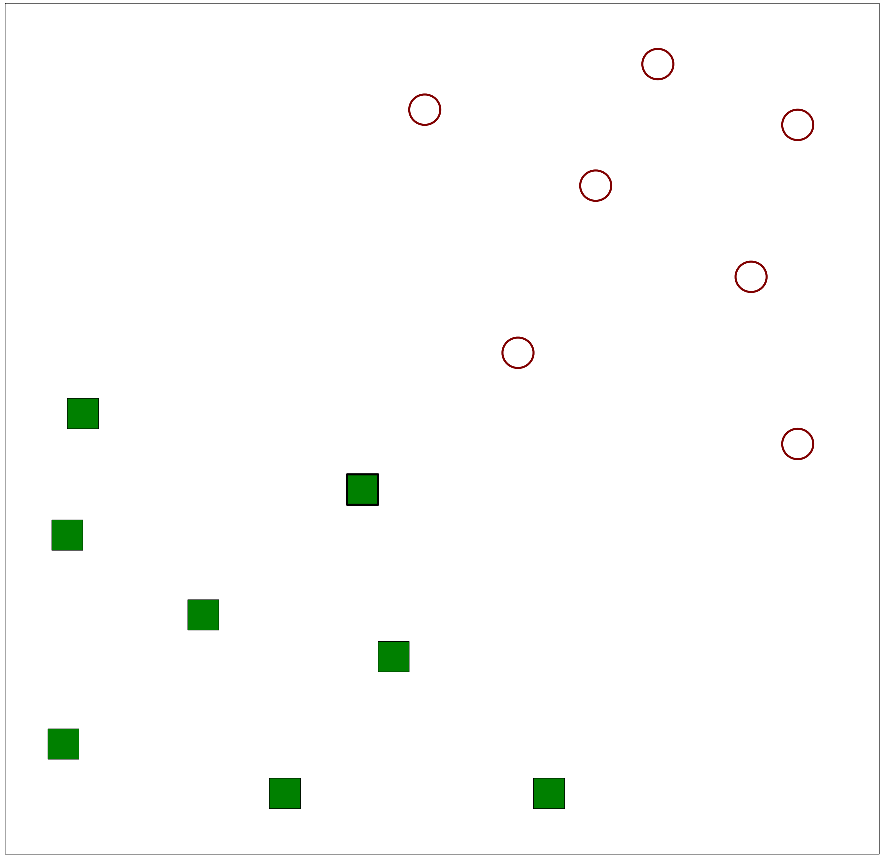
Support vector machine (SVM)

Support Vector Machine Overview

- Developed at AT&T Bell Laboratories by Vapnik with colleagues in 90's.
- SVMs are one of the most robust prediction methods
- SVMs can perform both linear and non-linear classification

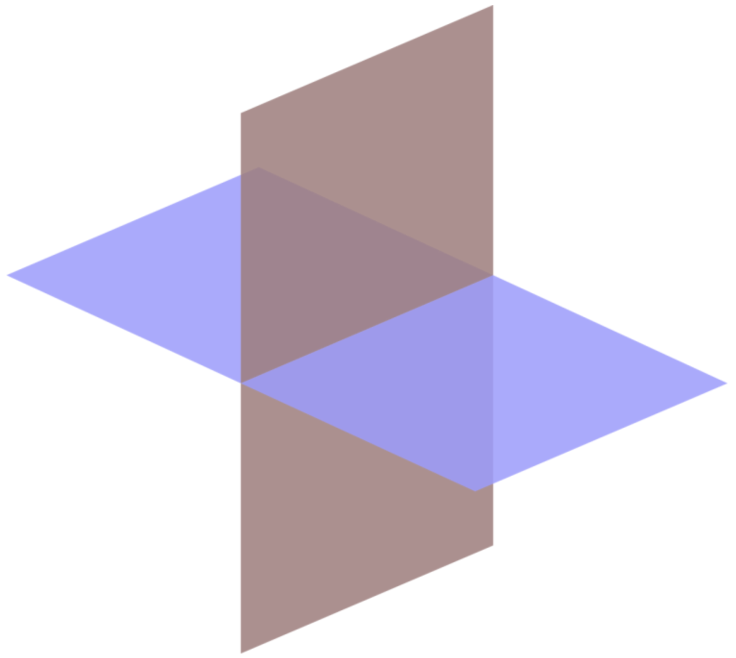
Support Vector Machine



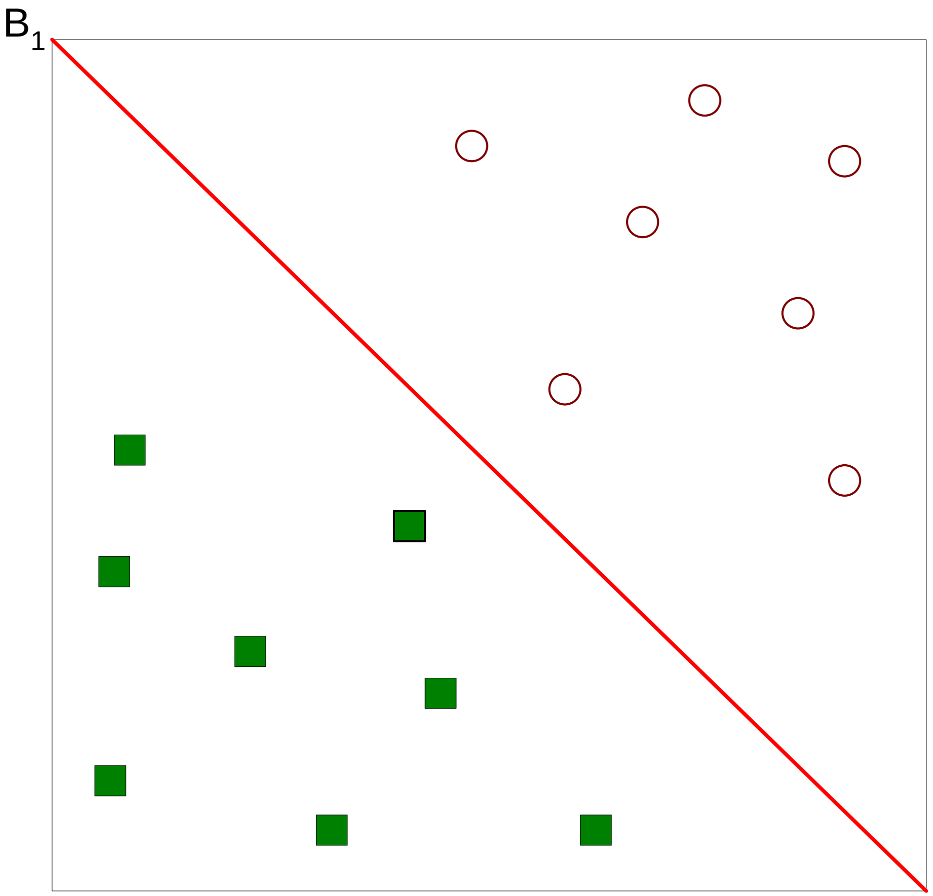
- Find a linear hyperplane (decision boundary) that will separate the data

Hyperplane

- In a p-dimensional feature space, a *hyperplane* is a flat affine subspace of hyperplane dimension $p - 1$.

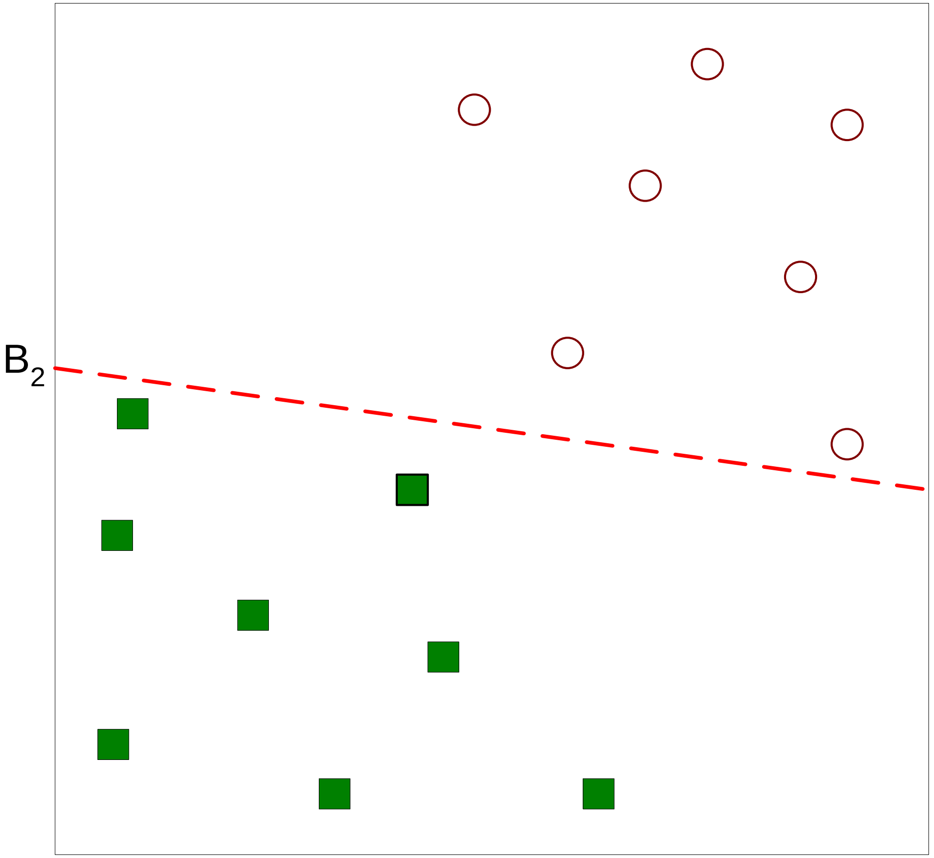


Support Vector Machine



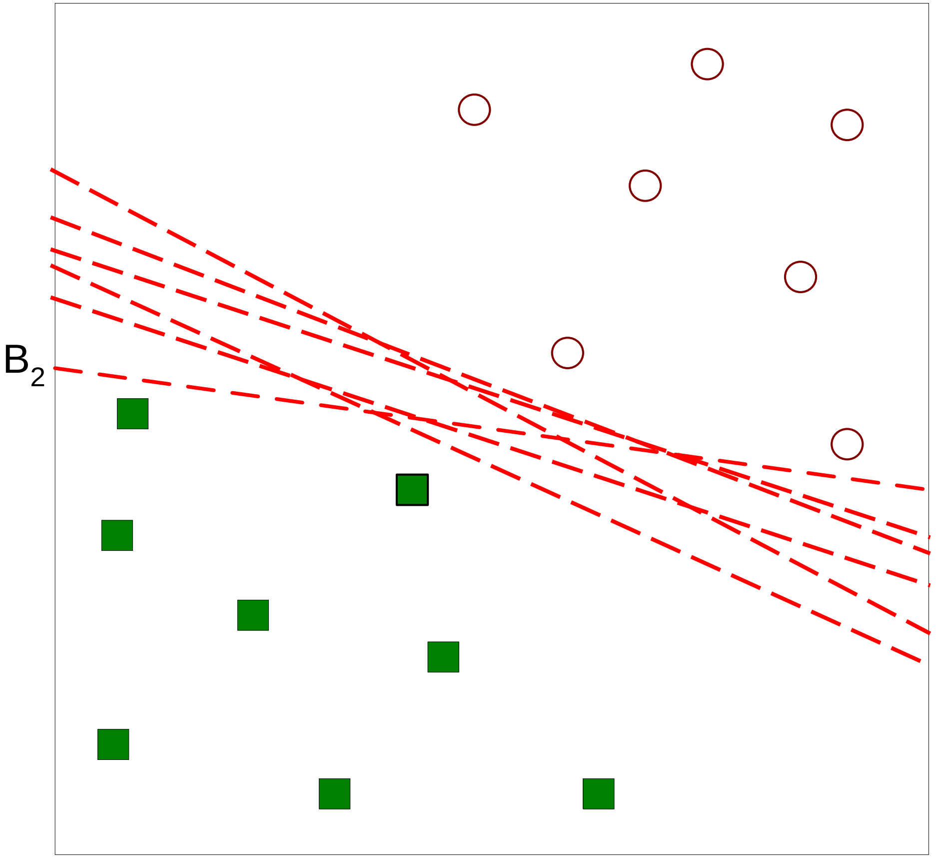
- One Possible Solution

Support Vector Machine



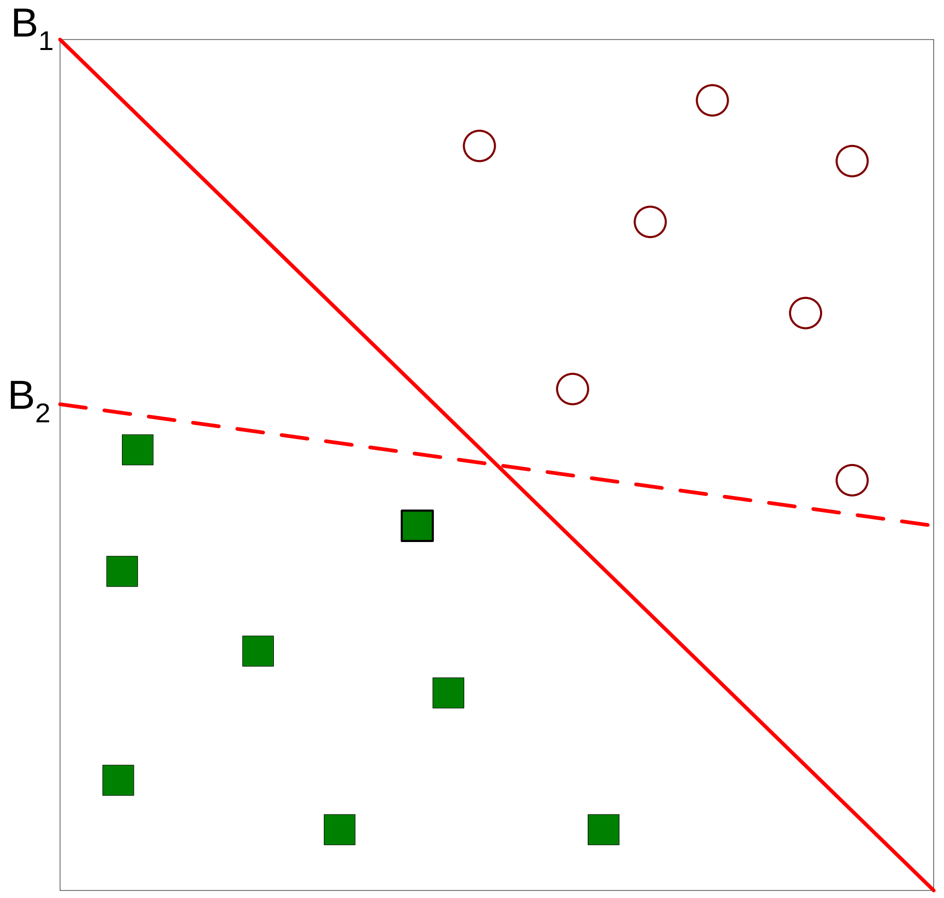
- Another possible solution

Support Vector Machine



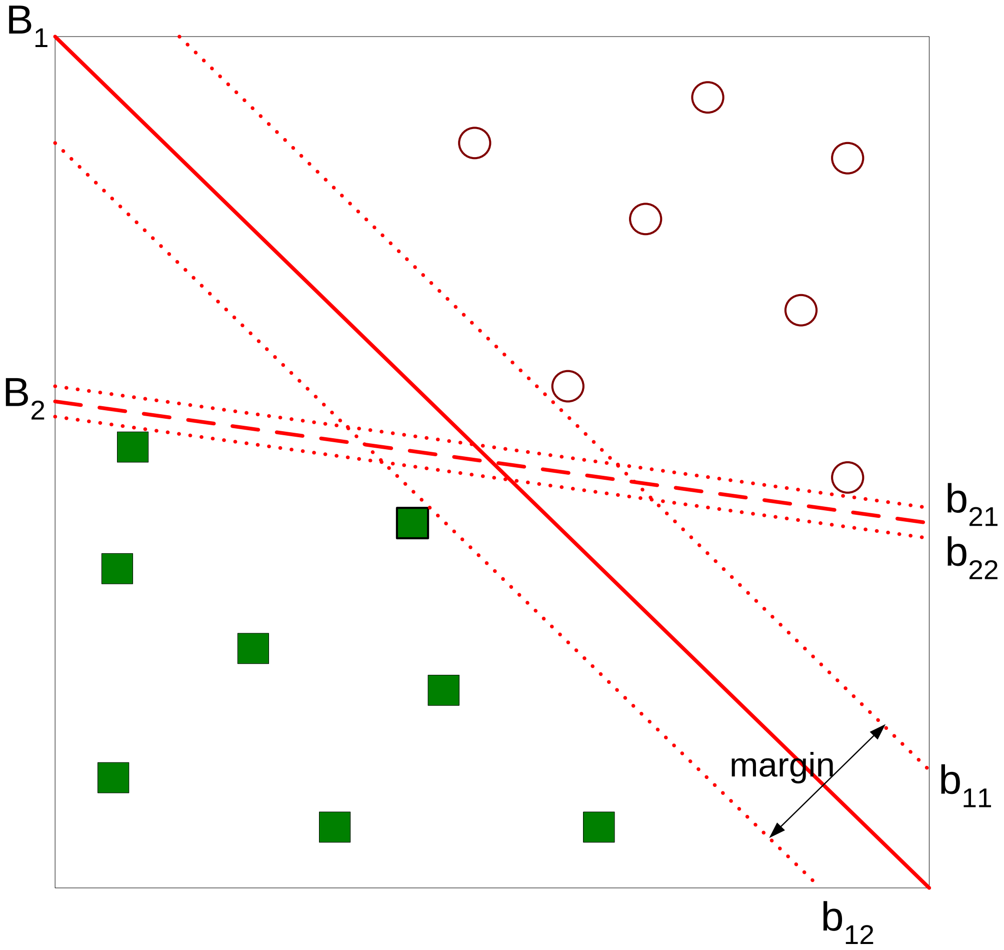
- Other possible solutions

Support Vector Machine



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

Support Vector Machine

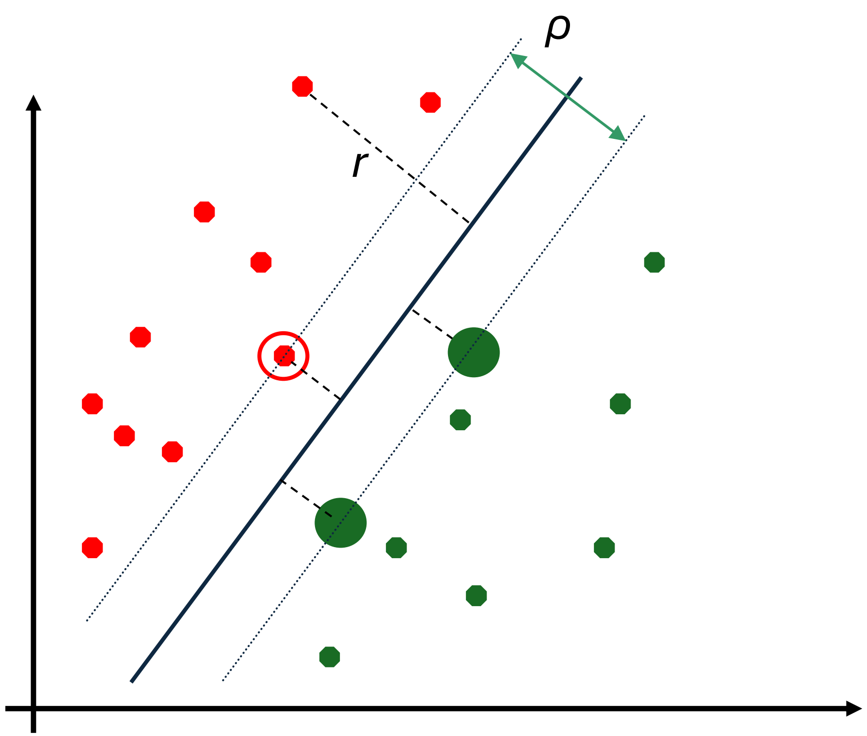


- Find hyperplane that maximizes the margin => B1 is better than B2

Notation

- $\mathbf{x}_i$: data point *i*, where $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{id})^T$
- y_i: class of data point *i* {+1, -1}
- Classifier is $f(\mathbf{x}_i) = \mathbf{w}^T\mathbf{x}_i + b$
- **w**: decision hyperplane normal vector
- Goal: find a hyperplane that correctly classify the data

A “Fat” Hyperplane



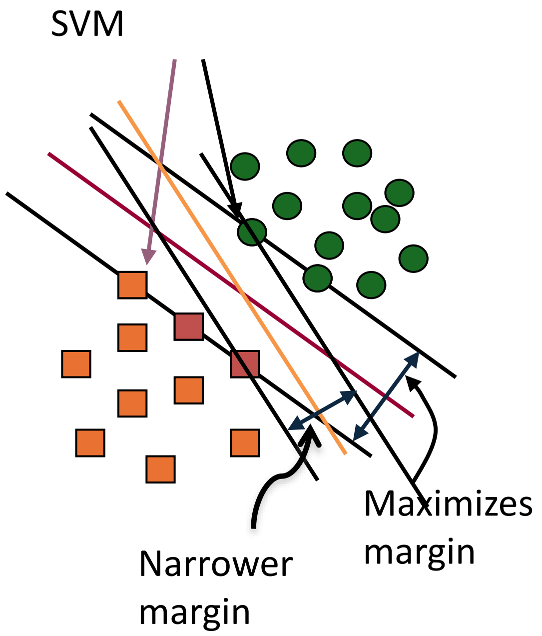
- Distance from example $\mathbf{x}_i$ to the separator is

$$r = y\frac{\mathbf{w}^T\mathbf{x}_i + b}{\|\mathbf{w}\|}$$

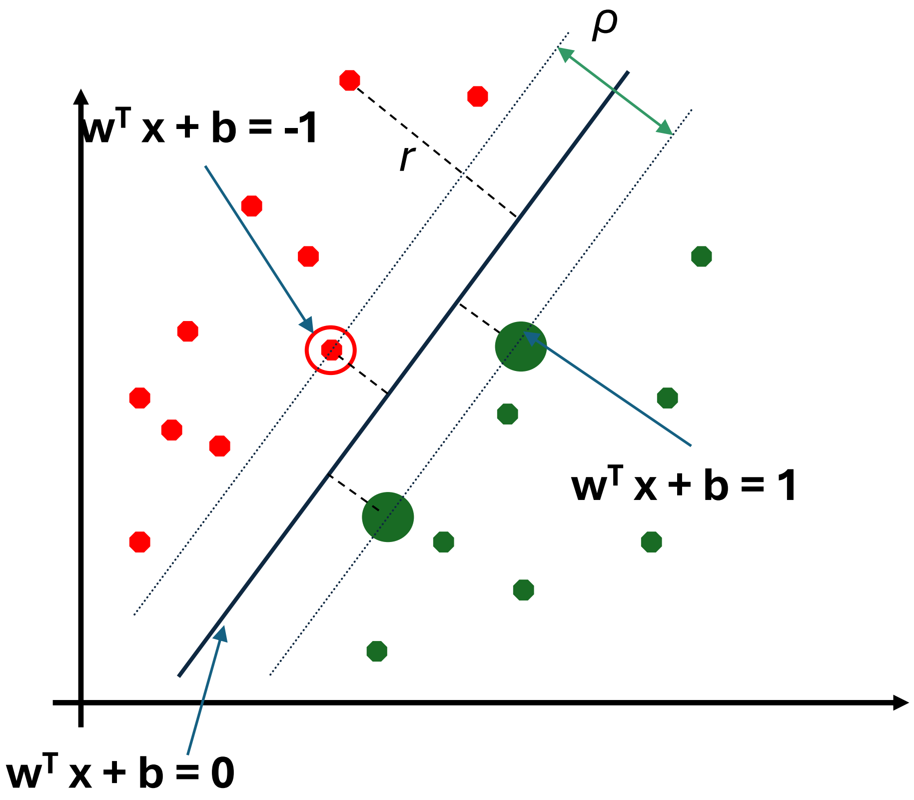
- ***Support vectors***: cases closest to the hyperplane
  - Difficult points
  - Uncertain decisions
- ***Margin*** ρ of the separator is the distance between support vectors.

SVM: A Maximum-Margin Hyperplane

- SVM maximizes the margin around the separating hyperplane
- The decision function is fully specified by a subset of training samples, *the support vectors*.
  - Other training examples are ignorable.
- Solving SVMs is a *quadratic programming* problem



The Linearly Separable Case



- Hyperplane: $\mathbf{w}^T \mathbf{x} + b = 0$
- Classifier: $\mathbf{w}^T\mathbf{x_i} + b \geq 1 \quad \text{if } y_i = 1$

  $\mathbf{w}^T\mathbf{x_i} + b \leq -1 \quad \text{if } y_i = -1$
  - For support vectors, the inequality becomes an equality
- Distance from example $\mathbf{x}_i$ to the separator is
  $$r = y\frac{\mathbf{w}^T\mathbf{x}_i + b}{\|\mathbf{w}\|}$$
- The margin is: $r = \frac{2}{\|\mathbf{w}\|}$

The Linearly Separable Case

- Objective is to find w and b such that:

  Margin $= \dfrac{2}{||\vec{w}||}$ is maximized

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

$$y_i = \begin{cases} 1, \ if\ \vec{w} \cdot \overrightarrow{x_1} + b \geq 1 \\ -1, if\ \vec{w} \cdot \overrightarrow{x_1} + b \leq -1 \end{cases}$$

  Or,

$$y_i(\vec{w} \bullet \overrightarrow{x_i} + b) \geq 1, \qquad i = 1,2, \ldots, N$$

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

Solving the Optimization Problem

- Need to optimize a *quadratic* function subject to *linear* constraints.
- Quadratic optimization problems are a well-known class of mathematical programming problems for which several (non-trivial) algorithms exist.
- The solution involves constructing a *dual problem* where a *Lagrange multiplier* λ_i is associated with every inequality constraint in the primal (original) problem

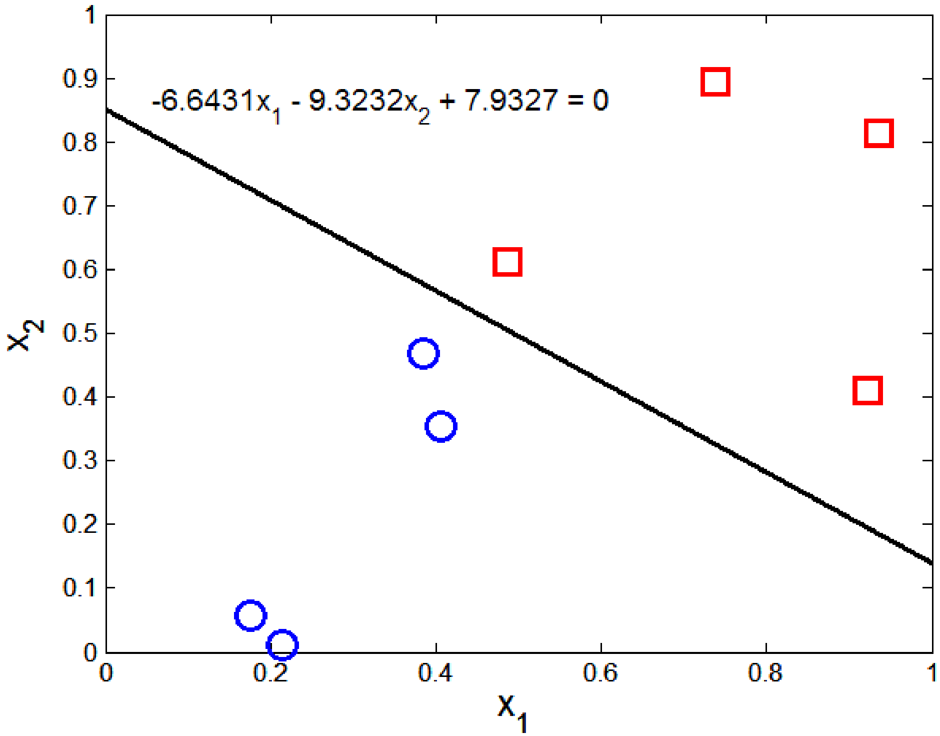
$$L_P = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{N} \lambda_i \Big(y_i(\mathbf{w} \cdot \mathbf{x_i} + b) - 1 \Big)$$

$$\frac{\partial L_p}{\partial \mathbf{w}} = 0 \Longrightarrow \mathbf{w} = \sum_{i=1}^{N} \lambda_i y_i \mathbf{x}_i,$$

$$\boxed{L_D = \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j \mathbf{x_i} \cdot \mathbf{x_j}.}$$

$$\frac{\partial L_p}{\partial b} = 0 \Longrightarrow \sum_{i=1}^{N} \lambda_i y_i = 0.$$

Example of Linear SVM

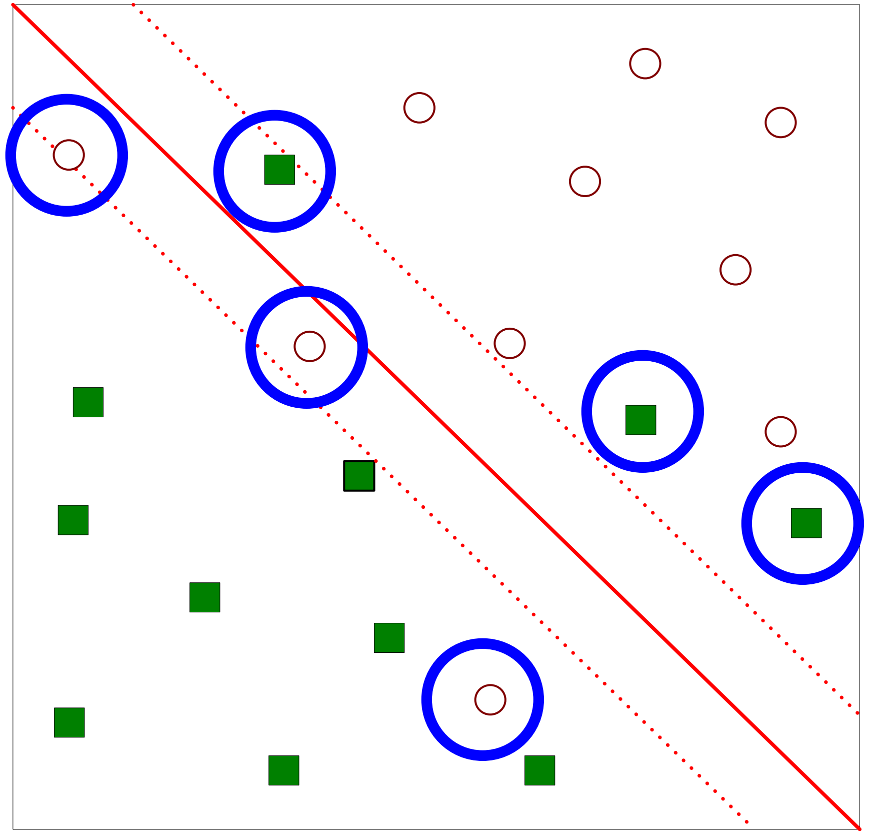


Support vectors

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

Support Vector Machines

- What if the problem is not linearly separable?



Soft Margin Approach

- 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:

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

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

Soft Margin Classification

- The old formulation:

Find $\mathbf{w}$ and b such that
$\mathbf{\Phi}(\mathbf{w}) = \mathbf{w}^T\mathbf{w}$ is minimized
and for all $(\mathbf{x}_i, y_i)$, i=1..n : $y_i (\mathbf{w}^T\mathbf{x}_i + b) \geq 1$

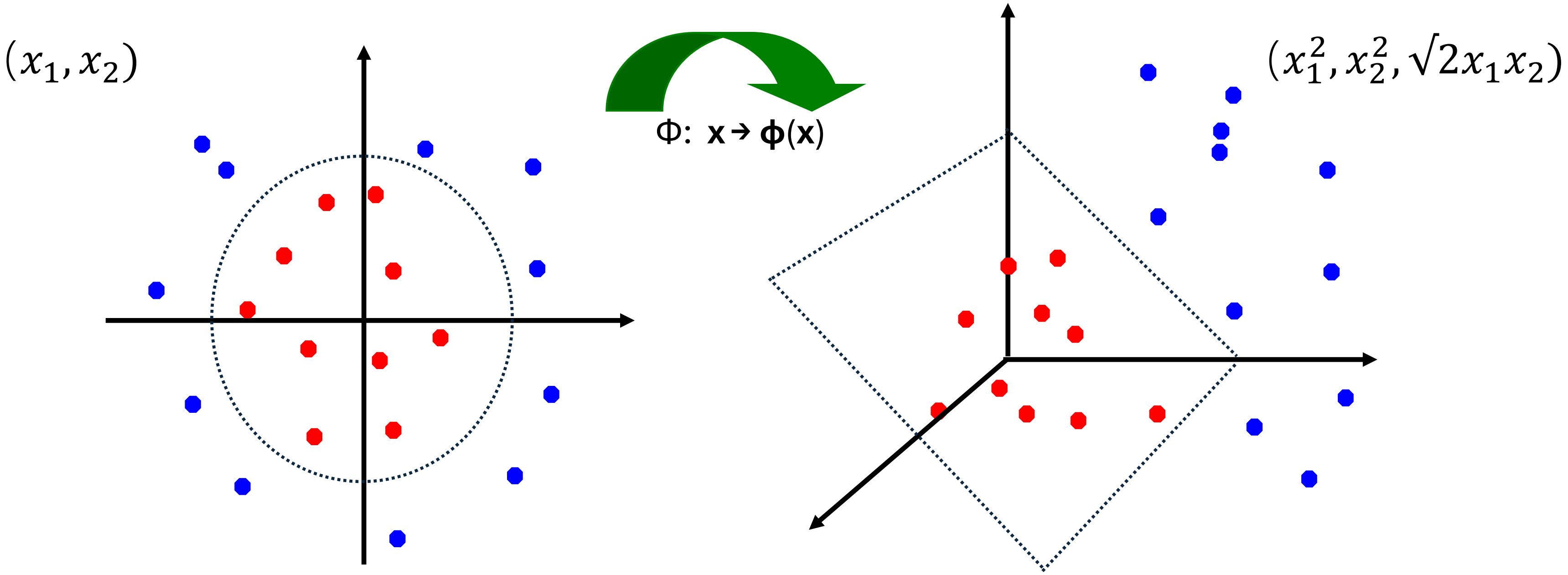
- Modified formulation incorporates slack variables:

Find $\mathbf{w}$ and b such that
$\mathbf{\Phi}(\mathbf{w}) = \mathbf{w}^T\mathbf{w} + C\Sigma\xi_i$ is minimized
and for all $(\mathbf{x}_i, y_i)$, i=1..n : $y_i (\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i$, $\xi_i \geq 0$

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

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:



The “Kernel Trick”

❖ The linear classifier relies on inner product between vectors
K$(x_i, x_j) = x_i \cdot x_j$

❖ If every data point is mapped into high-dimensional space via some transformation Φ: **x** → ϕ(**x**), the inner product becomes:

$$K(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j)$$

❖ A kernel function is a function that is equivalent to an inner product in some feature space.

❖ Examples of kernel functions:

- Linear, polynomial, Gaussian, Sigmoid

Kernels

- Why use kernels?
  - Do not have to know the exact form of the mapping function $\boldsymbol{\phi}(\mathbf{x})$
  - Computing using kernel functions is considerable cheaper
  - Avoid curse of dimensionality
- Common kernels
  - Linear: $K(\mathbf{x}_i,\mathbf{x}_j)= \mathbf{x}_i^T\mathbf{x}_j$
    - Mapping Φ: $\mathbf{x} \rightarrow \boldsymbol{\phi}(\mathbf{x})$, where $\boldsymbol{\phi}(\mathbf{x})$ is $\mathbf{x}$ itself
  - Polynomial of power p: $K(\mathbf{x}_i,\mathbf{x}_j)= (1+ \mathbf{x}_i^T\mathbf{x}_j)^p$
    - Mapping Φ: $\mathbf{x} \rightarrow \boldsymbol{\phi}(\mathbf{x})$, where $\boldsymbol{\phi}(\mathbf{x})$ has $\binom{d+p}{p}$ dimensions
  - Gaussian (radial-basis function): $K(\mathbf{x}_i,\mathbf{x}_j) = e^{-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{2\sigma^2}}$
    - Mapping Φ: $\mathbf{x} \rightarrow \boldsymbol{\phi}(\mathbf{x})$, where every point is mapped to a function (a Gaussian); combination of functions for support vectors is the separator.

SVM Summary

- The classifier is a *separating hyperplane*.
  - Maximum-margin in a feature space
  - The feature space is constructed by a kernel function

- Most “important” training points are support vectors; they define the hyperplane.
  - Support vector = “critical” point close to decision boundary

- Quadratic optimization algorithms can identify which training points xi are support vectors with non-zero Lagrangian multipliers λ_i

SVM Applications

- SVMs are currently among the best performers for several 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 several tasks such as regression [Vapnik *et al.* ’97], principal component analysis [Schölkopf *et al.* ’99], etc.
- Tuning SVMs remains a black art: selecting a specific kernel and parameters is usually done in a try-and-see manner.
  - Select Kernel function and related parameters
    - E.g., Gamma for a Gaussian Kernel
  - Select cost parameter, c, to control soft margin

Advantages of SVM

- Finds a global, unique minimum.
- The kernel trick.
- A simple geometric interpretation.
- Strong ability to generalize.
- Less sensitive to outliners
- The complexity of the calculations does not depend on the dimension of the input space
  - Avoids the curse of dimensionality

Disadvantages of SVM

- Which kernel function?
- How to select the parameters of the kernel function?