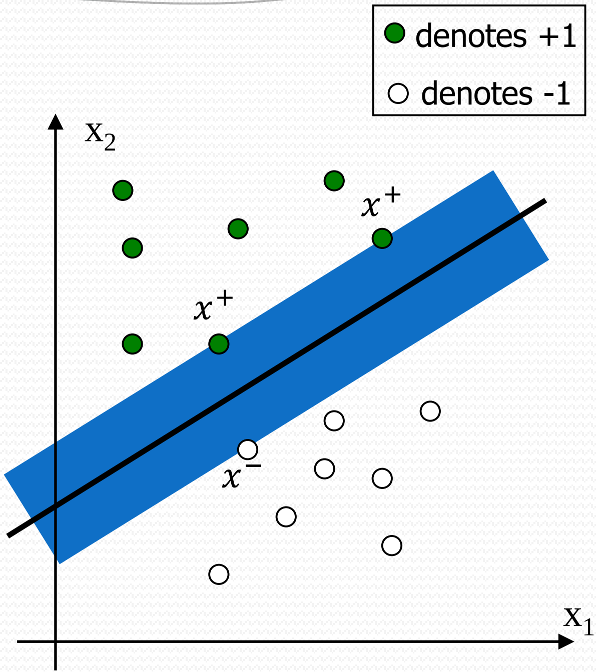
Proof of normalization

- Since $w^T x + b = 0$ and $c(w^T x + b) = 0$ define the same plane, we have the freedom to choose the normalization of w

- Choose normalization such that
  - $w^T x^+ + b = +1$
  - $w^T x^- + b = -1$

  for the positive and negative **support vectors** respectively

- Support vectors are the data points that lie closest to the decision surface



Large Margin Linear Classifier

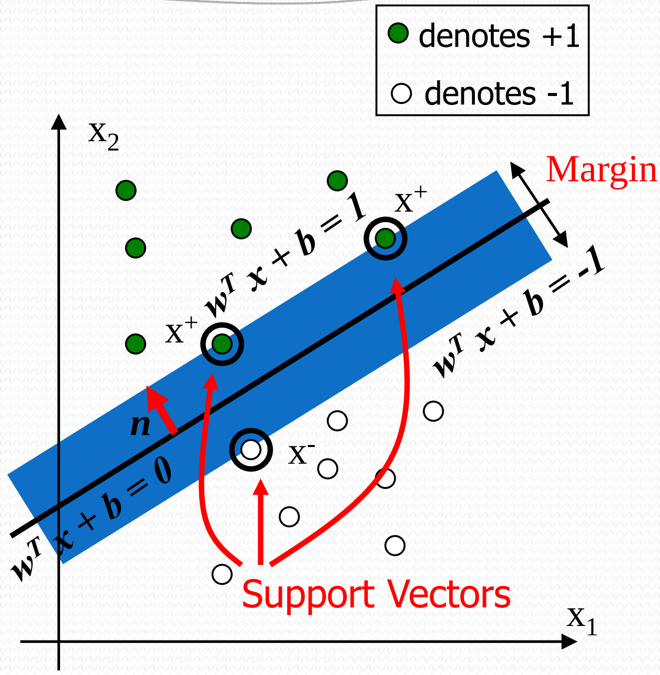
- We know that

$$\mathbf{w}^T \mathbf{x}^+ + b = 1$$

$$\mathbf{w}^T \mathbf{x}^- + b = -1$$

- The margin width M is:

$$\begin{aligned} \mathrm{M} &= (x^+ - x^-) \cdot \boldsymbol{n} \\ &= (x^+ - x^-) \cdot \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|} \\ &= \frac{1}{\|\boldsymbol{w}\|} \boldsymbol{w}^T (x^+ - x^-) \\ &= \frac{1}{\|\boldsymbol{w}\|} (1 - b - (-1 - b)) \\ &= \frac{2}{\|\boldsymbol{w}\|} \end{aligned}$$



Large Margin Linear Classifier

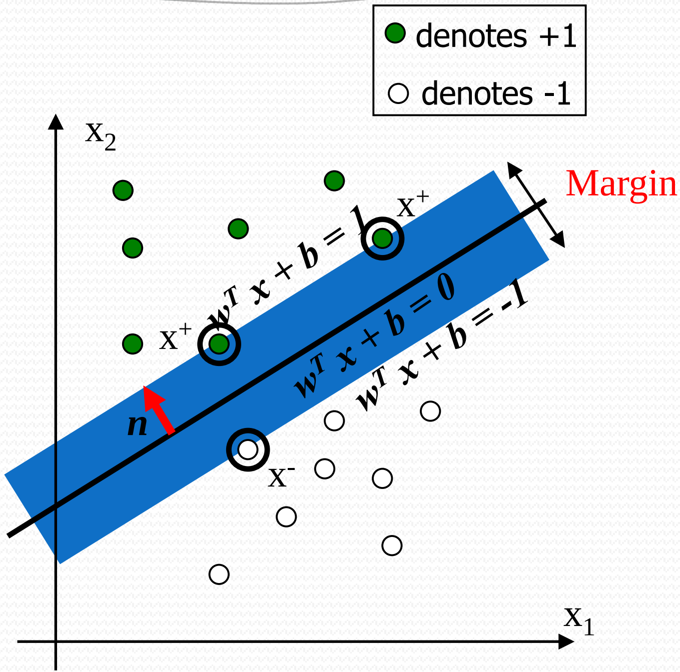
- Formulation:

$$\text{maximize} \quad \frac{2}{\|\mathbf{w}\|}$$

such that

$$\text{For } y_i = +1, \quad \mathbf{w}^T\mathbf{x}_i + b \geq 1$$

$$\text{For } y_i = -1, \quad \mathbf{w}^T\mathbf{x}_i + b \leq -1$$



Large Margin Linear Classifier

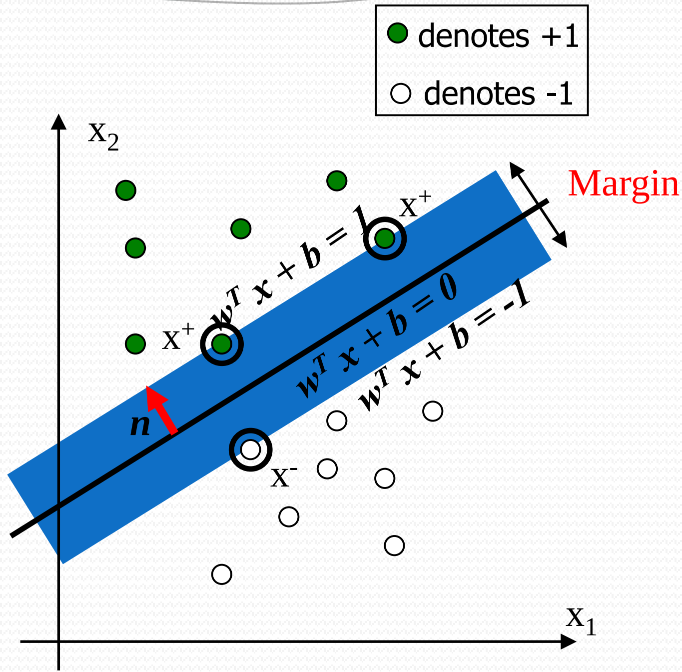
- Formulation:

$$\text{minimize} \quad \frac{1}{2}\|\mathbf{w}\|^2$$

such that

$$\text{For } y_i = +1, \quad \mathbf{w}^T\mathbf{x}_i + b \geq 1$$

$$\text{For } y_i = -1, \quad \mathbf{w}^T\mathbf{x}_i + b \leq -1$$



Large Margin Linear Classifier

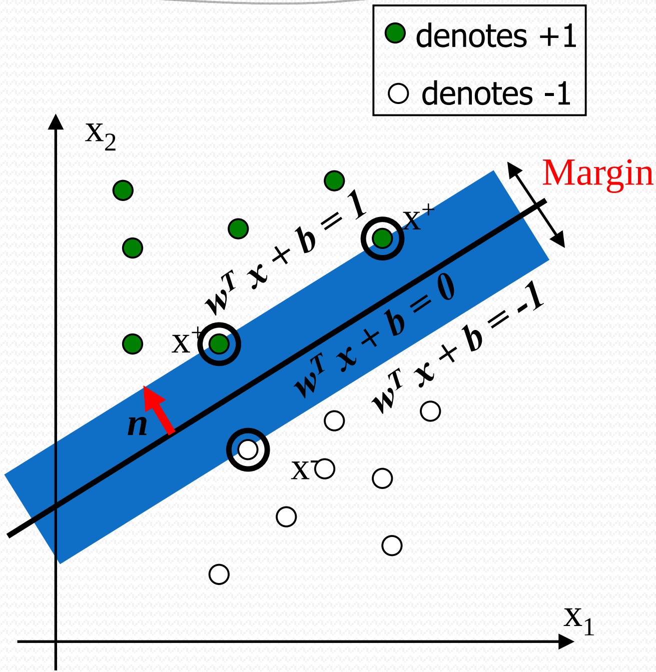
- Formulation:

$$\text{minimize} \quad \frac{1}{2}\|\mathbf{w}\|^2$$

such that

$$y_i(w^T x_i + b) \geq 1, \forall i = 1, \ldots, n$$

=> Quadratic programming with n linear constraints.



3 Compute the Large Margin SVM

Direct Optimization

- Many methods have been proposed for solving the quadratic programming problem with linear constraints.

- Many are based on optimization methods, or can be interpreted using tools from the analysis of optimization algorithms.

- Methods compared via a variety of metrics:
  - CPU time to find solution of given quality (e.g. error rate)
  - Theoretical efficiency
  - Data storage requirements
  - Simplicity

- However, directly solving this problem is difficult because the constraints are quite complex and there is a lack of interpretation => solving the dual problem!

Solving the Optimization Problem

Quadratic programming with linear constraints

$$\text{minimize} \quad \frac{1}{2}\|\mathbf{w}\|^2$$

$$\text{s.t.} \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$$

Lagrangian Problem

$$\text{minimize} \quad L_p(\mathbf{w}, b, \alpha_i) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i \left(y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 \right)$$

$$\text{s.t.} \quad \alpha_i \geq 0$$

Solving the Optimization Problem

$$\text{minimize } \; L_p(\mathbf{w}, b, \alpha_i) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i \left(y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 \right)$$

$$\text{s.t.} \qquad \alpha_i \geq 0$$

$$\frac{\partial L_p}{\partial \mathbf{w}} = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$$

$$\frac{\partial L_p}{\partial b} = 0 \quad \Rightarrow \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

Solving the Optimization Problem

$$\text{minimize} \;\; L_p(\mathbf{w}, b, \alpha_i) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i \left(y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 \right)$$

$$\text{s.t.} \qquad \alpha_i \geq 0$$

Lagrangian Dual Problem

$$\text{maximize} \;\; \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

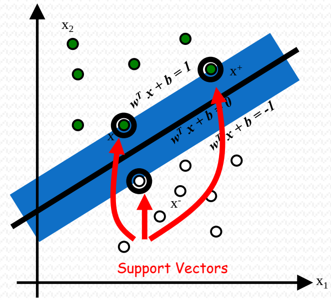
$$\text{s.t.} \qquad \alpha_i \geq 0 \text{, and} \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

Solving the Optimization Problem

- From KKT condition, we know:

$$\alpha_i \left(y_i (\mathbf{w}^T \mathbf{x}_i + b) - 1 \right) = 0$$

- Thus, only support vectors can have $\alpha_i \neq 0$



- The solution has the form:

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = \sum_{i \in \mathrm{SV}} \alpha_i y_i \mathbf{x}_i$$

get b from $y_i (\mathbf{w}^T \mathbf{x}_i + b) - 1 = 0$,

where $\mathbf{x}_i$ is support vector

Solution of the Optimization Problem

- The linear discriminant function is:

$$g(\boldsymbol{x}) = \boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x} + b = \sum_{i \in SV} \alpha_i y_i \boldsymbol{x}_i^T \boldsymbol{x} + b$$

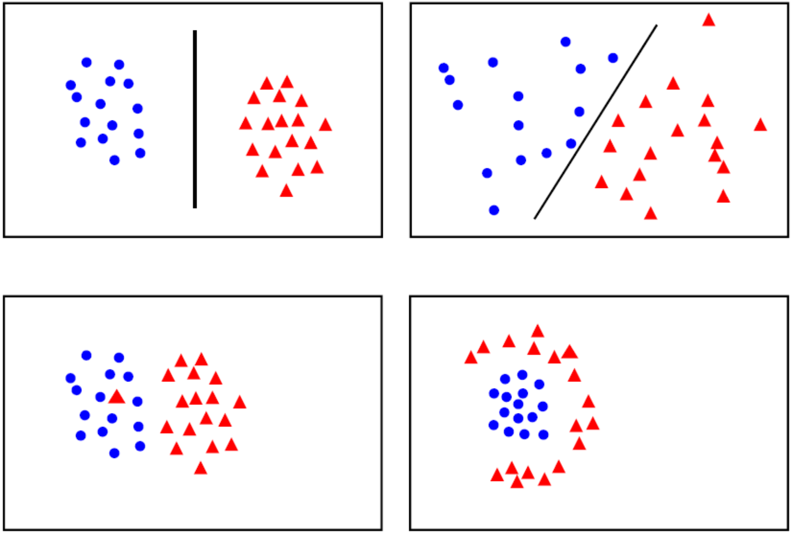
- Notice it relies on a dot product between the test point $\boldsymbol{x}$ and the support vectors $\boldsymbol{x}_i$

- Also bear in mind that solving the optimization problem involved computing the dot products $\boldsymbol{x}_i^T \boldsymbol{x}_j$ between all pairs of training points

4 Soft Margin SVM

Linear separability

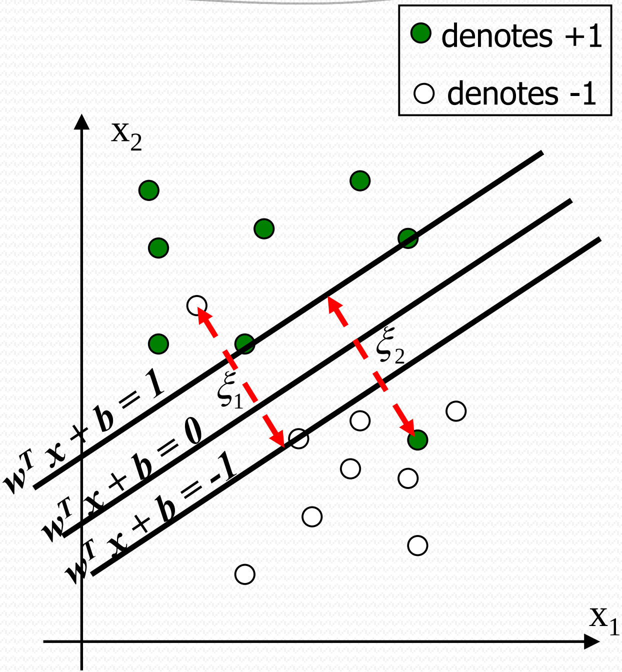
Linearly separable



Not linearly separable

Soft Margin Linear Classifier

- What if data is not linear separable? (noisy data, outliers, etc.)
- Slack variables ξ_i can be added to allow mis-classification of difficult or noisy data points
- Still, try to minimize training set errors, and to place hyperplane "far" from each class (large margin)



Soft Margin Linear Classifier

- Formulation:

$$\text{minimize} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$

such that

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

- Parameter *C* can be viewed as a way to control over-fitting: it “trades off” the relative importance of maximizing the margin and fitting the training data.

Soft Margin Classification – Solution

- The dual problem for soft margin classification:

> Find $\alpha_1 \ldots \alpha_N$ such that
> $\mathbf{Q}(\boldsymbol{\alpha}) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j \mathbf{x_i}^T\mathbf{x_j}$ is maximized and
> (1) $\Sigma\alpha_i y_i = 0$
> (2) $0 \leq \alpha_i \leq C$ for all α_i

- Neither slack variables ξ_i nor their Lagrange multipliers appear in the dual problem!
- Again, $\mathbf{x_i}$ with non-zero α_i will be support vectors.
- Solution to the dual problem is:

$$\mathbf{w} = \Sigma\alpha_i y_i \mathbf{x_i}$$
$$b = y_k(1 - \xi_k) - \mathbf{w}^T\mathbf{x}_k$$
$$\text{where } k = \underset{i}{\operatorname{argmax}}\, \alpha_i$$

w is not needed explicitly for classification!

$$f(\mathbf{x}) = \Sigma\alpha_i y_i \mathbf{x_i}^T\mathbf{x} + b$$

5 Large-scale Soft Margin SVM

Rewiting the Soft Margin SVM

$$\text{minimize} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$

such that

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

$$\xi_i = \begin{cases} 0 & \text{if } y_i(\mathbf{w}^T x_i + b) \geq 1 \\ 1 - y_i(\mathbf{w}^T x_i + b) & \text{otherwise} \end{cases}$$

Understanding the Soft Margin SVM

$$\xi_i = \begin{cases} 0 & \text{if } y_i(\mathbf{w}^T x_i + b) \geq 1 \\ 1 - y_i(\mathbf{w}^T x_i + b) & \text{otherwise} \end{cases}$$

$$\Downarrow$$

$$\begin{aligned} \xi_i &= \max(0, 1 - y_i(\mathbf{w}^T x_i + b)) \\ &= \max(0, 1 - y_i \hat{y}_i) \end{aligned}$$

Rewriting the Soft Margin SVM

$$\text{minimize} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$

such that

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i$$

$$\xi_i = \max(0, 1 - y_i(\mathbf{w}^T x_i + b))$$

$$\xi_i \geq 0$$

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\max(0, 1 - y_i(\mathbf{w}^T x_i + b))$$

Unconstrained problem!

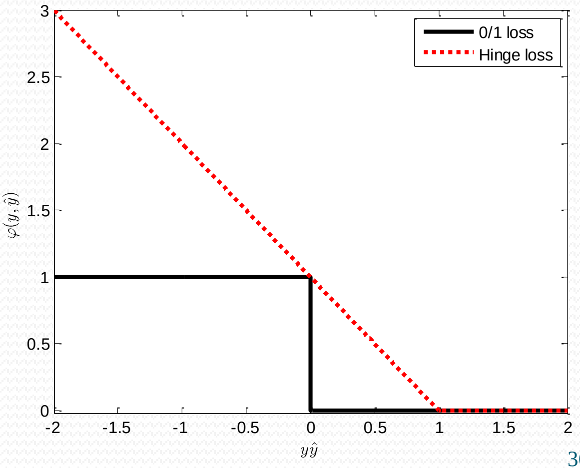
Hinge loss and its interpretation

The problem $\min_{\mathbf{w},b} \ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\max(0, 1 - y_i(\mathbf{w}^T x_i + b))$

can be interpreted as the relaxation of $\min_{\mathbf{w},b} \ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} 1\left[y_i(\mathbf{w}^T x_i + b) \le 0\right]$

0/1 loss: $\varphi_{0/1}(y,\hat{y}) = 1\left[y\hat{y} \le 0\right]$

Hinge loss: $\varphi_{\text{hinge}}(y,\hat{y}) = \max(0, 1 - y\hat{y})$



Reinterpreting the Soft Margin SVM

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \varphi_{hinge}(y_i, \mathbf{w}^T x_i + b)$$

is equivalent to

$$\min_{\mathbf{w},b} \sum_{i=1}^{n} \varphi_{hinge}(y_i, \mathbf{w}^T x_i + b) + \lambda\|\mathbf{w}\|^2$$

- It a typical optimization problem with a regularization term (hinge loss + L2 regularization):

$$\operatorname*{argmin}_{\mathbf{w},b} \sum_{i=1}^{n} \text{loss}(y_i, \hat{y}_i) + \lambda \text{ regularizer}(\mathbf{w}, b)$$

Gradient descent problem!

Warning!

- The hinge loss is not diffentiable!

$$\ell(\mathbf{w},b)=\varphi_{\text{hinge}}(y,\mathbf{w}^T x+b)=\max(0,1-y(\mathbf{w}^T x+b))$$

$$\frac{\partial \ell}{\partial w_j}(\mathbf{w},b)=\begin{cases} 0 \quad \text{if} \quad y(\mathbf{w}^T x+b)>1 \\ \text{-}yx_i \quad \text{if} \quad y(\mathbf{w}^T x+b)<1 \\ ??? \quad \text{if} \quad y(\mathbf{w}^T x+b)=1 \end{cases}$$

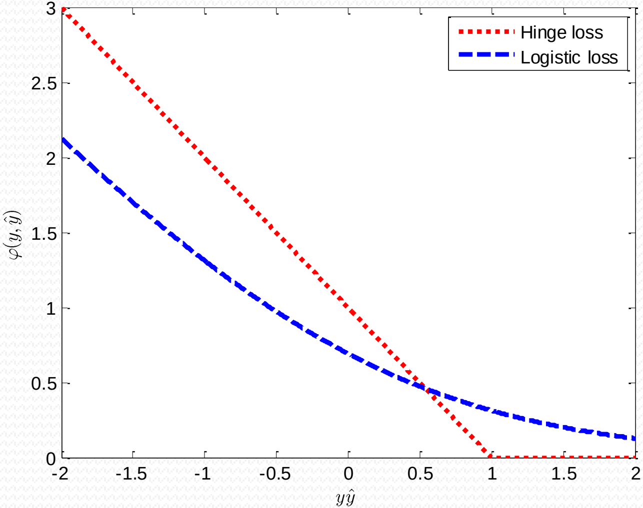
- Calculating the gradient is impossible.

- Two solutions:
  - Subgradient optimization algorithms (not studied in this lecture)
  - Smoothing the loss

Smoothing the loss

Hinge loss: $\varphi_{\text{hinge}}(y,\hat{y}) = \max(0, 1 - y\hat{y})$

Logistic loss: $\varphi_{\log}(y,\hat{y}) = \log(1 + e^{-y\hat{y}})$



Smoothing the loss

- Come back to the Hinge loss approximation
- Select a differentiable approximation (logistic loss for example)

$$\min_{\mathbf{w},b} \sum_{i=1}^{n} \log\left(1 + e^{-y_i\left(\mathbf{w}^T x_i + b\right)}\right) + \lambda \|\mathbf{w}\|^2 = \min_{\mathbf{w},b} \mathrm{L}(\mathbf{w}, b)$$

- Calculate the gradient:

$$\nabla_{\mathbf{w}} \mathrm{L}(\mathbf{w}, b) = -\sum_{i=1}^{n} y_i x_i \frac{e^{-y_i\left(\mathbf{w}^T x_i + b\right)}}{1 + e^{-y_i\left(\mathbf{w}^T x_i + b\right)}}$$

$$\nabla_{b} \mathrm{L}(\mathbf{w}, b) = -\sum_{i=1}^{n} y_i \frac{e^{-y_i\left(\mathbf{w}^T x_i + b\right)}}{1 + e^{-y_i\left(\mathbf{w}^T x_i + b\right)}}$$

- Solve with a gradient descent algorithm

Large scale processing

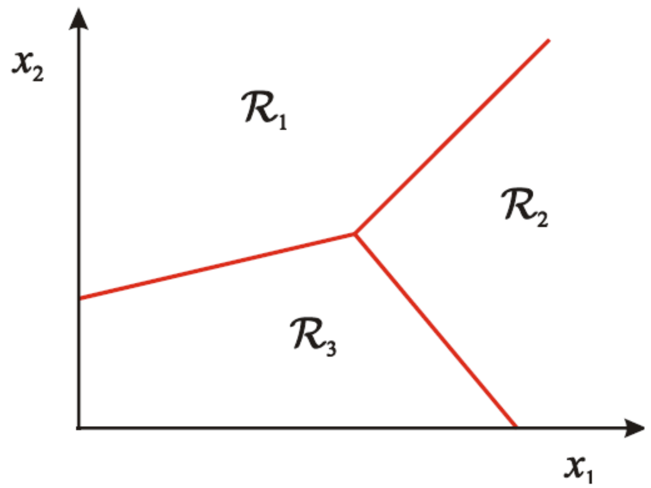
- Gradient descent is well adapted to distributed computing
- Gradient computation is well adapted to clusterized data.
- Good solvers:
  - SVM Torch: http://www.torch.ch
  - libSVM: http://www.csie.ntu.edu.tw/~cjlin/libsvm/

6 Multi-class SVM

Multi-Class Classification

- Assign input vector $\boldsymbol{x}$ to one of K classes C_k
- Goal: a decision rule that divides input space into K decision regions R_k separated by decision boundaries

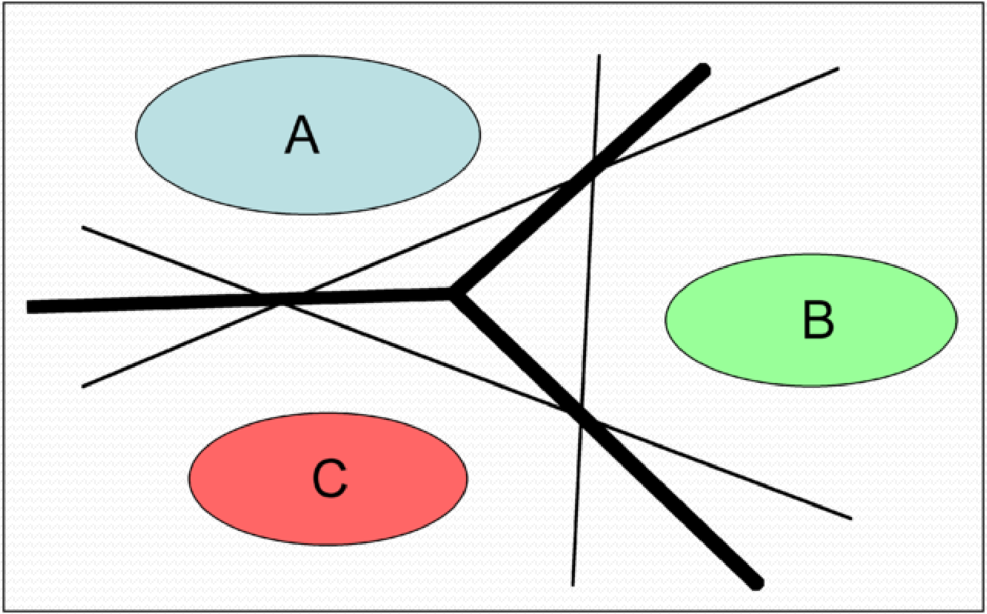
- Three main methods:
  - One-versus-all
  - One-versus-one
  - Multi-class



One-Versus-All

- Method for N classes:
  - For each class C_k learn binary classifier
    $$h_k(x) = w_k^T x + b_k$$
  - Combine binary classifiers via voting mechanism
    - Example: choose class C_i such as
      $$h_i(x) = \max_{1 \le k \le N} h_k(x)$$
- Drawbacks:
  - Calibration: classifier scores not comparable.
  - Nevertheless: simple and frequently used in practice, computational advantages in some cases.

One-Versus-All: 3 classes A, B and C

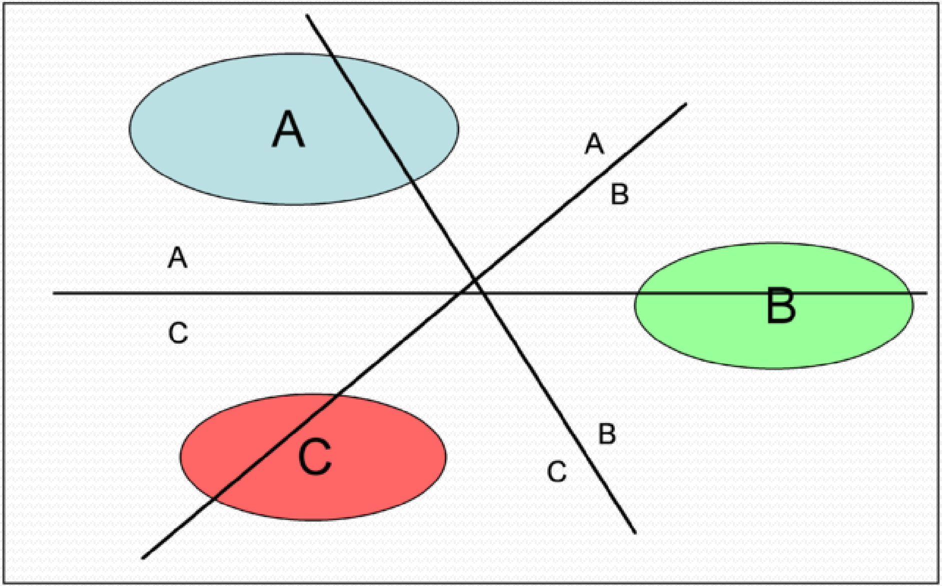


One-Versus-One

- Method for N classes:
  - For each pair $(\ell, \ell'), \ell \neq \ell'$, learn binary classifier
    $$h_{(\ell,\ell')}(x) = w_{(\ell,\ell')}^T x + b_{(\ell,\ell')}$$
  - Combine binary classifiers via voting mechanism
    - Example (majority vote): choose class C_i such as
      $$i = \operatorname*{argmax}_{1 \leq \ell' \leq N} \left| \left\{ \ell : h_{(\ell,\ell')}(x) > 0 \right\} \right|$$
      where $|A|$ is the number of elements in the set A.

- Drawbacks:
  - Computational: train binary classifiers.
  - Overfitting: size of training sample could become small for a given pair.

One-Versus-All: 3 classes A, B and C



Multi-class SVM (3 classes case)

- Learn $\boldsymbol{w} = (\boldsymbol{w_1}, \boldsymbol{w_2}, \boldsymbol{w_3})$ solving the problem:

$$\min_{\mathbf{w}} ||\mathbf{w}||^2 \quad \text{subject to}$$

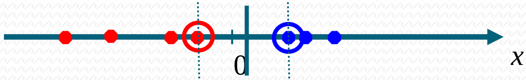
$$\begin{array}{lll}
{\mathbf{w}_1}^\top \mathbf{x}_i \geq {\mathbf{w}_2}^\top \mathbf{x}_i \;\; \& \;\; {\mathbf{w}_1}^\top \mathbf{x}_i \geq {\mathbf{w}_3}^\top \mathbf{x}_i & & \text{for } i \in \text{class } 1 \\
{\mathbf{w}_2}^\top \mathbf{x}_i \geq {\mathbf{w}_3}^\top \mathbf{x}_i \;\; \& \;\; {\mathbf{w}_2}^\top \mathbf{x}_i \geq {\mathbf{w}_1}^\top \mathbf{x}_i & & \text{for } i \in \text{class } 2 \\
{\mathbf{w}_3}^\top \mathbf{x}_i \geq {\mathbf{w}_1}^\top \mathbf{x}_i \;\; \& \;\; {\mathbf{w}_3}^\top \mathbf{x}_i \geq {\mathbf{w}_2}^\top \mathbf{x}_i & & \text{for } i \in \text{class } 3
\end{array}$$

- This is a quadratic optimization problem subject to linear constraints and there is a unique minimum
- A margin can also be included in the constraints
- In practice there is a little or no improvement over the multiple binary classification approach.

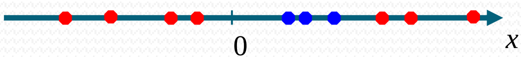
7 Non-Linear SVM

Non-linear SVMs

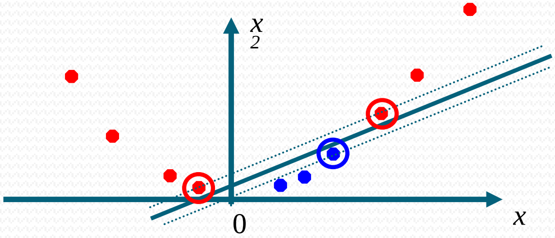
- Datasets that are linearly separable with some noise work out great:



- But what are we going to do if the dataset is just too hard?

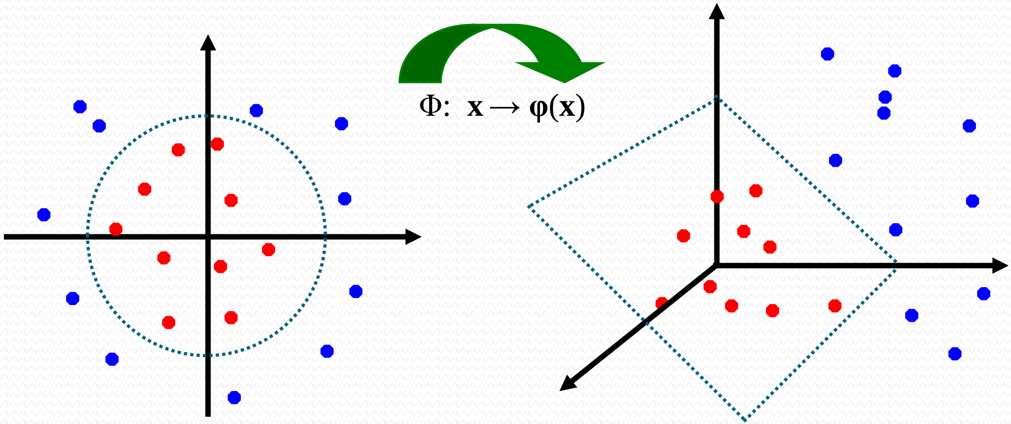


- How about… mapping data to a higher-dimensional space:



Non-linear SVMs: Feature spaces

- General idea: the original feature space can always be mapped to some higher-dimensional feature space where the training set is separable:



8 Conclusion

SVM applications

- SVMs are relevant for a number of classification tasks ranging from text to genomic data.
- SVMs can be applied to complex data types beyond feature vectors (e.g. graphs, sequences, relational data) by designing kernel functions for such data.
- SVM techniques have been extended to a number of tasks such as regression, principal component analysis, etc.
- Tuning SVMs remains a black art: selecting a specific kernel and parameters is usually done in a try-and-see manner.