Data Mining

Support Vector Machines

Introduction to Data Mining, 2nd Edition
by
Tan, Steinbach, Karpatne, Kumar

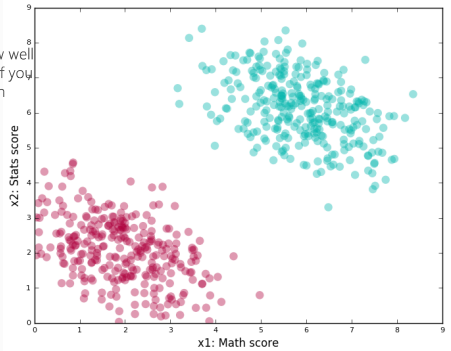
What problem are we solving?

- Consider a machine learning (ML) course offered at a university. The course instructors have observed that students get the most out of it if they are good at Math or Stats. Over time, they have recorded the scores of the enrolled students in these subjects. Also, for each of these students, they have a label depicting their performance in the ML course: "Good" or "Bad."
- Using this, can you specify a pre-requisite for enrolling in ML
- Let us represent the data that has been collected.

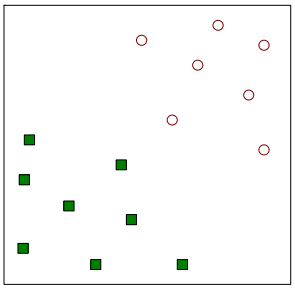
What problem are we solving?

- Green indicates students did well in ML course
- Red indicates students did not do well in ML course

For a new student,
Can you predict how well
The student will do if you
Know Student's math
and stats score?

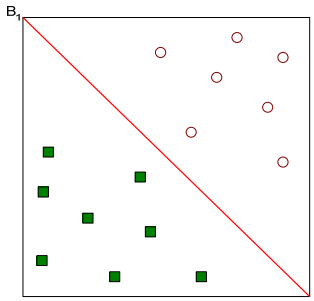


Support Vector Machines



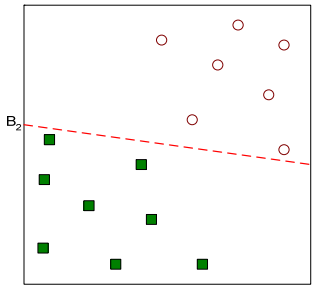
- Find a linear hyperplane (decision boundary) that will separate the data

Support Vector Machines



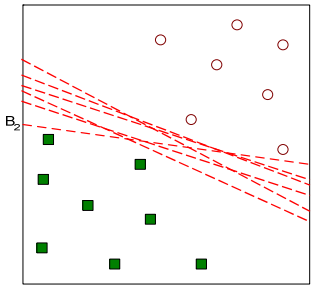
- One Possible Solution

Support Vector Machines



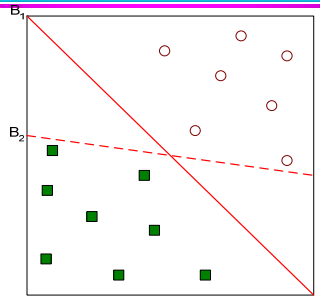
- Another possible solution

Support Vector Machines



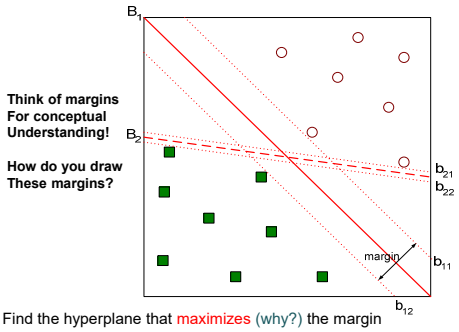
- Other possible solutions

Support Vector Machines



- What is the difference between B1 and B2? Which one is better?
- How do you define better?

Support Vector Machines



Think of margins For conceptual Understanding!

How do you draw These margins?

- Find the hyperplane that maximizes (why?) the margin
- So, B1 is better than B2! (why?)

SVMs should:

1. Find lines that correctly classify the training data
2. Among all such lines, pick the one that has the greatest distance to the points closest to it (largest margin)

➢ The closest points that identify this line are known as *support vectors*. And the region they define around the line is known as the *margin*.

➢ Support Vector Machines give you a way to pick between many possible classifiers in a way that guarantees a higher chance of correctly labeling your test data.

➢ Pretty neat, right?

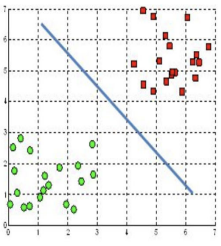
SVMs

➢ Find

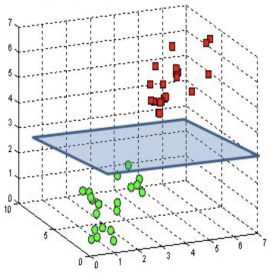
- ❖ A line in two dimensions
- ❖ A Plane in 3 dimensions.
- ❖ A Hyperplane in higher dimensions

➢ The **equation of** a **line** is typically written as y = mx + b where m is the slope and b is the y-intercept. b is 0 if it passes through origin (0, 0)

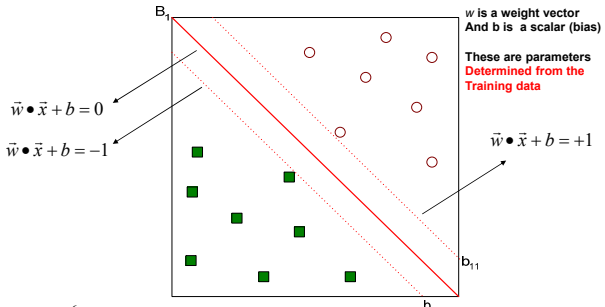
A hyperplane in $\mathbb{R}^2$ is a line



A hyperplane in $\mathbb{R}^3$ is a plane



Support Vector Machines



$$f(\vec{x}) = \begin{cases} 1 & \text{if } \vec{w} \bullet \vec{x} + b \geq 1 \\ -1 & \text{if } \vec{w} \bullet \vec{x} + b \leq -1 \end{cases}$$

$$\text{Margin} = \frac{2}{\|\vec{w}\|}$$

Linear SVM

- Linear model:

$$f(\vec{x}) = \begin{cases} 1 & \text{if } \vec{w} \bullet \vec{x} + b \geq 1 \\ -1 & \text{if } \vec{w} \bullet \vec{x} + b \leq -1 \end{cases}$$

- Learning the model is equivalent to determining the values of $\vec{w}$ and b
  - How to find $\vec{w}$ and b from training data?

Learning Linear SVM

- Objective is to maximize: $\text{Margin} = \frac{2}{\|\vec{w}\|}$
  - Which is equivalent to minimizing: $L(\vec{w}) = \frac{\|\vec{w}\|^2}{2}$
  - Subject to the following constraints:

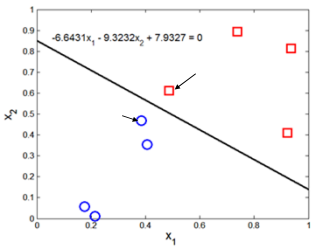
$$y_i = \begin{cases} 1 & \text{if } \vec{w} \bullet \vec{x}_i + b \geq 1 \\ -1 & \text{if } \vec{w} \bullet \vec{x}_i + b \leq -1 \end{cases}$$

or

$$y_i(\mathbf{w} \bullet \mathbf{x}_i + b) \geq 1, \quad i = 1,2,...,N$$ **(training data size)**

- This is a constrained optimization problem
  - Solve it using Lagrange multiplier method

Example of Linear SVM



Support vectors

x1	x2	y	λ
0.3858	0.4687	1	65.5261
0.4871	0.611	-1	65.5261
0.9218	0.4103	-1	0
0.7382	0.8936	-1	0
0.1763	0.0579	1	0
0.4057	0.3529	1	0
0.9355	0.8132	-1	0
0.2146	0.0099	1	0

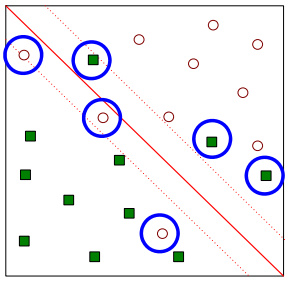
Learning Linear SVM

- Decision boundary depends only on support vectors
  - If you have data set with same support vectors, decision boundary will not change
  - How to classify using SVM once **w** and *b* are found? Given a test record, x_i

$$f(\vec{x}_i) = \begin{cases} 1 & \text{if } \vec{w} \bullet \vec{x}_i + b \geq 1 \\ -1 & \text{if } \vec{w} \bullet \vec{x}_i + b \leq -1 \end{cases}$$

Support Vector Machines

- What if the problem is not linearly separable?



Support Vector Machines

- What if the problem is not linearly separable?
  - Introduce slack variables
    - Need to minimize:

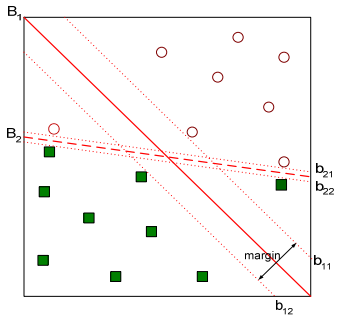
$$L(w) = \frac{\|\vec{w}\|^2}{2} + C\left(\sum_{i=1}^{N} \xi_i^k\right)$$

    - Subject to:

$$y_i = \begin{cases} 1 & \text{if } \vec{w} \bullet \vec{x}_i + b \geq 1 - \xi_i \\ -1 & \text{if } \vec{w} \bullet \vec{x}_i + b \leq -1 + \xi_i \end{cases}$$

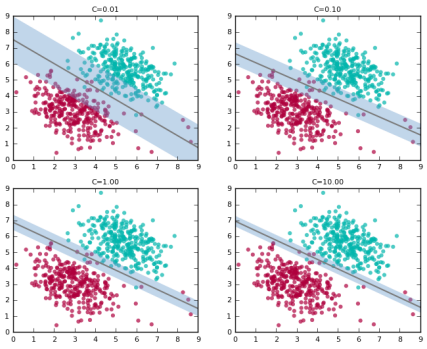
    - If k is 1 or 2, this leads to same objective function as linear SVM but with different constraints (see textbook)

Support Vector Machines



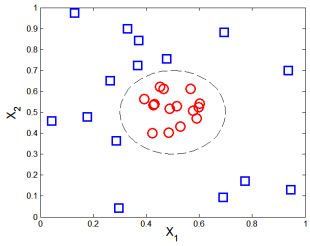
- Find the hyperplane that optimizes both factors

SVM classifier



Nonlinear Support Vector Machines

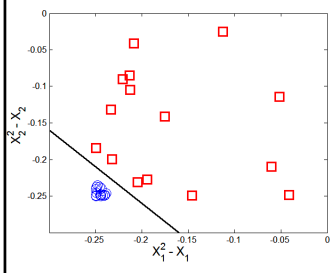
- What if decision boundary is not linear?



$$y(x_1, x_2) = \begin{cases} 1 & \text{if } \sqrt{(x_1-0.5)^2+(x_2-0.5)^2} > 0.2 \\ -1 & \text{otherwise} \end{cases}$$

Nonlinear Support Vector Machines

- Trick: Transform data into higher dimensional space



$$x_1^2 - x_1 + x_2^2 - x_2 = -0.46.$$

$$\Phi : (x_1, x_2) \longrightarrow (x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, 1).$$

$$w_4 x_1^2 + w_3 x_2^2 + w_2\sqrt{2}x_1 + w_1\sqrt{2}x_2 + w_0 = 0.$$

Decision boundary:

$$\vec{w} \bullet \Phi(\vec{x}) + b = 0$$

Learning Nonlinear SVM

- Optimization problem:

$$\min_{w} \frac{\|\mathbf{w}\|^2}{2}$$
$$subject\ to \quad y_i(\boldsymbol{w} \cdot \Phi(\boldsymbol{x}_i) + b) \geq 1,\ \forall\{(\boldsymbol{x}_i, y_i)\}$$

- Which leads to the same set of equations (but involve Φ(x) instead of x)

$$L_D = \sum_{i=1}^{n} \lambda_i - \frac{1}{2}\sum_{i,j} \lambda_i \lambda_j y_i y_j \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) \qquad \mathbf{w} = \sum_i \lambda_i y_i \Phi(\mathbf{x}_i)$$

$$\lambda_i\{y_i(\sum_j \lambda_j y_j \Phi(\mathbf{x}_j) \cdot \Phi(\mathbf{x}_i) + b) - 1\} = 0,$$

$$f(\mathbf{z}) = sign(\mathbf{w} \cdot \Phi(\mathbf{z}) + b) = sign(\sum_{i=1}^{n} \lambda_i y_i \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{z}) + b).$$

Learning NonLinear SVM

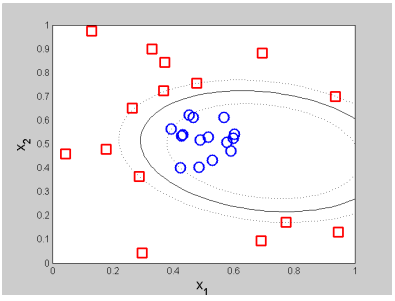
- Issues:
  - What type of mapping function Φ should be used?
  - How to do the computation in high dimensional space?
    - Most computations involve dot product $\Phi(x_i) \bullet \Phi(x_j)$
    - Curse of dimensionality?

Learning Nonlinear SVM

- Kernel Trick:
  - $\Phi(x_i) \bullet \Phi(x_j) = K(x_i, x_j)$
  - $K(x_i, x_j)$ is a kernel function (expressed in terms of the coordinates in the original space)
    - Examples:

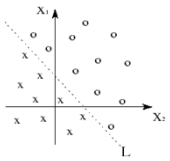
$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^p$$
$$K(\mathbf{x}, \mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|^2/(2\sigma^2)}$$
$$K(\mathbf{x}, \mathbf{y}) = \tanh(k\mathbf{x} \cdot \mathbf{y} - \delta)$$

Example of Nonlinear SVM



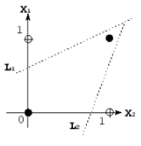
SVM with polynomial degree 2 kernel

Linear and not linearly separable (XOR)

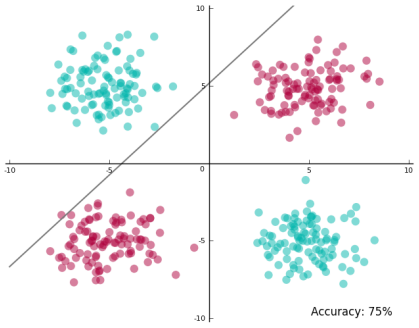


X_1	X_2	Y
0	0	0
0	1	1
1	0	1
1	1	0

$Y = X_1 \oplus X_2$



Non-linearly separable data (variant of XOR data set)



SVM for Not linearly separable data

- SVM's are really good at finding hyperplanes!
- Here is data that is not linearly separable
- Solution: If we can project data into a space where it is linearly separable, we can find a hyperplane in that space and map it back.
- When mapped back to original space, the separating boundary is not a line anymore

Projection

- Project the data into a space where it *is* linearly separable and find a hyperplane in this space!

$$X_1 = x_1^2$$
$$X_2 = x_2^2$$
$$X_3 = \sqrt{2}x_1x_2$$

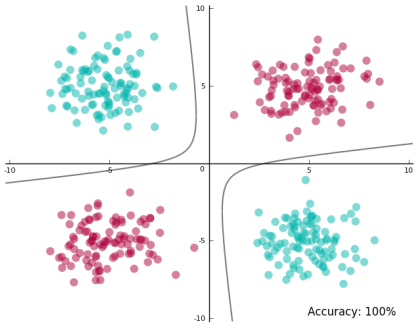
- Project the data set into a three-dimensional space using the above mapping
- Click-on-this

https://miro.medium.com/max/725/0*Ojchw_Exefs4qiok.

SVM for Not linearly separable data

- SVM's to do the projection for you.
- SVMs use something called *kernels* to do these projections, and these are fast (as it involves the computation of a few dot products)
- A kernel, short for *kernel function*, takes as input two points in the original space, and directly gives the dot product in the projected space.
- For point $\vec{x_i} = (x_{i1}, x_{i2})$ the projected point is $\vec{X_i} = (x_{i1}^2, x_{i2}^2, \sqrt{2}x_{i1}x_{i2})$
- The dot product in the 3 dimension needs 3 products
- SVM libraries come with pre-packaged popular kernels, such as polynomial, radical basis function (RBF), and sigmoid.

After mapping back

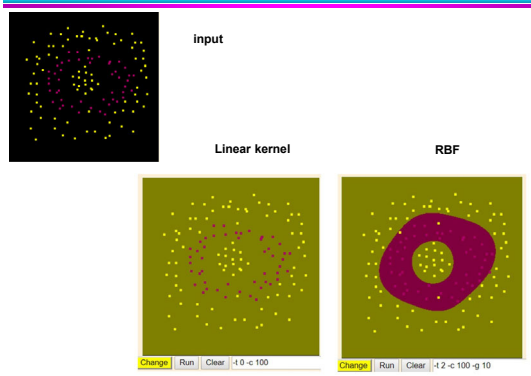


The shape of the boundary In the original space depends on the projection.

In the projected space, it Is always a hyperplane (4 support vectors)

The goal of projection was to use SVM's capability to find a hyperplane!

Another example



Learning Nonlinear SVM

- Advantages of using kernel:
  - Don't have to know the mapping function Φ
  - Computing dot product $\Phi(x_i) \bullet \Phi(x_j)$ in the original space avoids curse of dimensionality
- Not all functions can be kernels
  - Must make sure there is a corresponding Φ in some high-dimensional space
  - Mercer's theorem (see textbook)

Advantages of SVMs

- High-Dimensionality - The SVM is an effective tool in high-dimensional spaces, which is particularly applicable to document classification and sentiment analysis where the dimensionality can be extremely large (≥106).
- Memory Efficiency - Since only a subset of the training points are used in the actual decision process of assigning new members, only these points need to be stored in memory (and calculated upon) when making decisions.
- Versatility - Class separation is often highly non-linear. The ability to apply new kernels allows substantial flexibility for the decision boundaries, leading to greater classification performance.

Disadvantages of SVMs

- P > n - In situations where the number of features for each object (p) exceeds the number of training data samples (n), SVMs can perform poorly. This can be seen intuitively, as if the high-dimensional feature space is much larger than the samples, then there are less effective support vectors on which to support the optimal linear hyperplanes, leading to poorer classification performance as new unseen samples are added.
- Non-Probabilistic - Since the classifier works by placing objects above and below a classifying hyperplane, there is no direct probabilistic interpretation for group membership. However, one potential metric to determine "effectiveness" of the classification is how far from the decision boundary the new point is.