SUPPORT VECTOR MACHINE

Dr. Hilman F. Pardede

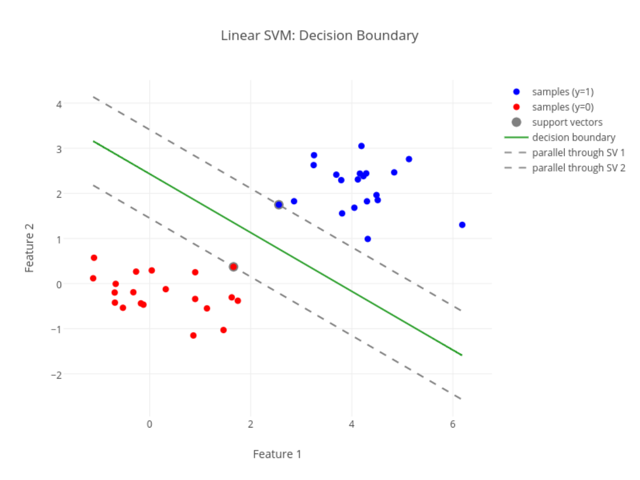
Research Center for Informatics
Indonesian Institute of Sciences

The materials are compiled from the following resources:

- https://github.com/joaquinvanschoren/ML-course
- https://www.cse.iitk.ac.in/users/piyush/courses/ml_autumn16/ML.html
- http://sli.ics.uci.edu/Classes/2015W-273a

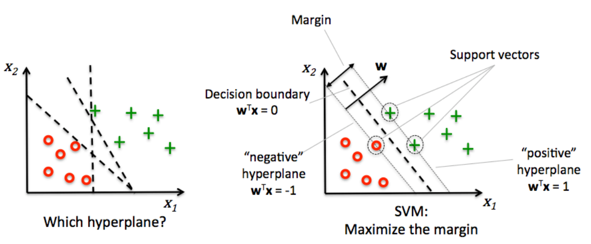
SVM BASICS

Find hyperplane maximizing the margin between the classes

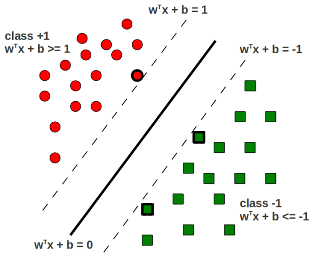


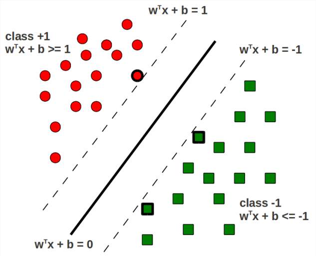
- Aims to find a (hyper)plane that separates the examples of each class.
- For binary classification (2 classes), we aim to fit the following function:
  $\hat{y} = w_0 * x_0 + w_1 * x_1 + ... + w_p * x_p + b > 0$
- When $\hat{y} < 0$, predict class -1, otherwise predict class $+1$

- In several other linear models, we minimized (misclassification) error
- In SVMs, the optimization objective is to maximize the margin
- The margin is the distance between the separating hyperplane and the support vectors
- The support vectors are the training samples closest to the hyperplane
- Intuition: large margins generalize better, small margins may be prone to overfitting

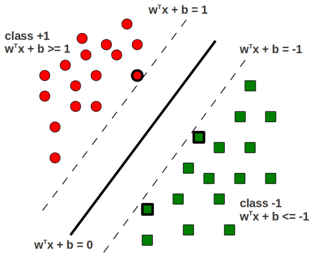


MAXIMUM MARGIN

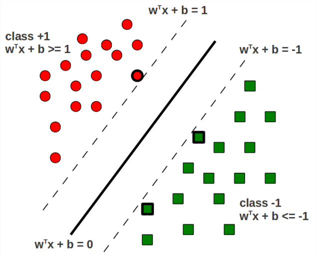




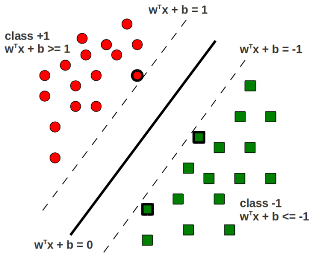
- For now, we assume that the data is linearly separable.



- For now, we assume that the data is linearly separable.
- The positive hyperplanes is defined as: $b + \mathbf{w}^{\mathsf{T}}\mathbf{x}_{+} = 1$ with $\mathbf{x}_{+}$ the positive support vectors.

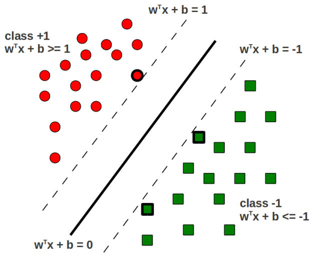


- For now, we assume that the data is linearly separable.
- The positive hyperplanes is defined as: $b + \mathbf{w}^{\mathsf{T}}\mathbf{x}_{+} = 1$ with $\mathbf{x}_{+}$ the positive support vectors.
- Likewise, the negative hyperplanes is defined as: $b + \mathbf{w}^{\mathsf{T}}\mathbf{x}_{-} = -1$



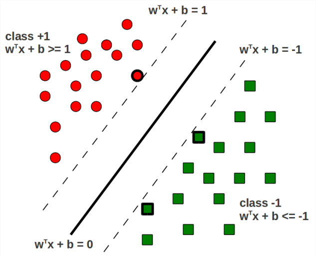
- For now, we assume that the data is linearly separable.
- The positive hyperplanes is defined as: $b + \mathbf{w}^\mathsf{T}\mathbf{x}_+ = 1$ with $\mathbf{x}_+$ the positive support vectors.
- Likewise, the negative hyperplanes is defined as: $b + \mathbf{w}^\mathsf{T}\mathbf{x}_- = -1$

- Substracting them yields: $\mathbf{w}^\mathsf{T}(\mathbf{x}_+ - \mathbf{x}_-) = 2$



- For now, we assume that the data is linearly separable.
- The positive hyperplanes is defined as: $b + \mathbf{w}^{\mathbf{T}}\mathbf{x}_{+} = 1$ with $\mathbf{x}_{+}$ the positive support vectors.
- Likewise, the negative hyperplanes is defined as: $b + \mathbf{w}^{\mathbf{T}}\mathbf{x}_{-} = -1$

- Substracting them yields: $\mathbf{w}^{\mathbf{T}}(\mathbf{x}_{+} - \mathbf{x}_{-}) = 2$
- We can normalize by the length of vector w, defined as $||w|| = \sqrt{\sum_{j=1}^{m} w_j^2}$



- For now, we assume that the data is linearly separable.
- The positive hyperplanes is defined as: $b + \mathbf{w}^\mathsf{T}\mathbf{x}_+ = 1$ with $\mathbf{x}_+$ the positive support vectors.
- Likewise, the negative hyperplanes is defined as: $b + \mathbf{w}^\mathsf{T}\mathbf{x}_- = -1$

- Substracting them yields: $\mathbf{w}^\mathsf{T}(\mathbf{x}_+ - \mathbf{x}_-) = 2$
- We can normalize by the length of vector w, defined as $||w|| = \sqrt{\sum_{j=1}^{m} w_j^2}$
- Yielding $\frac{\mathbf{w}^\mathsf{T}(\mathbf{x}_+ - \mathbf{x}_-)}{||w||} = \frac{2}{||w||}$ which is the margin that we want to maximize.

- Hence, we want to maximize $\frac{2}{||w||}$
- Maximizing $\frac{2}{||w||}$ can be done by minimizing $\frac{||w||^2}{2}$
- The constraints are that all samples are classified correctly:
  $b + \mathbf{w^T x^{(i)}} \geq 1 \;\; if \;\; y^{(i)} = 1 \;\; b + \mathbf{w^T x^{(i)}} \leq -1 \;\; if \;\; y^{(i)} = -1$
  i.e. all negative examples should fall on one side of the negative hyperplane and vice versa.
- so our optimization problem would be:
  Minimize $\frac{||w||^2}{2}$
  Subject to $y^{(i)}(b + \mathbf{w^T x^{(i)}}) \geq 1 \;\; \forall i$
- This is a Quadratic Program (QP) with N linear inequality constraints
- Can be solved using Langrage multipliers

- The Primal formulation of the Lagrangian objective function is:

$$\min L_P = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{l} a_i y_i(\mathbf{x_i} * \mathbf{w} + b) + \sum_{i=1}^{l} a_i \tag{1}$$

so that

$$\forall i a_i \geq 0 \tag{2}$$

$$\mathbf{w} = \sum_{i=1}^{l} a_i y_i \mathbf{x_i} \tag{3}$$

$$\sum_{i=1}^{l} a_i y_i = 0 \tag{4}$$

with l the number of training examples and a the dual variable, which acts like a weight for each training example.

- It has a Dual formulation as follows:

$$\min L_D(a_i) = \sum_{i=1}^{l} a_i - \frac{1}{2}\sum_{i=1}^{l} a_i a_j y_i y_j (\mathbf{x_i}.\mathbf{x_j}) \tag{5}$$

so that

$$\forall i a_i \geq 0 \tag{6}$$

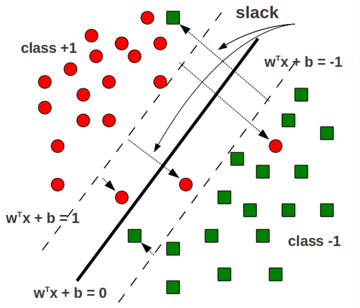
$$\sum_{i=1}^{l} a_i y_i = 0 \tag{7}$$

Dual form is interesting:

- Because now we can solve the problem by just computing the inner products of $\mathbf{x_i}, \mathbf{x_j}$
- Knowing the dual coefficients a_i we can find the weights w for the maximal margin separating hyperplane: $\mathbf{w} = \sum_{i=1}^{l} a_i y_i \mathbf{x_i}$
- Hence, we can classify a new sample $\mathbf{u}$ by looking at the sign of $\mathbf{w} * \mathbf{u} + b$
- Most of the a_i will turn out to be 0
- The training samples for which a_i is not 0 are the support vectors
- Hence, the SVM model is completely defined by the support vectors and their coefficients

DEALING WITH NONLINEARLY SEPARABLE DATA: SVM WITH SOFT MARGIN

- If the data is not linearly separable, (hard) margin maximization becomes meaningless because the constraints would contradict
- We can allow for violatings of the margin constraint by introducing slack variables $\xi^{(i)}$
  $b + \mathbf{w}^{\mathbf{T}}\mathbf{x}^{(\mathbf{i})} \geq 1 - \xi^{(i)}$ *if* $y^{(i)} = 1$
  $b + \mathbf{w}^{\mathbf{T}}\mathbf{x}^{(\mathbf{i})} \leq -1 + \xi^{(i)}$ *if* $y^{(i)} = -1$



- The new objective (to be minimized) becomes:

$$\frac{||w||^2}{2} + C(\sum_i \xi^{(i)}) \tag{8}$$

- C is a penalty for misclassification
- This is known as the soft margin SVM (or large margin SVM)

- C can be used to control the size of the margin and tune the bias-variance trade-off
  - Large C: Increases bias, reduces variance, more underfitting
  - Small C: Reduces bias, increases variance, more overfitting
- The penalty term $C(\sum_i \xi^{(i)})$ acts as an L1 regularizer on the dual coefficients
  - Also known as hinge loss
  - This induces sparsity: large C values will set many dual coefficients to 0, hence fewer support vectors
  - Small C values will typically lead to more support vectors
  - Again, it depends on the data how flexible or strict you need to be
- The least squares SVM is a variant that does L2 regularization
- Will have many more support vectors (with low weights)

- A (Mercer) Kernel on a space X is a (similarity) function

$$k : X \times X \to \mathbb{R} \tag{9}$$

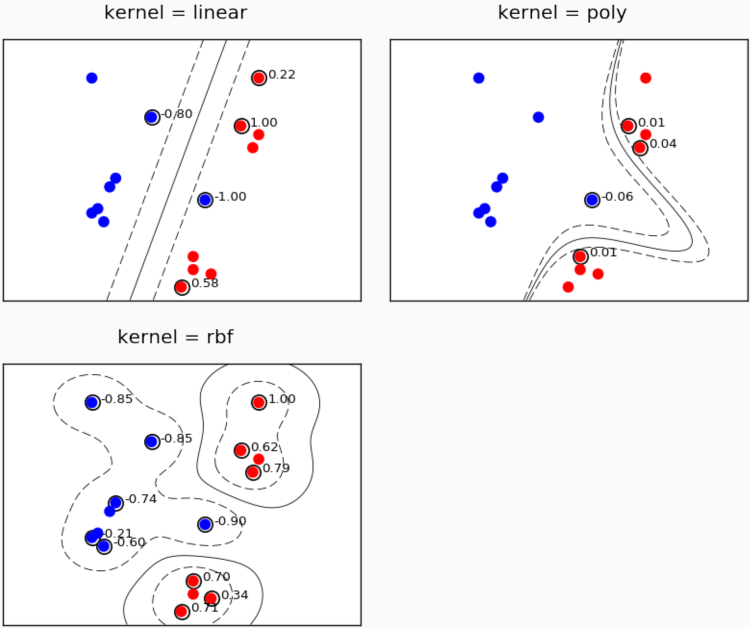
  Of two arguments with the properties:
  - Symmetry: $k(x_1, x_2) = k(x_2, x_1)\ \forall x_1, x_2 \in X$
  - Positive definite: for each finite subset of data points $x_1, ..., x_n$, the kernel Gram matrix is positive semi-definite
- Kernel matrix $= K \in \mathbb{R}^{n \times n}$ with $K_{ij} = k(x_i, x_j)$

- Mercer's Theorem states that there exists a Hilbert space $\mathcal{H}$ of continuous functions $X \to \mathbb{R}$
- basically, a possibly infinite-dimensional vector space with inner product where all operations are meaningful
- and a continuous "feature map" $\phi : X \to \mathcal{H}$
- so that the kernel computes the inner product of the features $k(x_1, x_2) = \phi(x_1), \phi(x_2)$
- Hence, a kernel can be thought of as a 'shortcut' computation for the 2-step procedure feature map + inner product
- More complex feature transformations equals to use transformation on much simpler kernel operation

KERNEL EXAMPLES

- the polynomial kernel: $k(x_1, x_2) = (x_1^T x_2 + b)^d$, for $b \geq 0$ and $d \in \mathbb{N}$
- The 'radial' Gaussian kernel: $k(x_1, x_2) = exp(-\gamma||x_1 - x_2||^2)$, for $\gamma \geq 0$
- sigmoid kernel

- Adding nonlinear features can make linear models much more powerful
- Often we don't know which features to add, and adding many features might make computation very expensive
- Mathematical trick (kernel trick) allows us to directly compute distances (scalar products) in the high dimensional space
  - We can search for the nearest support vector in the high dimensional space
- A kernel function is a distance (similarity) function with special properties for which this trick is possible
  - Polynomial kernel: computes all polynomials up to a certain degree of the original features
  - Gaussian kernel, or radial basis function (RBF): considers all possible polynomials of all degrees


kernel = linear
0.22
-0.80
1.00
-1.00
0.58
kernel = poly
0.01
0.04
-0.06
0.01
kernel = rbf
-0.85
1.00
-0.85
0.62
0.79
-0.74
-0.90
-0.21
-0.60
0.70
0.34
0.71