Foundations of Data Science & Analytics: Support Vector Machines

Ezgi Siir Kibris

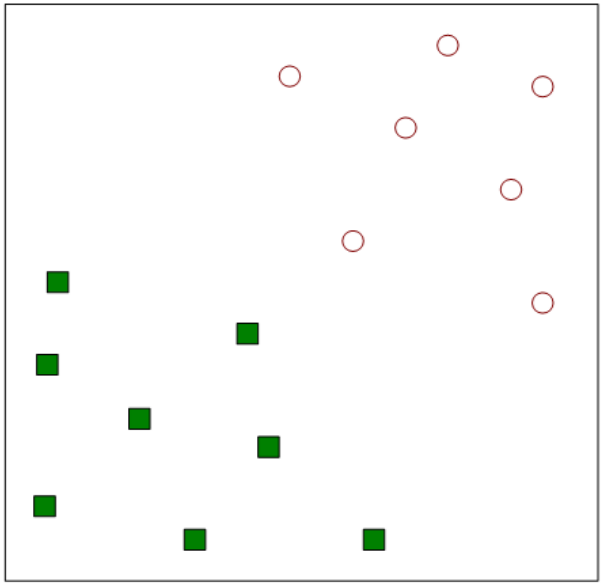
Introduction to Data Mining, 2nd Edition
by
Tan, Steinbach, Karpatne, Kumar

Classification Techniques

- **Base Classifiers**
  - Decision Tree based Methods
  - Rule-based Methods
  - Instance-based Methods (Nearest-neighbor)
  - Naïve Bayes
  - **Support Vector Machines**
  - Neural Networks and Deep Learning

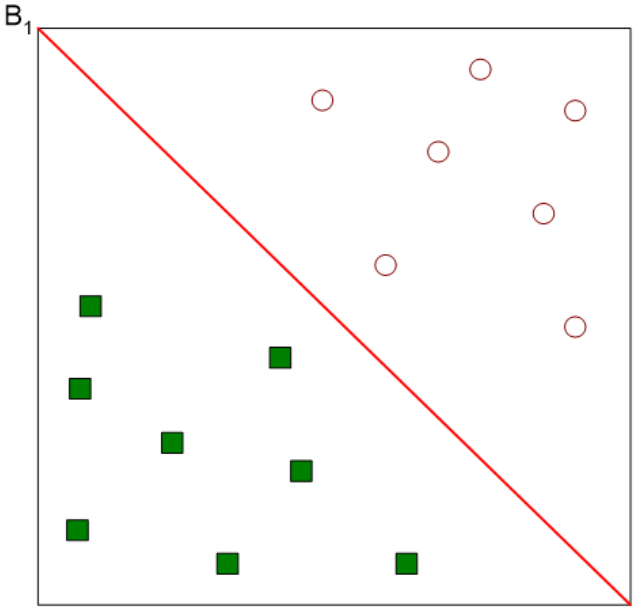
- **Ensemble Classifiers**
  - Boosting, Bagging, Random Forests

SVM



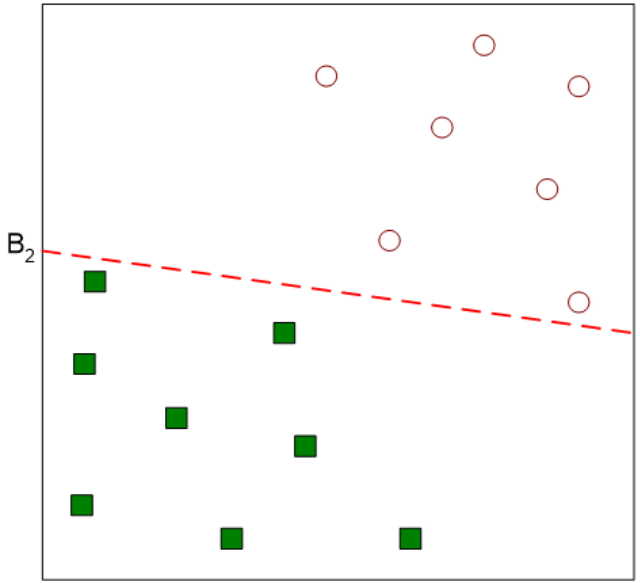
- Find a linear hyperplane (decision boundary) that will separate the data

SVM



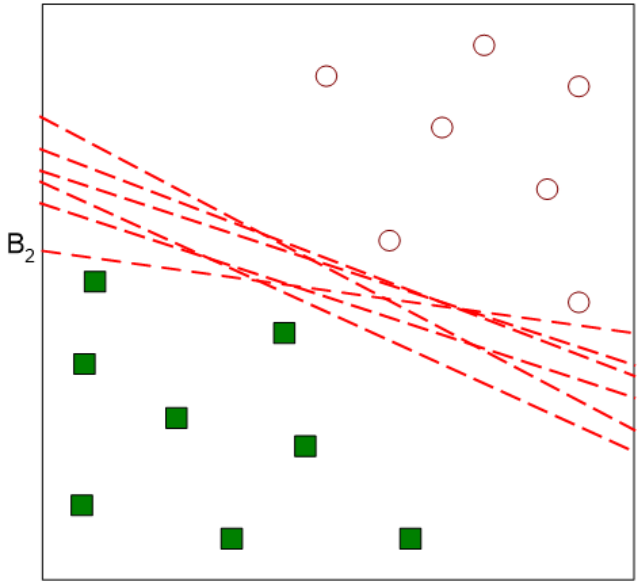
- One Possible Solution

SVM



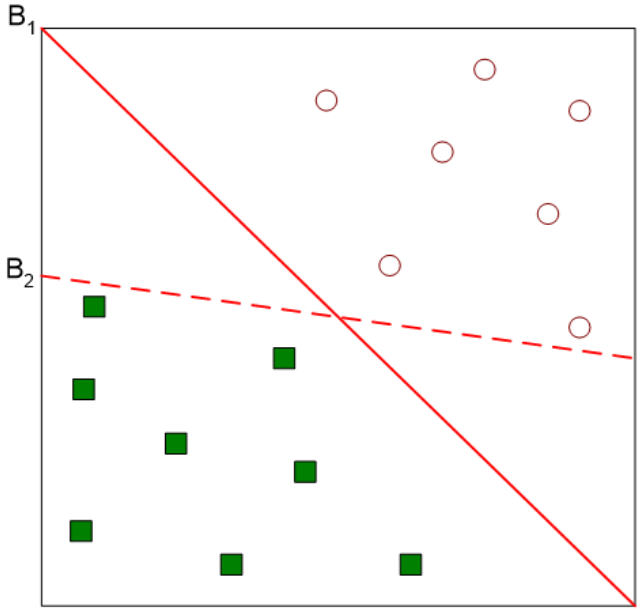
- Another possible solution

SVM



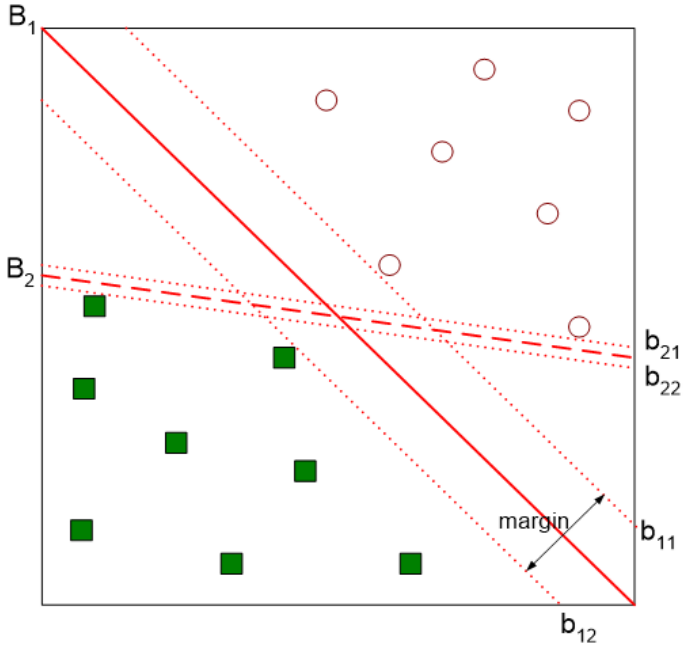
- Other possible solutions

SVM



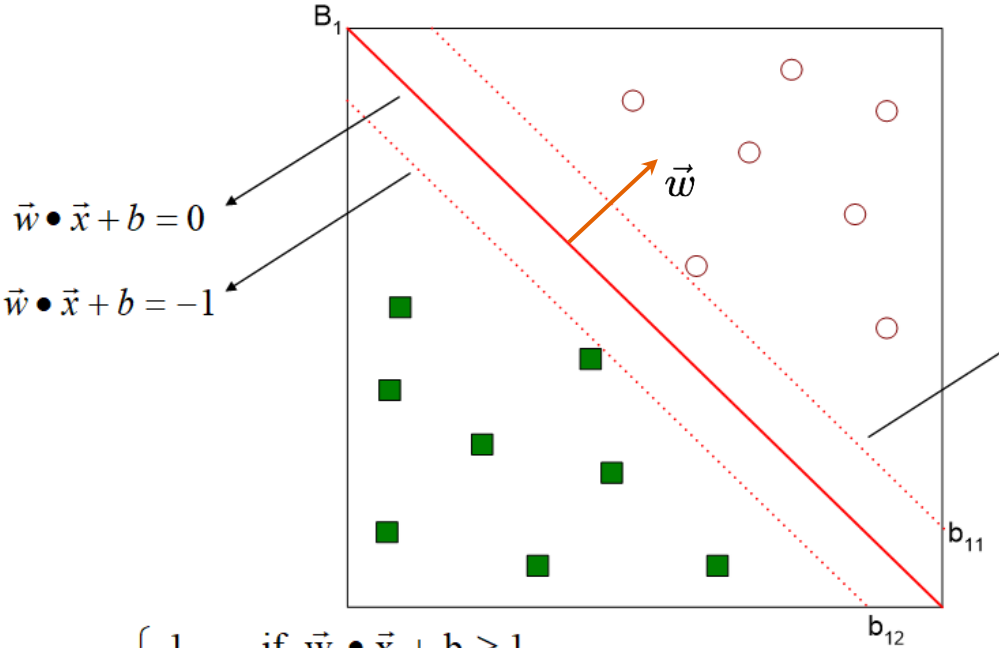
- Which one is better? B1 or B2?
- How do you define better?

SVM



- Find hyperplane maximizes the margin => B1 is better than B2

SVM

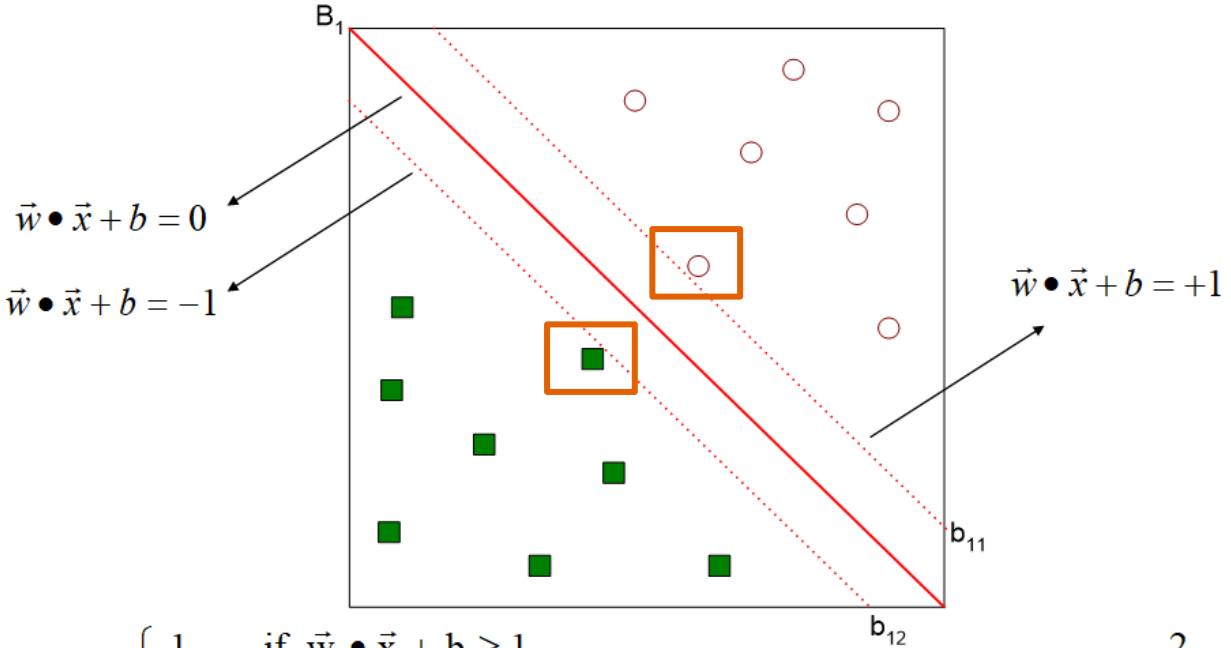


$$\vec{w} \cdot \vec{x}_1 + b = 1$$
$$\vec{w} \cdot \vec{x}_2 + b = -1$$
$$\vec{w} \cdot (\vec{x}_1 - \vec{x}_2) = 2$$
$$Margin = (\vec{x}_1 - \vec{x}_2) \cdot \frac{\vec{w}}{||\vec{w}||}$$

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

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

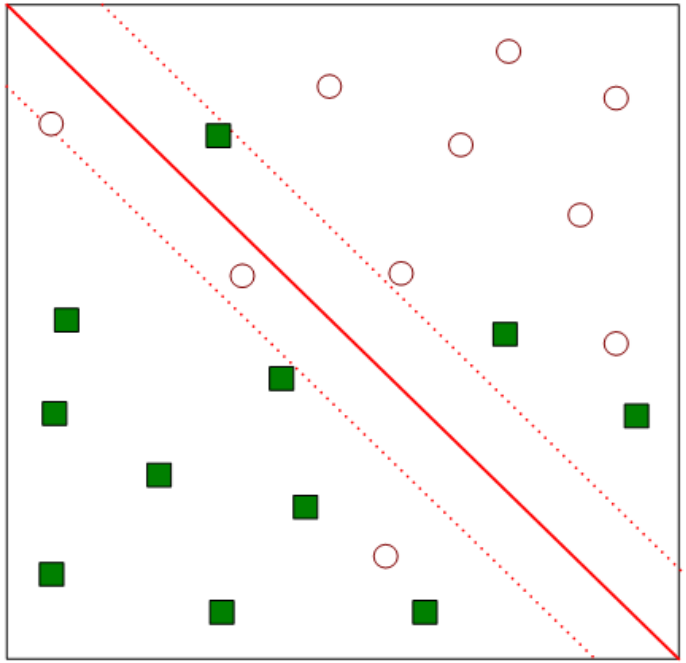
Support Vectors



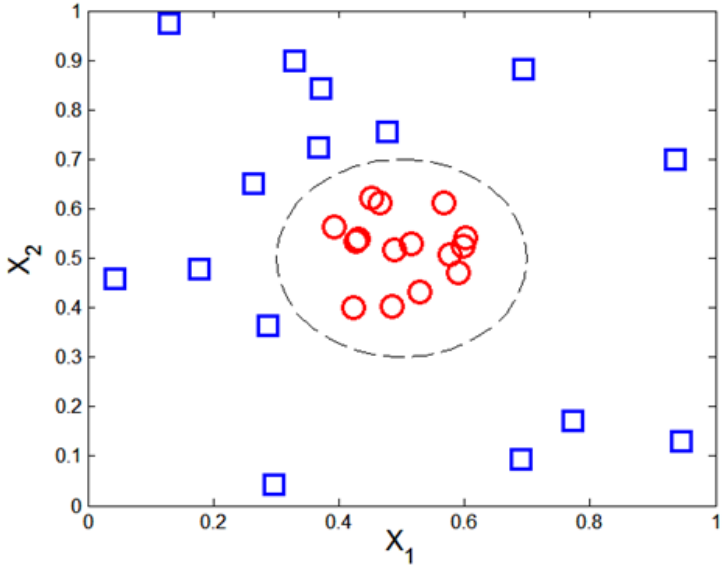
$$f(\vec{x}) = \begin{cases} 1 & \text{if } \vec{w} \bullet \vec{x} + b \geq 1 \\ -1 & \text{if } \vec{w} \bullet \vec{x} + b \leq -1 \end{cases}$$

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

Not linearly separable?

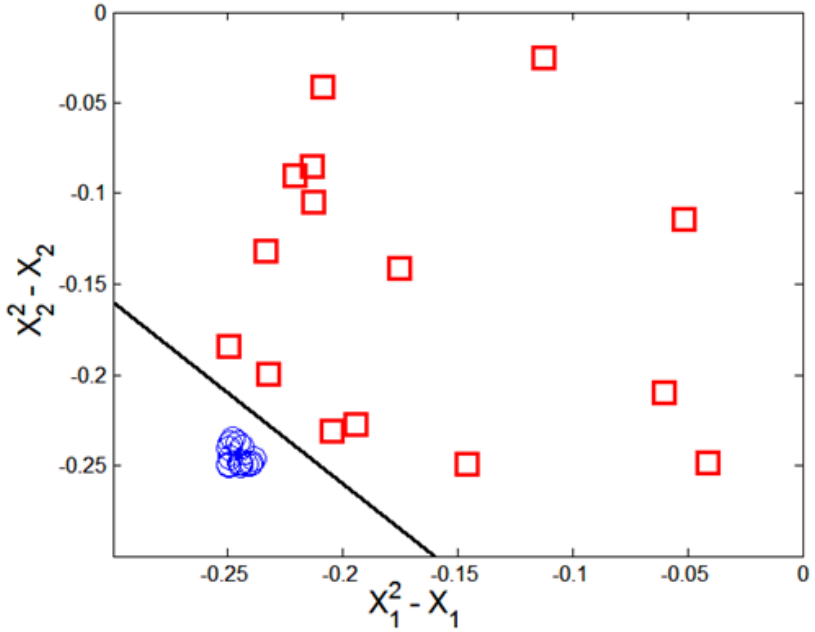


What if decision boundary is not linear?



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

Transform data into higher dimensional space – "Kernel Trick"



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

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

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

Decision boundary:

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

SVM

- Robust to noise
- Overfitting is handled by maximizing the margin of the decision boundary
- SVM can handle irrelevant and redundant data better than many other techniques
- The user needs to provide the type of kernel function and cost function
- High computational complexity for building the model
- Difficult to handle missing values
- What about categorical features?

Implementations on sklearn

- Linear SVM (soft-margin)
  - sklearn.svm.SVC (kernel=linear): based on libsvm.
  - sklearn.svm.LinearSVC: based on liblinear, more efficient.
  - sklearn.linear_model.SGDClassifier (loss=hinge): allows mini-batch training (suitable for online training when model can be incrementally updated with incoming training data), also efficient on large data.

- Nonlinear SVM
  - sklearn.svm.SVC (kernel != linear): based on libsvm.

Further Reading

http://web.mit.edu/6.034/wwwbob/svm-notes-long-08.pdf