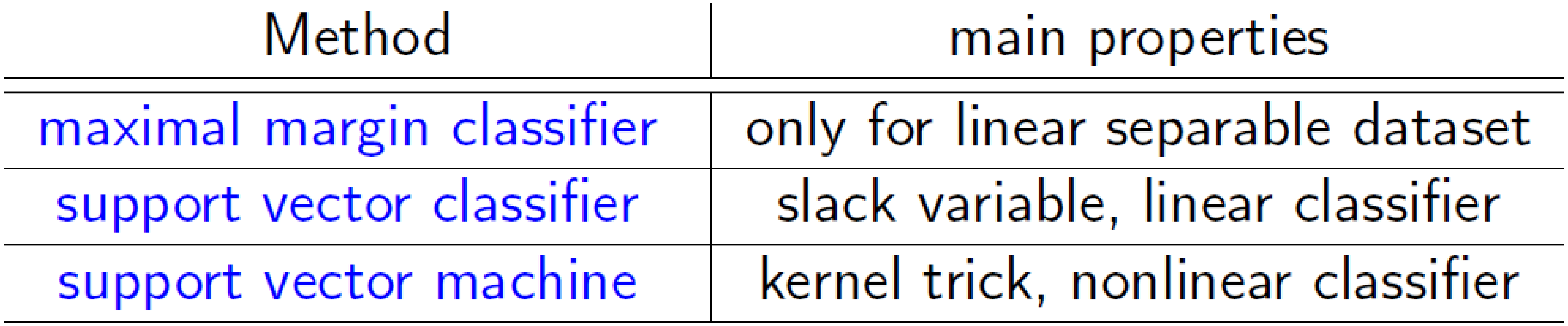
SDSC 5001: Statistical Machine Learning I

Topic 9. Support Vector Machine

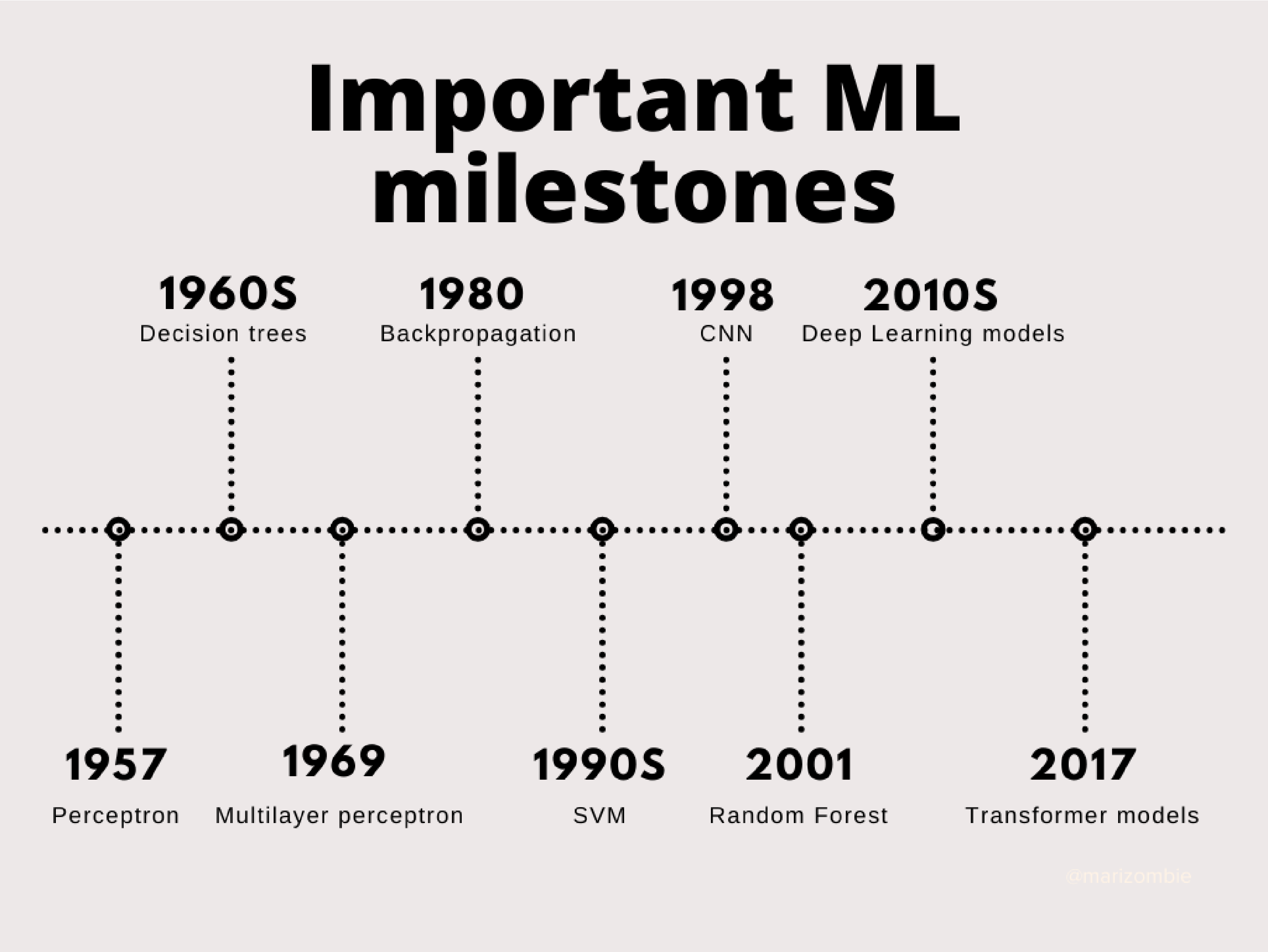
Support Vector Machine (SVM)

- Vapnik 1995: Geometric viewpoint + Primal-dual for quadratic programming (+ Kernel trick)
- Mainly developed by and dominantly hot in computer science/pattern recognition throughout 1990’s.
- One of the most important and successful developments before deep learning



Method	main properties
maximal margin classifier	only for linear separable dataset
support vector classifier	slack variable, linear classifier
support vector machine	kernel trick, nonlinear classifier

Machine Learning Milestones



Important ML milestones

1957
Perceptron

1960S
Decision trees

1969
Multilayer perceptron

1980
Backpropagation

1990S
SVM

1998
CNN

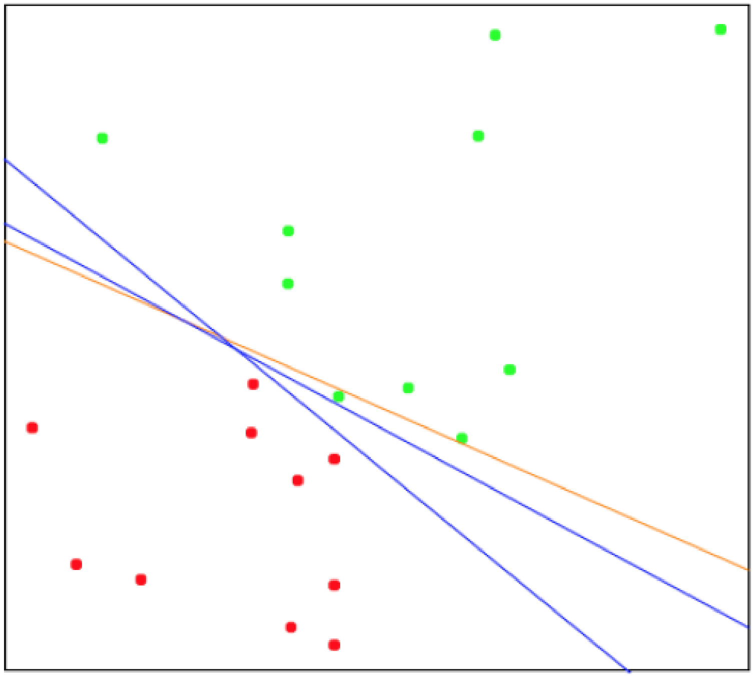
2001
Random Forest

2010S
Deep Learning models

2017
Transformer models

Idea of SVM

- We try and find a plane that separates the classes in feature space.
- If we cannot, we get creative in two ways: (1) soften what we mean by “separates”, and (2) enrich and enlarge the feature space so that separation is possible.



Review of Vector Algebra (Topic 6)

- A hyperplane L is defined by the equation:

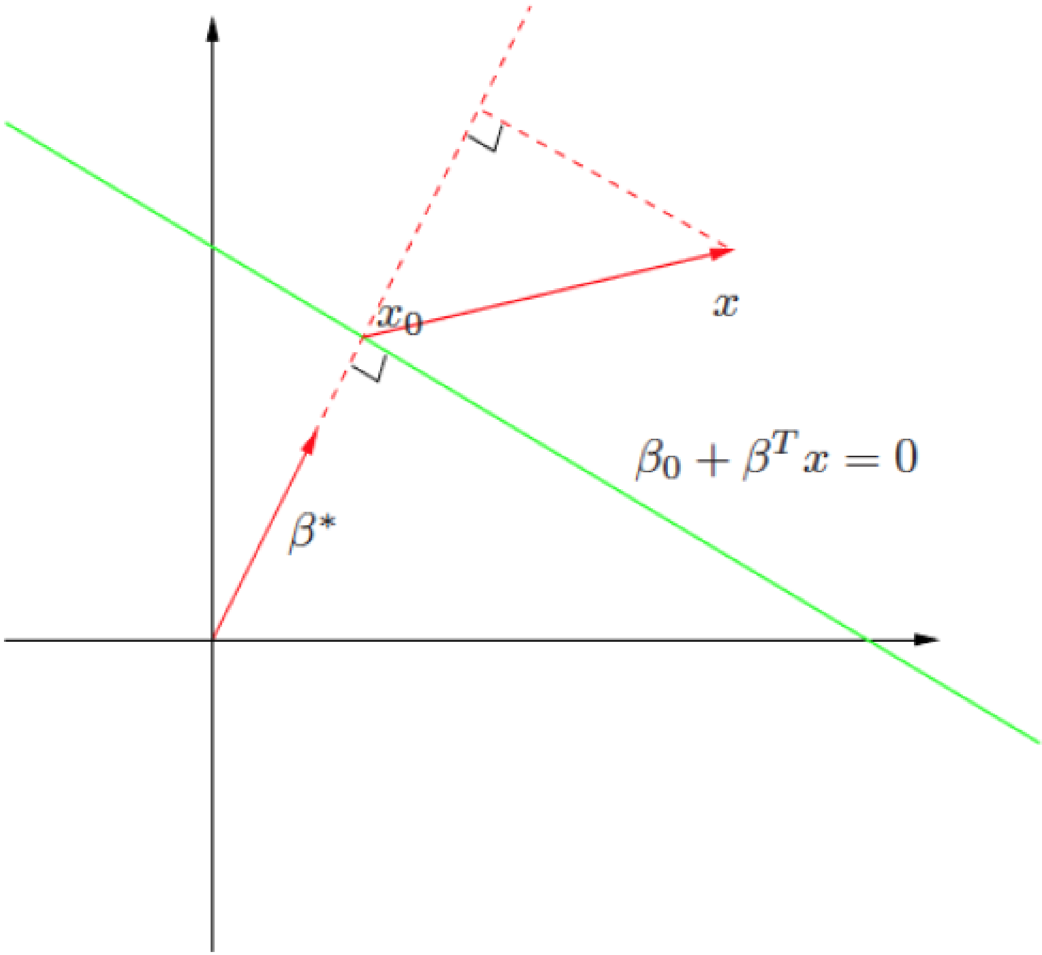
$$L = \{\mathbf{x}: f(\mathbf{x}) = \beta_0 + \mathbf{x}^T\beta = 0\}$$

- L separates the space into two parts: $\{\mathbf{x}: f(\mathbf{x}) > 0\}$ and $\{\mathbf{x}: f(\mathbf{x}) < 0\}$.

- Given a data point $(\mathbf{x}, y)$ with $y = \{-1, 1\}$, a correct classification implies that $yf(\mathbf{x}) > 0$.

- The signed distance of any point $(\mathbf{x}, y)$ to L is given by $\frac{1}{||\beta||} f(\mathbf{x})$.

Geometry of A Hyperplane



Preliminaries

- Training data: $(\mathbf{x}_i, y_i)$, $i = 1, \dots, n$ with $\mathbf{x}_i \in R^p$ and $y_i \in \{-1,1\}$.

- Define a separating hyperplane

$$L = \{\mathbf{x}: f(\mathbf{x}) = \beta_0 + \mathbf{x}^T\beta = 0\}$$

- A classification decision function is

$$G(\mathbf{x}) = \operatorname{sign}\big(f(\mathbf{x})\big) = \operatorname{sign}(\beta_0 + \mathbf{x}^T\beta)$$

Note: the notation of $y = \{-1, 1\}$ is convenient because

$$\operatorname{sign}\big(f(\mathbf{x})\big) = y \Leftrightarrow yf(\mathbf{x}) > 0$$

- A misclassified point $(\mathbf{x}, y) \Leftrightarrow yf(\mathbf{x}) < 0$, and vice versa.

Separable Case

- There exists a hyperplane L which linearly separates one class from the other without error. That is, for all data points $(\mathbf{x}_i, y_i)$, $i = 1, \dots, n$, the hyperplane satisfies that

$$y_i f(\mathbf{x}_i) > 0$$

- The **distance** of $(\mathbf{x}_i, y_i)$ to the separating hyperplane

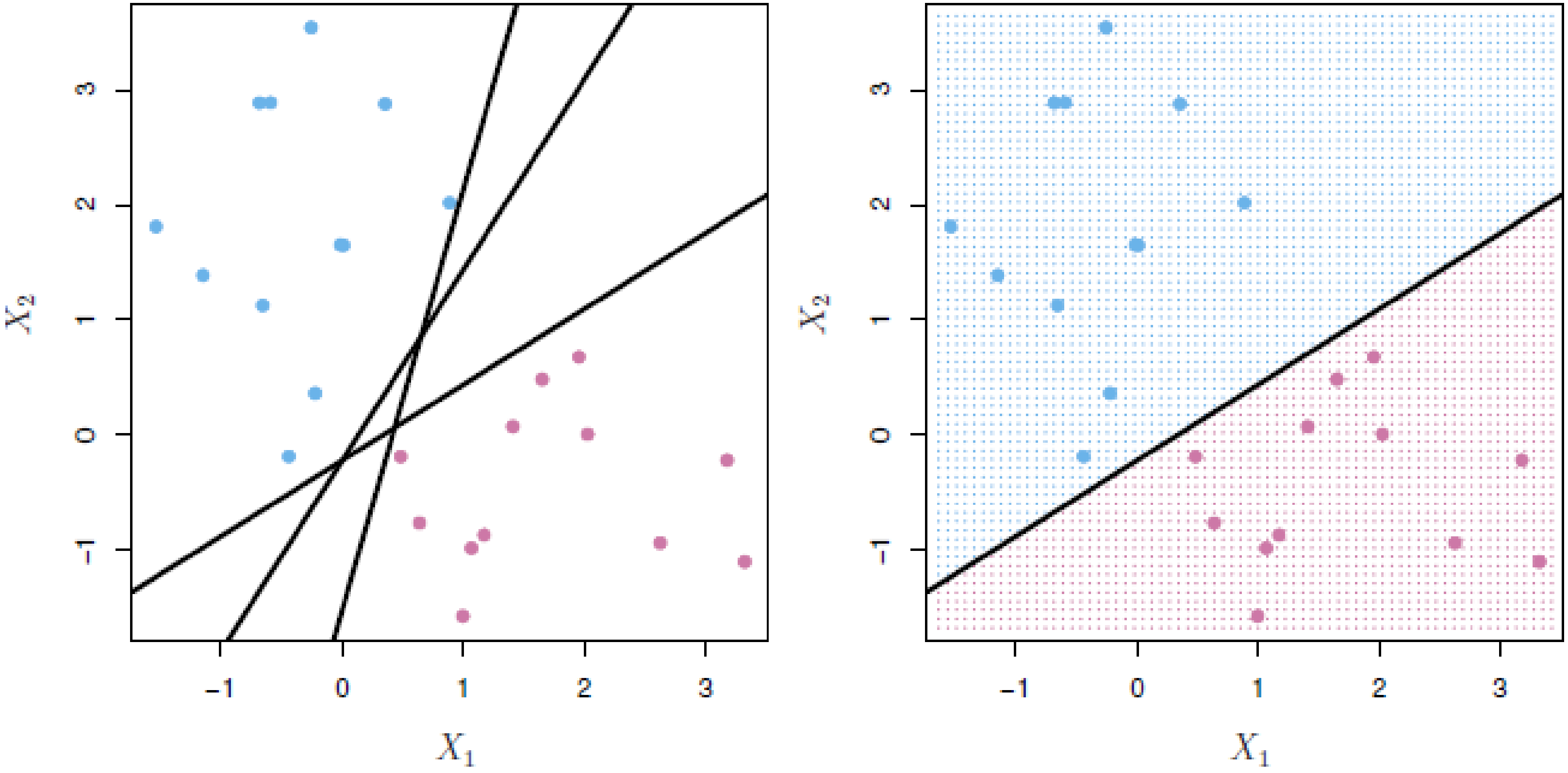
$$D_i = \frac{y_i f(\mathbf{x}_i)}{||\beta||}$$

- L separates the two classes $\Leftrightarrow$ $f(\mathbf{x})$ correctly classifies $(\mathbf{x}_i, y_i)$

$$\Leftrightarrow D_i > 0, i = 1, \dots, n$$

- A large value of D_i indicates a large distance to the decision boundary, i.e., large confidence of correctness in classification for $(\mathbf{x}_i, y_i)$.

Illustrative Plot: Separating Hyperplane



Maximal Margin Classifier

- Margin: the minimal D_i of all data points $(\mathbf{x}_i, y_i)$, $i = 1, \dots, n$

$$M = \min_{1 \le i \le n} D_i$$

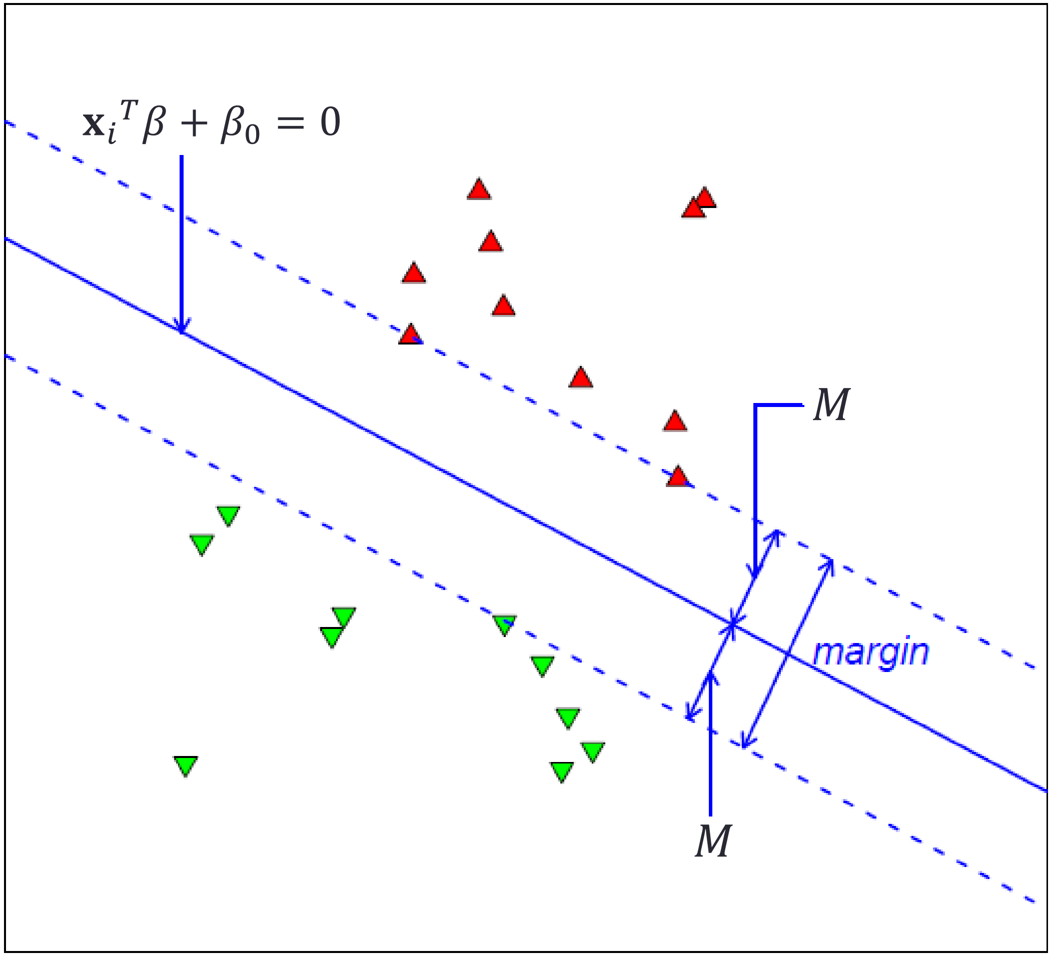
Note: Sometimes, the margin refers to $2M$.

- The margin depends on the parameters β_0, β. The optimal separating hyperplane is the one that has the maximal margin

$$M^* = \max_{\beta_0, \beta} \quad M$$

- If M^* is positive, the corresponding optimal β_0 and β gives the optimal linearly separating hyperplane, i.e., the dataset is linearly separable. If M^* is negative, there exists some data points misclassified, i.e., the dataset is not linearly separable.

Illustrative Plot: Maximal Margin Classifier



Support Vectors

- **Support vectors**: the training data points that lie on the margin and equidistant from the optimal separating hyperplane (maximal margin hyperplane)
  - "vectors": each data point is a vector in p-dimensional space
  - "support": they support the maximal margin hyperplane in that if they were moved slightly then the maximal margin hyperplane would move as well.

- The maximal margin hyperplane depends directly on the support vectors, but not on other training data points: a movement of any of those data points would not affect the hyperplane.

Optimization Formulation

- Suppose the margin is $2M$ units, then

$$\max_{\beta_0, \beta, \|\beta\|=1} \quad M$$

$$\text{subject to } y_i({\mathbf{x}_i}^T\beta + \beta_0) \geq M,\ i = 1, \dots, n$$

- The constraint $\|\beta\| = 1$ is only for the uniqueness of β_0 and β; without this constraint, the solution is a family up to a positive multiplication constant, which all share the same classifier since $\text{sign}\big(f(\mathbf{x})\big) = \text{sign}\big(\lambda f(\mathbf{x})\big)$.

- This is not a convex optimization problem, but it can be rephrased as a convex quadratic problem and solved efficiently.

Optimization Formulation (Cont.)

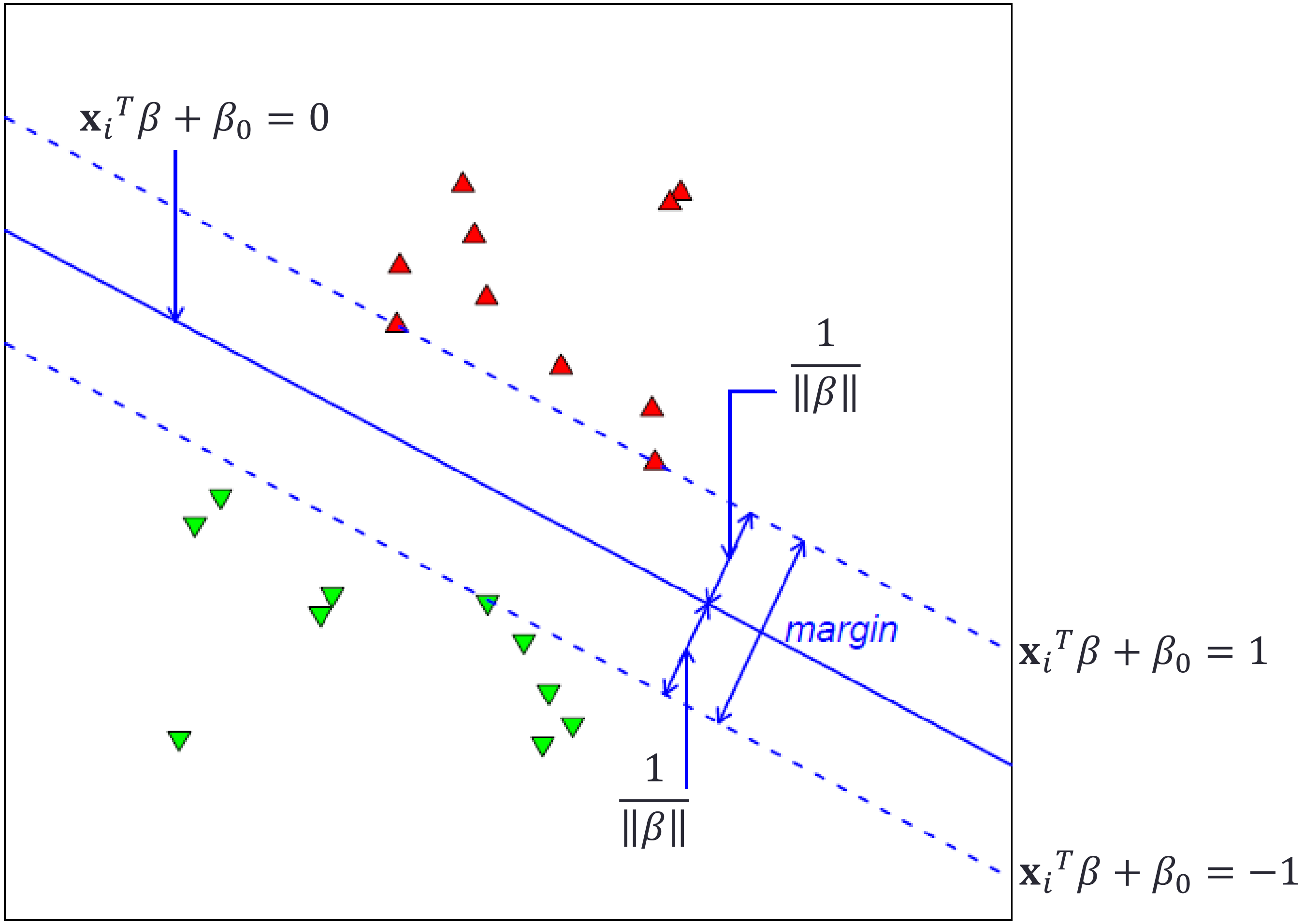
- Since we can scale β_0, β by a positive factor arbitrarily, we can use the normalization $M \cdot ||\beta|| = 1$ (i.e., $||\beta|| = 1/M$) instead of $||\beta|| = 1$. Then the problem can be rephrased as

$$\min_{\beta_0, \beta} \quad \frac{1}{2} ||\beta||^2$$

$$\text{subject to } y_i(\mathbf{x}_i{}^T \beta + \beta_0) \geq 1,\ i = 1, \ldots, n$$

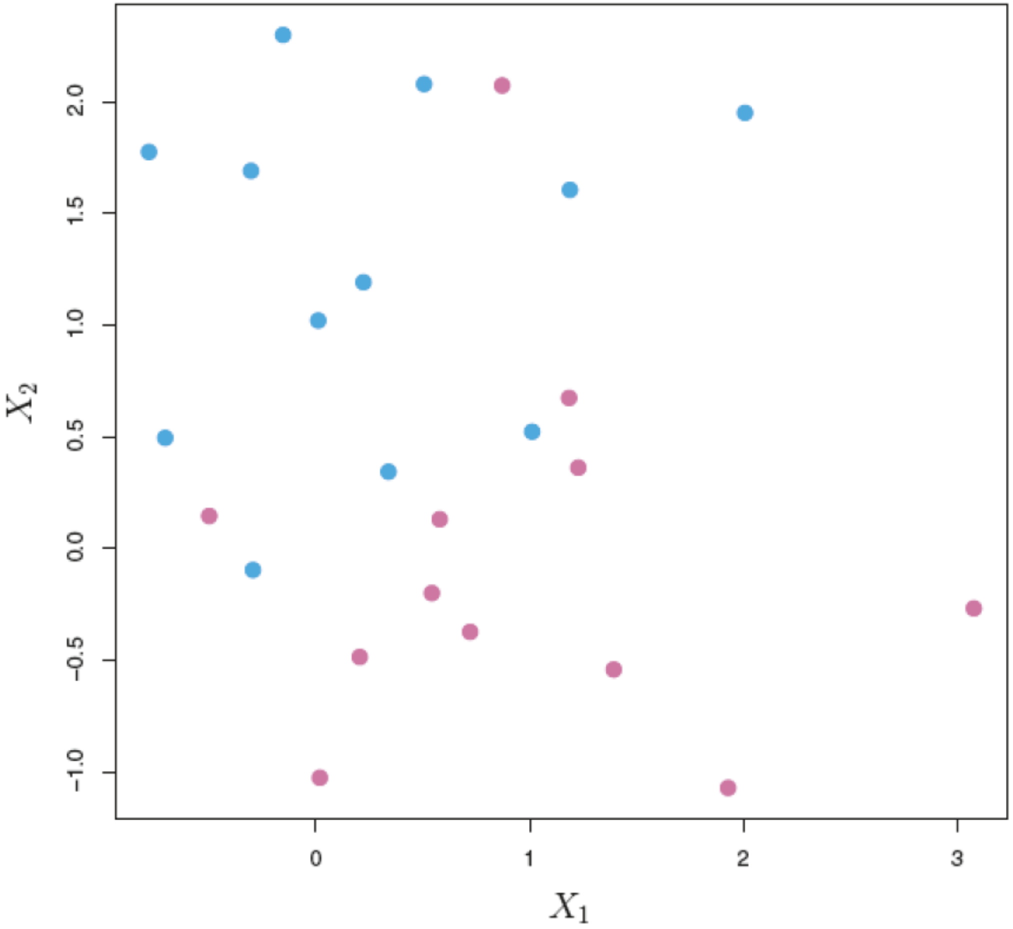
  - M disappeared. The margin $M = 1/||\beta||$.
  - The two dashed hyperplanes are set as $\mathbf{x}_i{}^T \beta + \beta_0 = \pm 1$.
  - The problem is a standard quadratic programming problem.
  - Support vectors are data points $(\mathbf{x}_i, y_i)$ such that the equality holds, i.e., $y_i(\mathbf{x}_i{}^T \beta + \beta_0) = 1$, in other words, on the dashed hyperplanes.

Illustrative Plot: Rephrased Formulation


$\mathbf{x}_i{}^T \beta + \beta_0 = 0$
$\frac{1}{\|\beta\|}$
margin
$\mathbf{x}_i{}^T \beta + \beta_0 = 1$
$\frac{1}{\|\beta\|}$
$\mathbf{x}_i{}^T \beta + \beta_0 = -1$

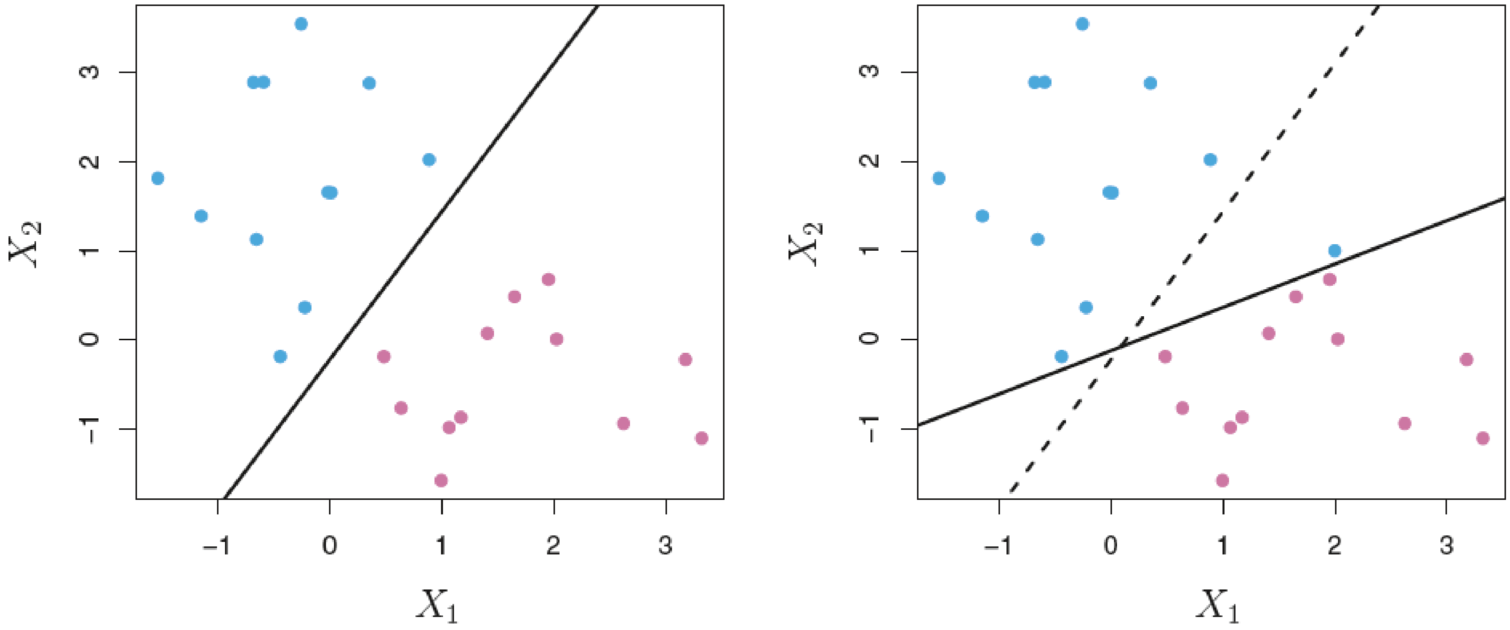
Non-Separable Case

- Data that are not linearly separable



Disadvantage of Maximal Margin Classifier

- In the separable case, the separating hyperplane is sensitive to individual observations: the addition of a single observation may lead to dramatic change in the maximal margin hyperplane.
- Reason: margin is tiny



Idea of Support Vector Classifier

- Consider a hyperplane that does not perfectly separate the two classes, in the interest of

  (1) Greater robustness to individual observations, and

  (2) Correct classification of most of the training observations

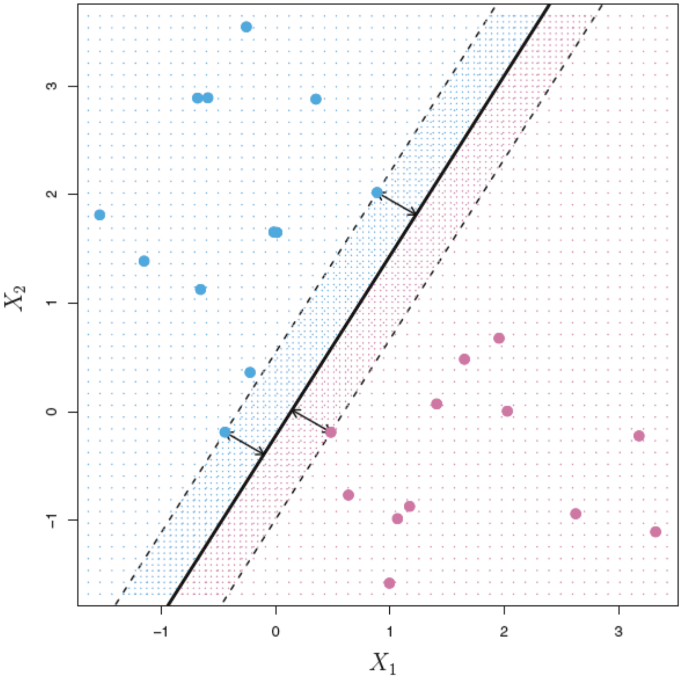
- It could be worthwhile to misclassify a few training points in order to do a better job in classifying the remaining points.

- Soft margin classifier: the margin is "soft" because it can be violated by some of the training points.

Idea of Support Vector Classifier (Cont.)

- Rather than seeking the largest possible margin so that every data point is not only on the correct side of the hyperplane but also on the correct side of the margin, we instead allow some data points to be on the incorrect side of the margin, or even the incorrect side of the hyperplane.

- A data point can be not only on the wrong side of the margin, but also on the wrong side of the hyperplane. Data points on the wrong side of the hyperplane correspond to training observations that are misclassified by the support vector classifier.



Optimization Formulation in Non-Separable Case

- When the data are not separable, the constraints need to be relaxed by introducing some slack variables ξ_i.

- A standard SVM formulation:

$$\min_{\beta_0,\beta,\xi_1,\xi_2,\ldots,\xi_n} \frac{1}{2}||\beta||^2$$

subject to $y_i(\mathbf{x}_i{}^T\beta + \beta_0) \geq 1 - \xi_i,\ i = 1, \ldots, n$

$$\xi_i \geq 0,\ \sum_{i=1}^{n} \xi_i \leq C$$

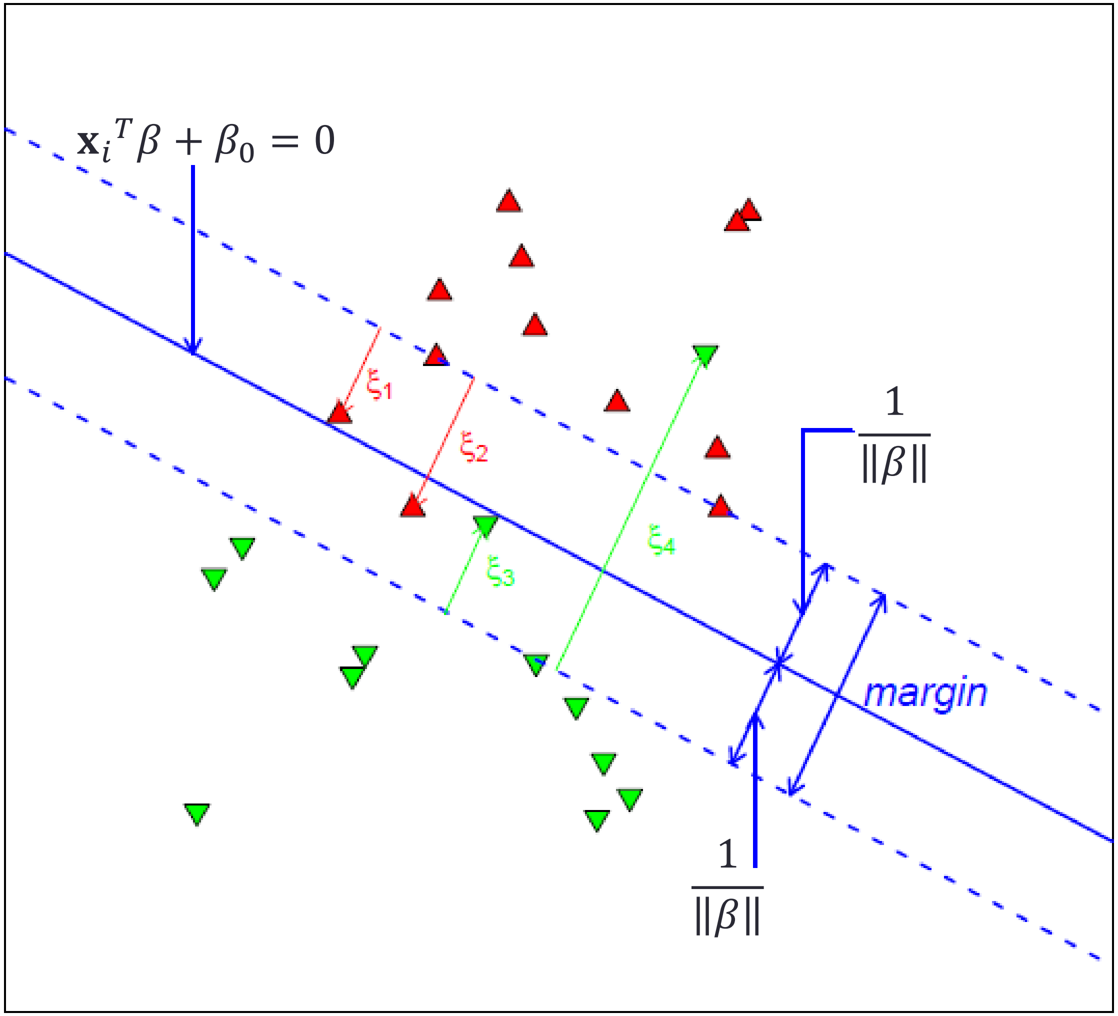
- C controls the size of slack variables, or the severity of the misclassified observations.

Slack Variables and Tuning Parameter

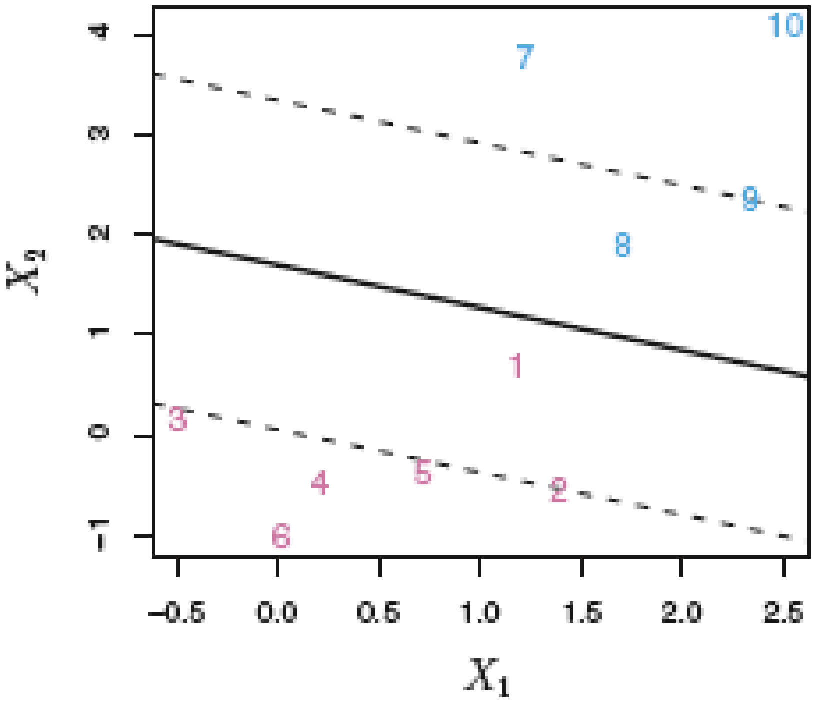
- For a sample i,
  - $\xi_i = 0$ margin not violated (on the correct side of the margin)
  - $\xi_i > 0$ margin violated (on the wrong side of the margin)
  - $\xi_i > 1$ hyperplane boundary crossed (on the wrong side of the hyperplane, or leakage)

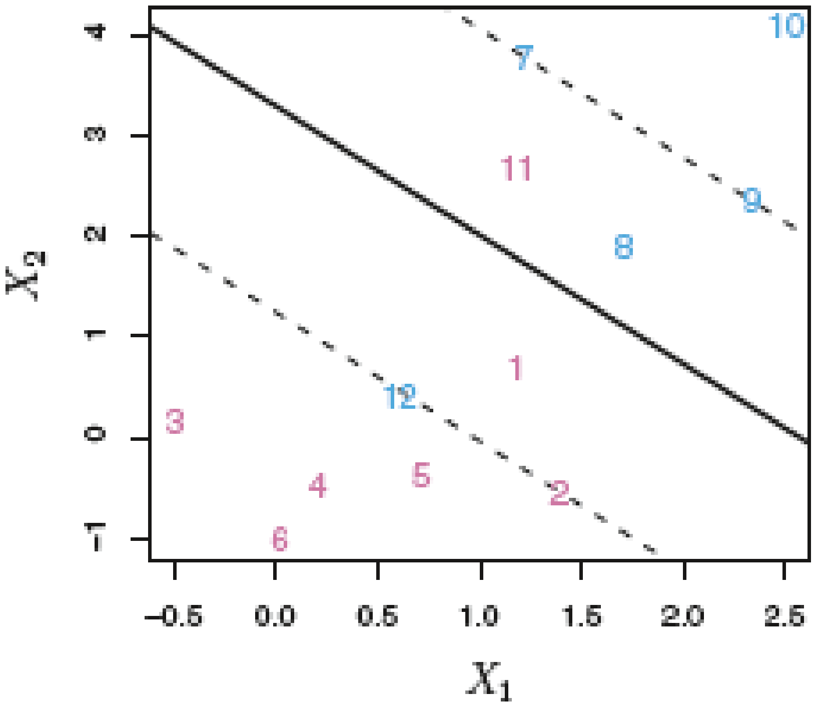
C: tuning parameter, a budget for the amount that the margin can be violated (a budget of relaxation)

Illustrative Plot: Nonseparable



Effect of Slack Variable





3,4,5,6: on the correct side of the margin
2: on the margin
1: on the wrong side of the margin

7,10: on the correct side of the margin
9: on the margin
8: on the wrong side of the margin

11,12: on the wrong side of the margin and the wrong side of the hyperplane

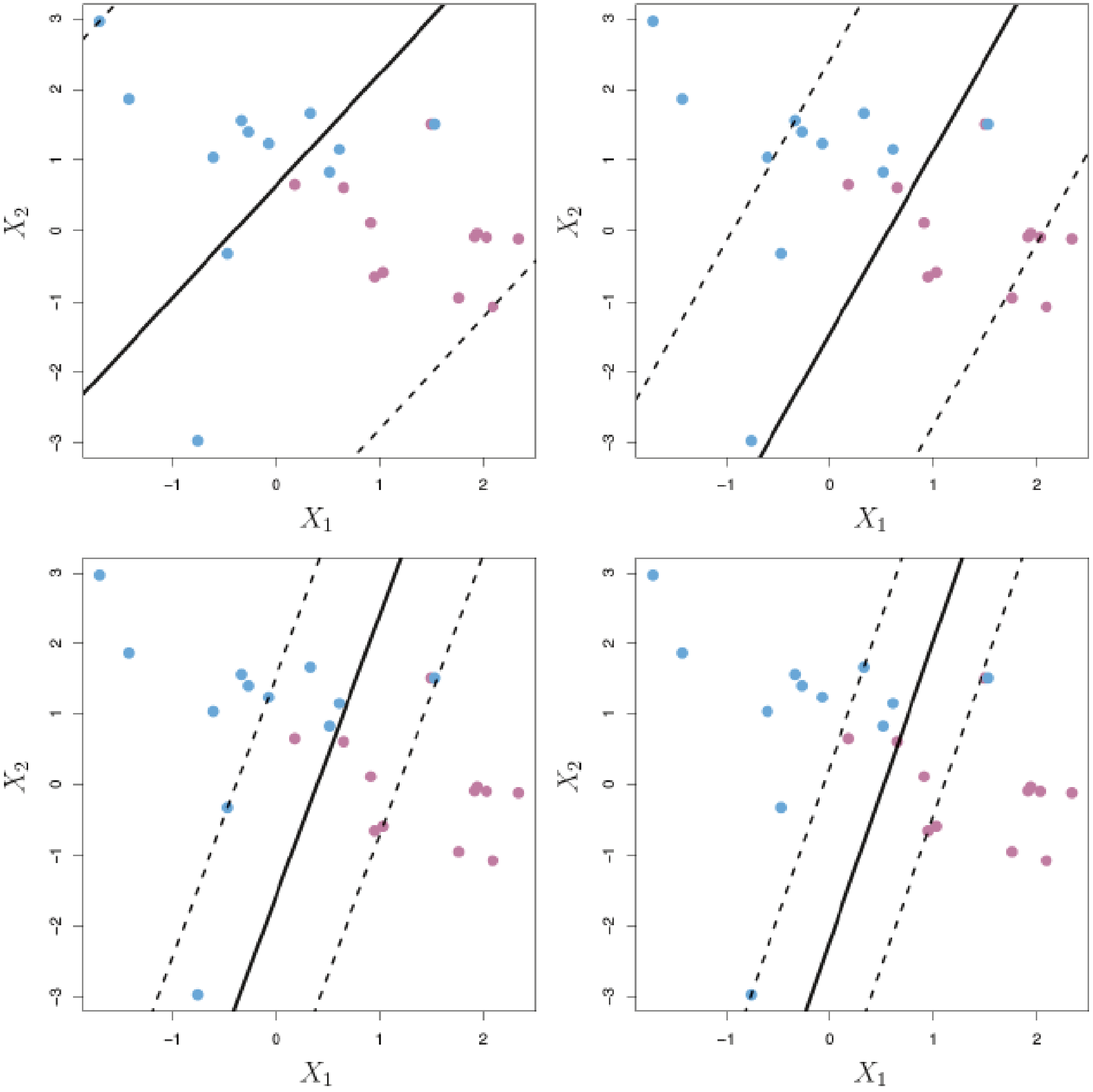
Tuning Parameter

- Different values of C lead to different hyperplanes.
  - If $C = 0$, all ξ_i must be 0, i.e., all data points have to be correctly classified and violation of margin is not allowed.
  - As C increases, it is more tolerant of misclassification, and so the margin will widen.

- Effect of C
  - Small C: low bias, high variance
  - Large C: low variance, high bias

- Optimal C can be determined by cross validation.

Example of Tuning Parameter



Optimization

- First, the SVM formulation is equivalent to

$$\min_{\beta_0,\beta,\xi_1,\xi_2,\ldots,\xi_n} \frac{1}{2}||\beta||^2 + C\sum_{i=1}^{n}\xi_i$$

subject to $y_i(\mathbf{x}_i{}^T\beta + \beta_0) \geq 1 - \xi_i,\ i = 1, \ldots, n,\ \xi_i \geq 0$

- This problem can be solved using Lagrange multiplier.

Optimization (Cont.)

$$\min_{\beta_0,\beta,\xi_1,\xi_2,\ldots,\xi_n} \frac{1}{2}||\beta||^2 + C\sum_{i=1}^{n}\xi_i$$

subject to $y_i(\mathbf{x}_i{}^T\beta + \beta_0) \geq 1 - \xi_i$, $i = 1, \ldots, n$, $\xi_i \geq 0$

- Lagrange (primal) function is

$$L_p = \frac{1}{2}||\beta||^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i(y_i(\mathbf{x}_i{}^T\beta + \beta_0) - 1 + \xi_i) - \sum_{i=1}^{n}\mu_i\,\xi_i$$

which we minimize w.r.t. β_0, β and ξ_i, subject to $\xi_i, \alpha_i, \mu_i \geq 0$.

Dual Form

- Setting the respective derivatives to zero

$$\beta = \sum_{i=1}^{n} \alpha_i \, y_i \mathbf{x}_i \qquad 0 = \sum_{i=1}^{n} \alpha_i \, y_i \qquad C = \alpha_i + \mu_i$$

as well as $\xi_i, \alpha_i, \mu_i \geq 0$.

- Substituting them back to the primal function yields the Wolfe dual function

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{i'=1}^{n} \alpha_i \alpha_{i'} y_i y_{i'} \langle \mathbf{x}_i, \mathbf{x}_{i'} \rangle$$

subject to $\sum_{i=1}^{n} \alpha_i y_i = 0$, $0 \leq \alpha_i \leq C$.

KKT Conditions

- The solutions to the primal and the dual forms are equivalent

$$\hat{\beta} = \sum_{i=1}^{n} \hat{\alpha}_i \, y_i \mathbf{x}_i$$

- The solutions to the primal and the dual forms must satisfy the Karush-Kuhn-Tucker conditions:

$$\alpha_i(y_i({\mathbf{x}_i}^T \beta + \beta_0) - 1 + \xi_i) = 0$$

$$\mu_i \xi_i = 0$$

$$y_i({\mathbf{x}_i}^T \beta + \beta_0) \geq 1 - \xi_i$$

More on KKT Conditions

- $\hat{\alpha}_i > 0$ only when $y_i\left(\mathbf{x}_i{}^T\hat{\beta} + \hat{\beta}_0\right) = 1 - \hat{\xi}_i$, or equivalently,

$$y_i\left(\mathbf{x}_i{}^T\hat{\beta} + \hat{\beta}_0\right) \leq 1$$

  These points are called **support vectors**.

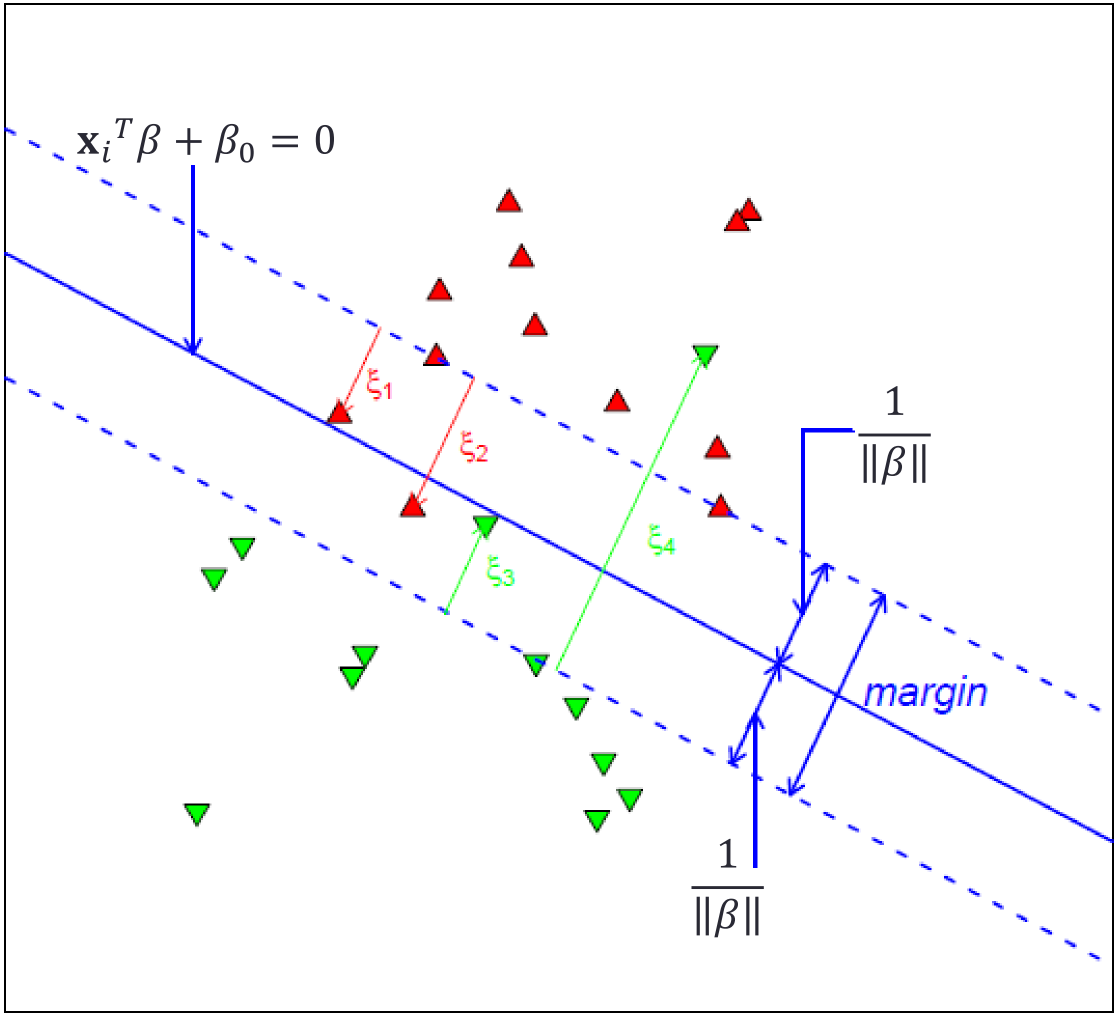
- Among support vectors, some have $\hat{\xi}_i > 0$ and $\hat{\alpha}_i = C$; and others lie on the edge of the margin $\hat{\xi}_i = 0$, and thus $0 < \hat{\alpha}_i < C$.

- The solution of the SVM problem

$$\hat{\beta} = \sum_{i \epsilon S} \hat{\alpha}_i \, y_i \mathbf{x}_i$$

  where $S = \{i: y_i\left(\mathbf{x}_i{}^T\hat{\beta} + \hat{\beta}_0\right) = 1 - \ \hat{\xi}_i\}$, i.e., the set of support vectors. SVM got its name because it is determined by support vectors rather than the entire training data.

Illustrative Plot: Support Vectors in Nonseparable Case



Remarks

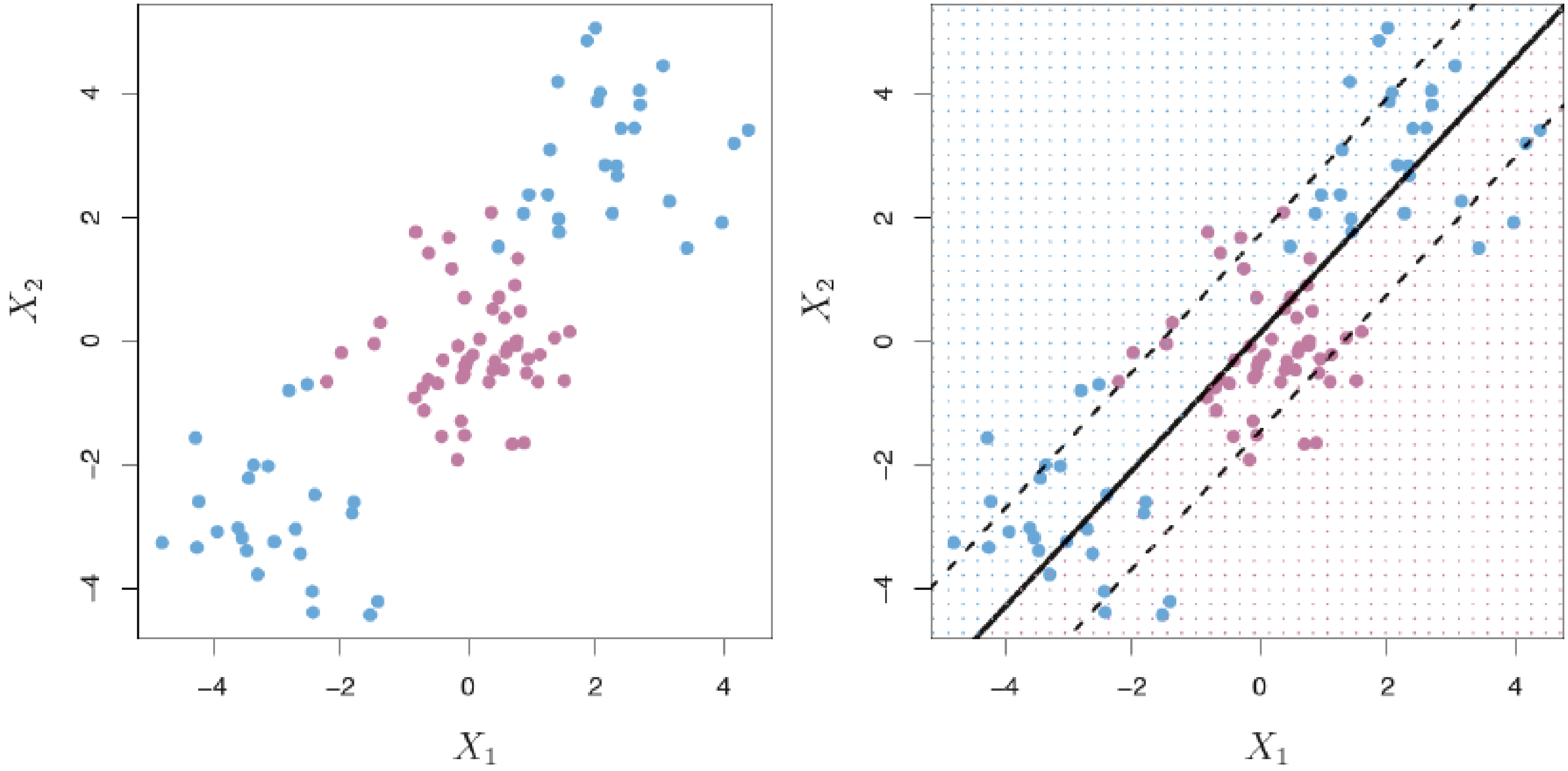
- Computational complexity of training support vector classifier is characterized by **the number of support vectors** rather than the dimensionality. So it works well on small as well as high dimensional problems.

- They are not suitable for larger datasets because the training time with SVM can be high and much more computationally intensive.

- The solution is robust because it is insensitive to outliers (data points that are significantly far from the decision boundary.)

Nonlinear Decision Boundary: A Toy Example

- When the classes have nonlinear boundary, linear SVM will perform very poorly.

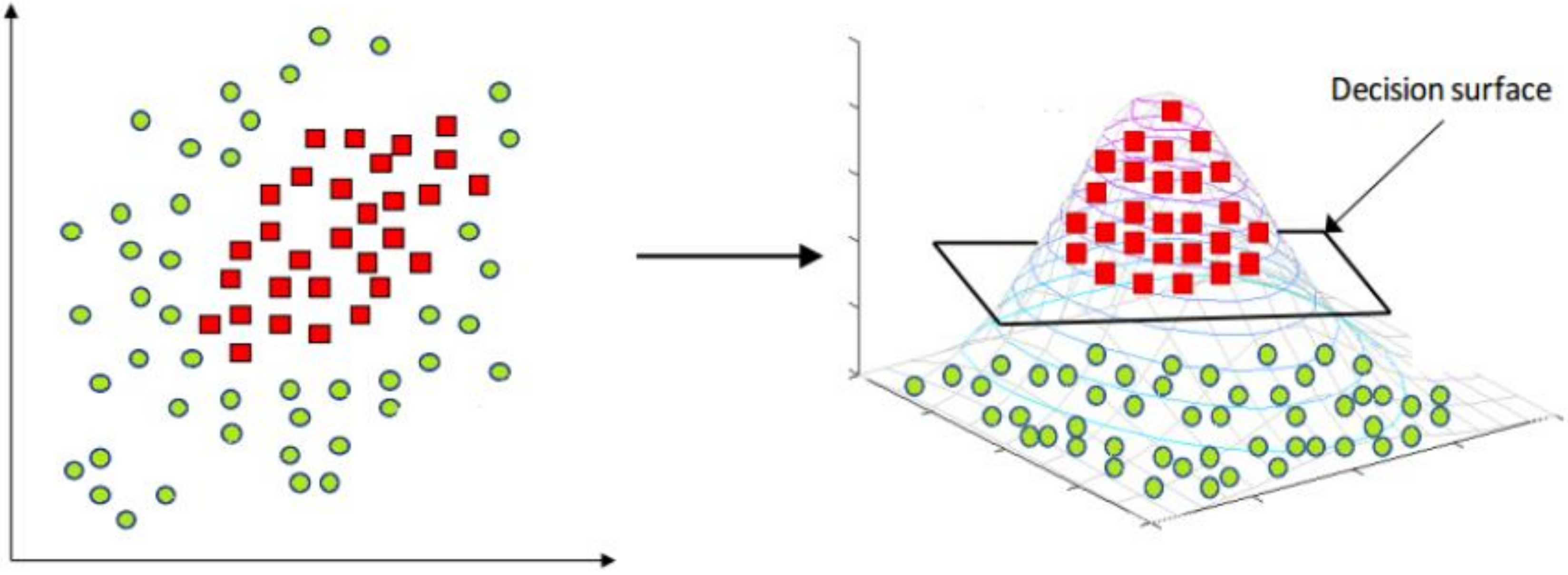


Extending Linear SVM to Nonlinear SVM

- The key idea of extending linear SVM, and many other linear procedures, to nonlinear is to:
  - Enlarge the predictor space using basis expansion functions $h_1(\mathbf{x}), \dots, h_M(\mathbf{x})$.
  - Construct a linear separating hyperplane $f(x) = \beta^T h(\mathbf{x}) + \beta_0$ in the enlarged space for better training performance.
  - The linear separating hyperplane in the enlarged space can be translated into a nonlinear separating hyperplane in the original space.

Illustrative Plot: Extension to Nonlinear Decision Boundary

- Cover’s theorem: pattern-classification problem cast in a high dimensional space non-linearly is more likely to be linearly separable than in a low dimensional space.



Recall: Linear Regression

- Linear regression may not work when the relationship between predictors and the response is nonlinear. The solution is to add transformations (e.g., quadratic and cubic terms) of predictors into the model. Similar idea can be applied here.

- For example: instead of fitting a support vector classifier using the p features $X_1, X_2, \ldots, X_p$, we can enlarge the feature space by using $X_1, X_1^2, X_2, X_2^2 \ldots, X_p, X_p^2$.

- The linear separating hyperplane is

$$f(x) = \beta_0 + \sum_{j=1}^{p} \beta_{j1} x_{ij} + \sum_{j=1}^{p} \beta_{j2} x_{ij}^2$$

Modified Optimization Formulation

- The SVM formulation becomes

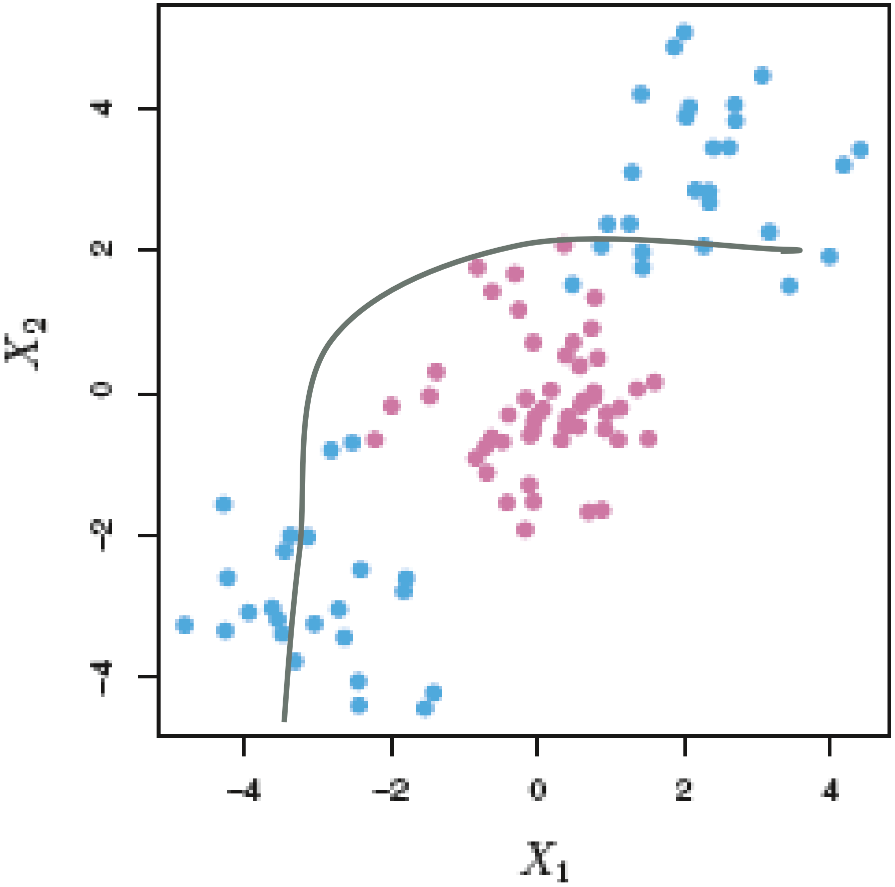
$$\min_{\beta_0,\beta_{11},\beta_{21},\ldots,\beta_{p1},\beta_{12},\ldots,\beta_{p2},\xi_1,\xi_2,\ldots,\xi_n} \frac{1}{2}||\beta||^2$$

subject to $y_i\left(\beta_0 + \sum_{j=1}^{p} \beta_{j1} x_{ij} + \sum_{j=1}^{p} \beta_{j2} x_{ij}^2\right) \geq 1 - \xi_i$

$$\xi_i \geq 0,\ \sum_{i=1}^{n} \xi_i \leq C$$

$$\sum_{i=1}^{p} \sum_{k=1}^{2} \beta_{jk}^2 = 1$$

Example



However…

- Including quadratic terms is only one way to enlarge the feature space in order to accommodate nonlinearity.
- There are many possible ways to enlarge the feature space. Unless we are careful, we could end up with a huge number of features. Then computations would become unmanageable.
- The support vector machine allows us to enlarge the feature space in a way that leads to efficient computations (**kernel trick**).

Inner Product

- Let us define an inner product (dot product) of two vectors:

$$\langle a, b \rangle = \sum_{i=1}^{r} a_i b_i$$

For real vectors, it is simply $a^T b$.

Kernel Trick

- In the enlarged space, the dual form becomes

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{i'=1}^{n} \alpha_i \alpha_{i'} y_i y_{i'} \langle h(\mathbf{x}_i), h(\mathbf{x}_{i'}) \rangle$$

subject to $\sum_{i=1}^{n} \alpha_i y_i = 0$, $0 \le \alpha_i \le C$.

- The solution is $\hat{\beta} = \sum_{i=1}^{n} \hat{\alpha}_i \, y_i h(\mathbf{x}_i)$, and

$$\hat{f}(\mathbf{x}) = \hat{\beta}^T h(\mathbf{x}) + \hat{\beta}_0 = \sum_{i=1}^{n} \hat{\alpha}_i \, y_i \langle h(\mathbf{x}_i), h(\mathbf{x}) \rangle + \hat{\beta}_0$$

- The interesting part is that the formulation relies on $h(\mathbf{x})$ only through their inner products.

Kernels

- Define

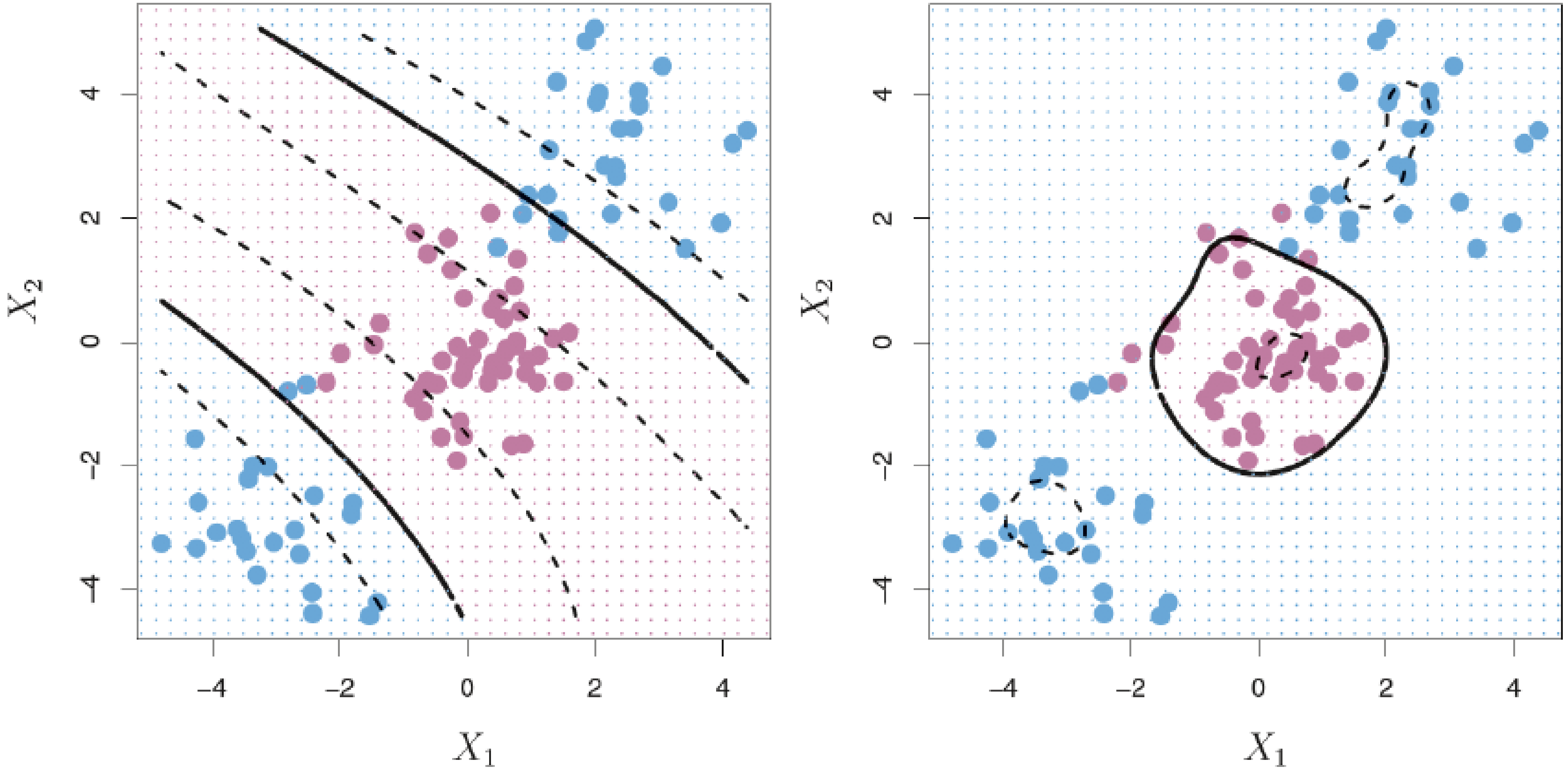
$$K(\mathbf{x}, \mathbf{x}') = \langle h(\mathbf{x}), h(\mathbf{x}') \rangle$$

and thus we need not satisfy $h(\cdot)$ at all, but only $K(\cdot,\cdot)$. $K(\cdot,\cdot)$ is called the **kernel** (function), which is a function that quantifies the similarity of two observations.

- Popular choices of $K(\cdot,\cdot)$:
  - Linear: $K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T\mathbf{x}'$
  - Degree-*d* polynomial: $K(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x}^T\mathbf{x}')^d$
  - Radial (Gaussian): $K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma||\mathbf{x} - \mathbf{x}'||^2)$

Example of Nonlinear SVM

- Left: SVM with a polynomial kernel of degree 3; Right: SVM with a radial kernel



Corresponding Basis Expansion Functions

- In general, it is hard to find the basis expansion functions $h(\cdot)$ corresponding to a kernel function except the polynomial kernel.

- Example: consider a polynomial kernel of degree 2 and $p = 2$. If we choose

$$h_1(\mathbf{x}) = 1$$
$$h_2(\mathbf{x}) = \sqrt{2}X_1$$
$$h_3(\mathbf{x}) = \sqrt{2}X_2$$
$$h_4(\mathbf{x}) = X_1^2$$
$$h_5(\mathbf{x}) = X_2^2$$
$$h_6(\mathbf{x}) = \sqrt{2}X_1X_2$$

then we have the polynomial kernel

$$K(\mathbf{x}, \mathbf{x}') = \langle h(\mathbf{x}), h(\mathbf{x}') \rangle = (1 + \mathbf{x}^T\mathbf{x}')^2$$

Advantage of Kernel

- We use a kernel rather than simply enlarge the feature space using functions of the original features.

- There is advantage on computation. Using kernel, we only need to compute the kernel $K(x_i, x_{i'})$ for $\binom{n}{2}$ distinct pairs $x_i, x_{i'}$. This can be done without explicitly working in the enlarged feature space.

- This is important because in many applications of SVMs, the enlarged feature space is so large that computations are intractable. For some kernels, such as the radial kernel, the feature space is implicit and infinite-dimensional, so we could never do the computations there anyway!

Radial Kernel

- The radial kernel

$$K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2)$$

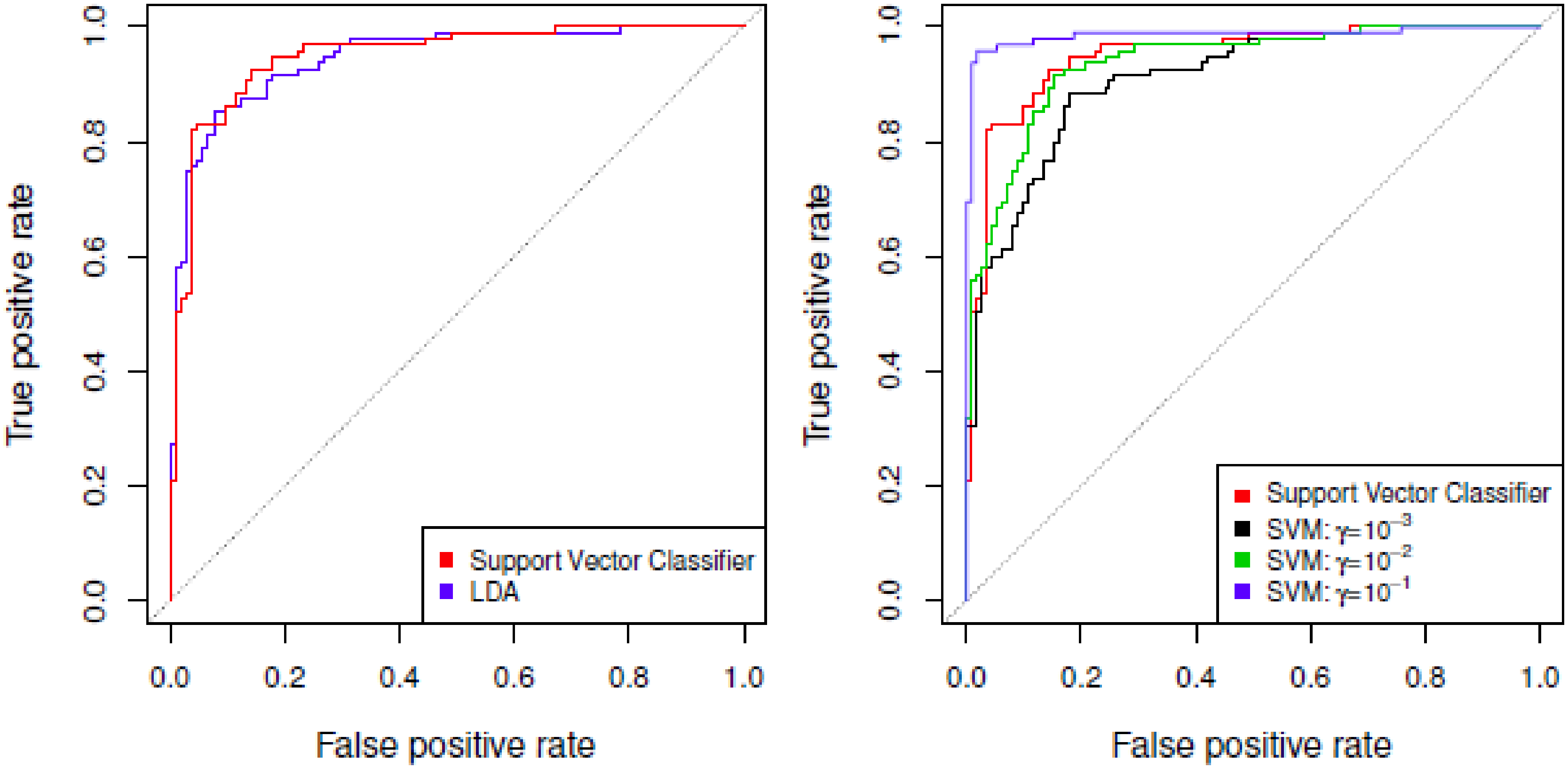
  is also known as the Gaussian kernel.

- When $\mathbf{x}_i$ is far away from $\mathbf{x}$, $K(\mathbf{x}_i, \mathbf{x})$ will be very tiny and thus $\mathbf{x}_i$ has little effect on $\hat{f}(\mathbf{x})$.

- The radial kernel has very local behavior, and only nearby training observations have effect on the prediction of a test observation.

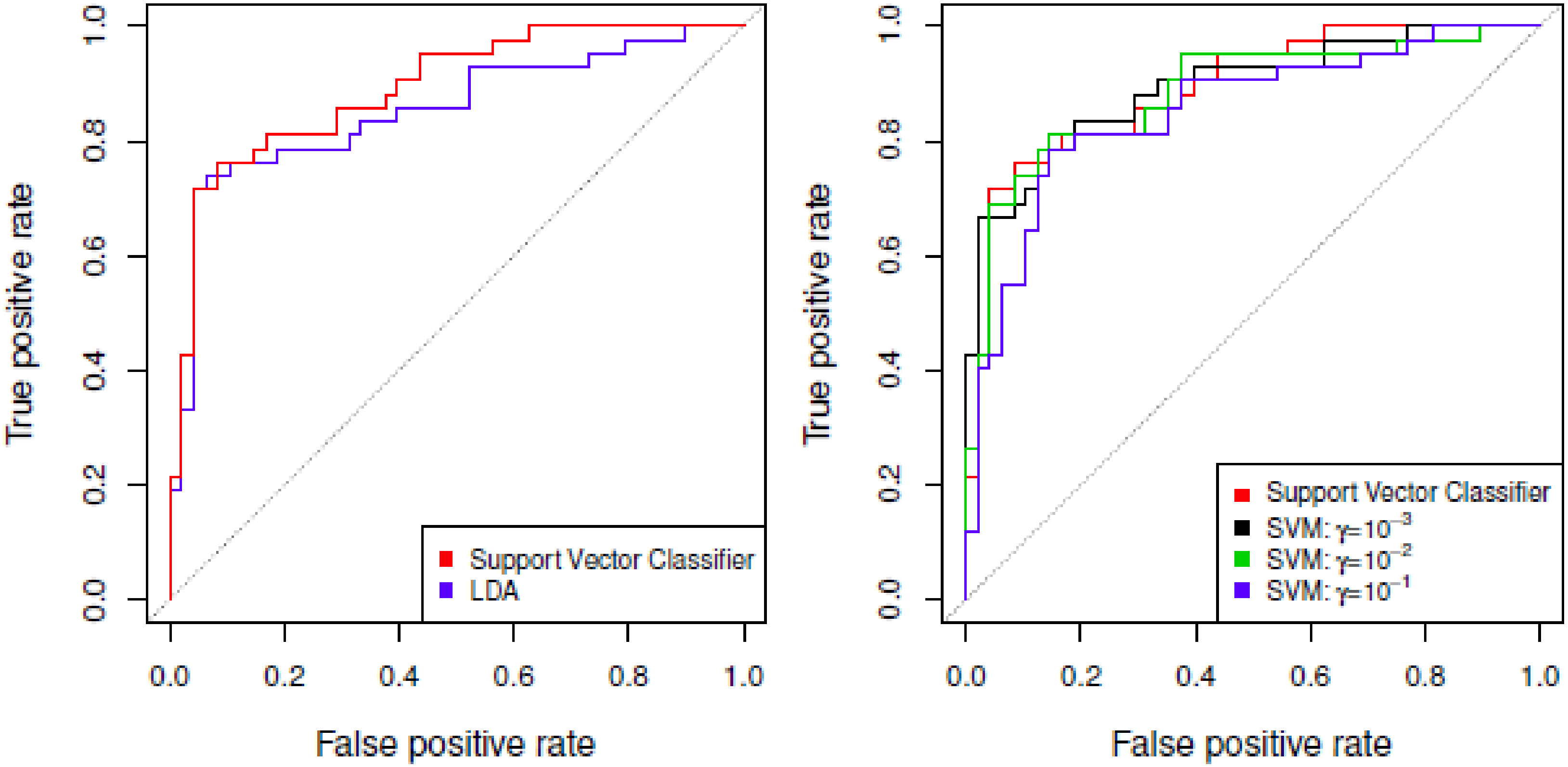
Example: Heart Data

- ROC curves for the training



Example: Heart Data (Cont.)

- ROC curves for testing



Multiclass SVM

When the response $y \in \{1, \dots K\}$ with $K > 2$:

- One-versus-one approach
  - Construct C_K^2 binary SVM classifiers, each compares one pair of classes.
  - Assign the test observation to the class to which it was most frequently assigned in these C_K^2 pairwise classification.
- One-versus-rest approach
  - Construct K binary SVM classifiers, each compares one class to the rest $K - 1$ classes.
  - Assign the test observation to the class for which the predicted function value is the largest.

Support Vector Regression

- SVM can be extended to regression, called support vector regression (SVR).
- Least squares regression seeks coefficients $\beta_0, \beta_1, \ldots, \beta_p$ such that the sum of squared residuals is minimized. SVR instead seeks coefficients that minimize a different type of loss function.
- In SVR, only residuals larger in absolute value than some positive constant contribute to the loss function, which is similar to robust regression. This is an extension of the margin used in support vector classifiers to the regression setting.