Support Vector Machine

Mohammad Emtiyaz Khan
EPFL

Oct 29, 2015

Motivation

By changing the cost function from Logistic to Hinge, we obtain SVMs. However, the resulting cost is difficult to optimize. We will use duality (similar to kernelized ridge) to show a surprising result: the solution to the dual problem is sparse. The non-zero entries will be our support vectors.

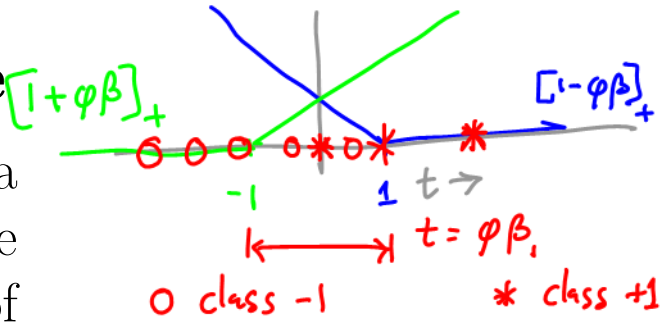
Logistic likelihood (recap)

$$\frac{1}{N}\sum_{n=1}^{N} -\log p\left(y_n \mid \tilde{\varphi}_n^T\beta\right) + \frac{1}{2}\sum_{j=1}^{M}\beta_j^2$$

$$= \begin{cases} \tilde{\varphi}_n^T\beta + \log\left(1+e^{\tilde{\varphi}_n^T\beta}\right), & \text{when } y_n=+1 \\ \log\left(1+e^{\tilde{\varphi}_n^T\beta}\right), & \text{when } y_n=-1 \end{cases}$$

Hinge loss:

$$[t]_+ = \max(0,t)$$



Support vector machine

Throughout, we will work with a classification problem and assume that $y_n \in \{-1,+1\}$ (instead of $\in \{0,1\}$). Also, we will work with $\tilde{\boldsymbol{\phi}}(\mathbf{x})$ instead of $\tilde{\mathbf{x}}$ (bias included).

SVM optimizes the following cost:

$$\min_{\beta} \sum_{n=1}^{N}[1-y_n\tilde{\boldsymbol{\phi}}_n^T\boldsymbol{\beta}]_+ + \frac{\lambda}{2}\sum_{j=1}^{M}\beta_j^2$$

where the first term is the Hinge loss defined as $[t]_+ = \max(0,t)$. A "conventional" definition is shown below:

$$\min_{\beta} \sum_{n=1}^{N} C[1-y_n\tilde{\boldsymbol{\phi}}_n^T\boldsymbol{\beta}]_+ + \frac{1}{2}\sum_{j=1}^{M}\beta_j^2.$$

Allowing an arbitrary Margin:

The minimum does not change with a rescaling, of β

$\times M$

$$\Rightarrow \min_{\beta} \sum \left[M - y_n\tilde{\varphi}_n^T(M\beta)\right]_+ + \frac{\lambda}{2M}\sum_j M^2\beta_j^2$$

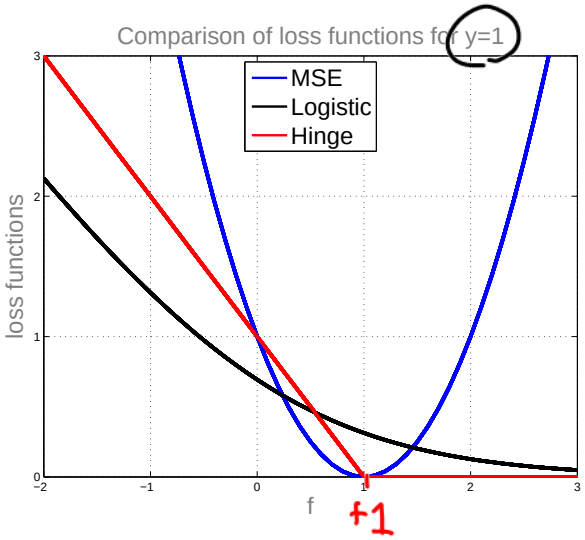
$\tilde{\beta}_j$

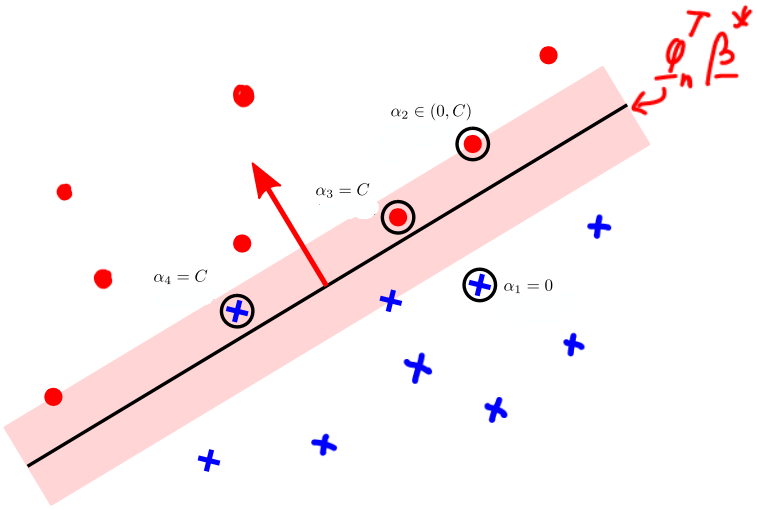
$$\min_{\tilde{\beta}} \sum \left[M - y_n\tilde{\varphi}_n^T\tilde{\beta}\right]_+ + \frac{\tilde{\lambda}}{2}\sum_j \tilde{\beta}_j^2$$

Hinge vs MSE and logistic

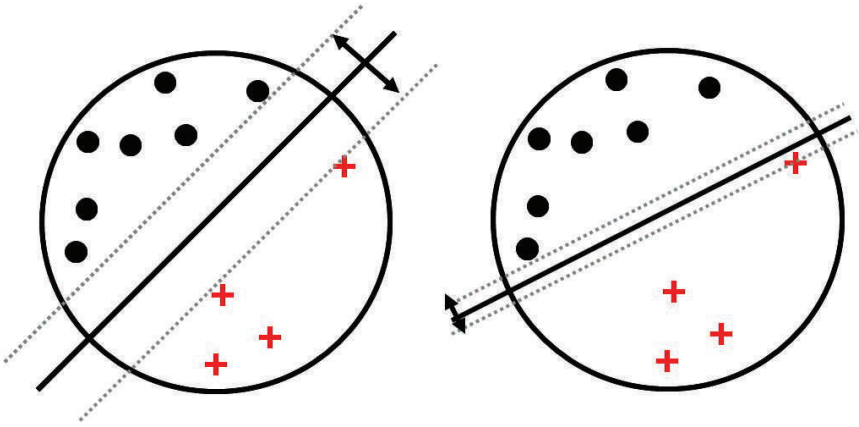
Consider $y \in \{-1, +1\}$ with prediction $f \in \mathbb{R}$, then the three cost functions can be written as follows:

$$Hinge(f) = [1 - yf]_+$$
$$MSE(f) = (1 - yf)^2$$
$$logisticLoss(f) = \log\left(1 + e^{-yf}\right)$$





Notice the margin in the Hinge loss.
SVM is a maximum margin method.



See section 14.5.2.2 of KPM book.

Issues with optimization

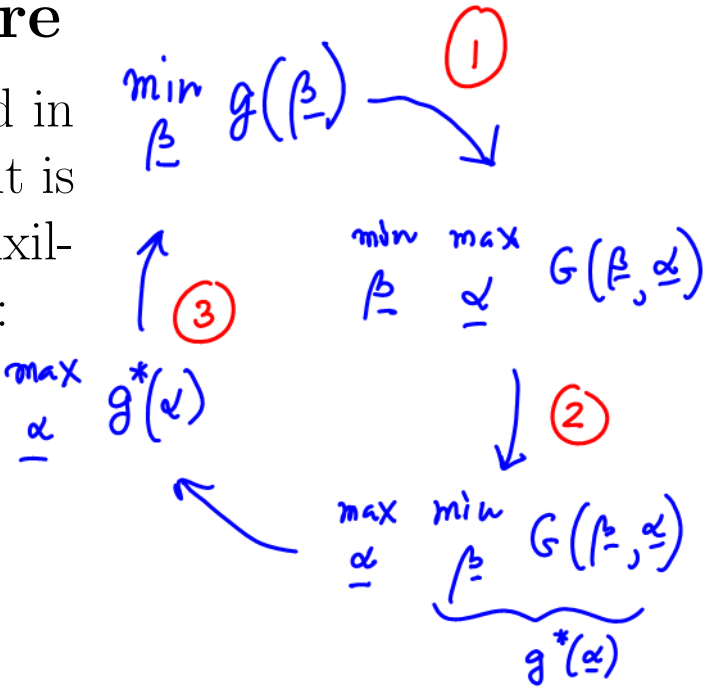
Is this function convex? Is it differentiable?

$$\min_{\beta} \sum_{n=1}^{N} C[1 - y_n \widetilde{\boldsymbol{\phi}}_n^T \boldsymbol{\beta}]_+ + \frac{1}{2} \sum_{j=1}^{M} \beta_j^2. \quad := g(\underline{\beta})$$

Duality: the big picture

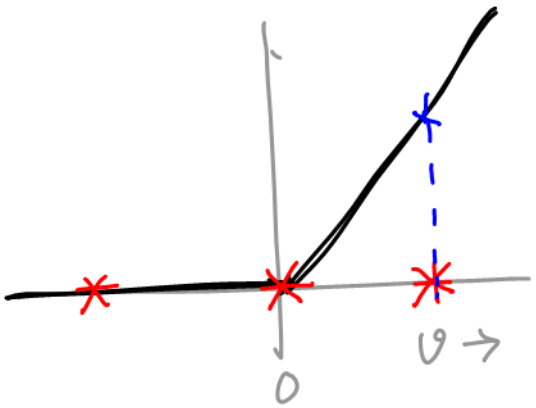
Let us say that we are interested in optimizing a function $g(\boldsymbol{\beta})$ and it is a difficult problem. Define an auxiliary function $G(\boldsymbol{\beta}, \boldsymbol{\alpha})$ as follows:

$$g(\boldsymbol{\beta}) = \max_{\alpha} G(\boldsymbol{\beta}, \boldsymbol{\alpha}).$$



Three questions.

1. How do you set $G(\boldsymbol{\alpha}, \boldsymbol{\beta})$?
2. When is it OK to switch max and min?
3. When is the dual better than the primal, and why?



Q1: How do you set $G(\boldsymbol{\alpha}, \boldsymbol{\beta})$?

$$C[v_n]_+ = \max(0, Cv_n) = \max_{\alpha_n} \alpha_n v_n \text{ where } \alpha_n \in [0, C]$$

$$C[1 - y_n \widetilde{\boldsymbol{\phi}}_n^T \boldsymbol{\beta}]_+ = \max_{\alpha_n \in [0,C]} \alpha_n (1 - y_n \widetilde{\boldsymbol{\phi}}_n^T \boldsymbol{\beta})$$

We can rewrite the problem as:

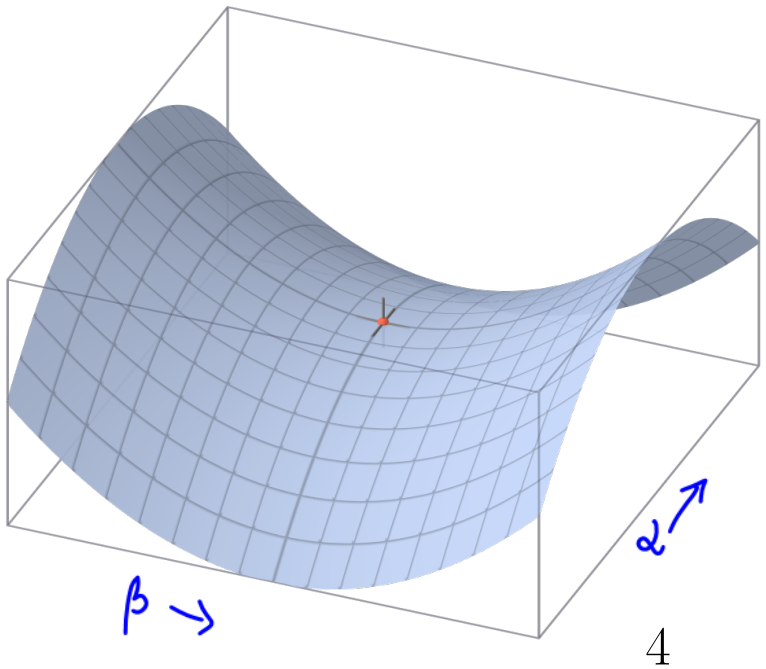
$$\min_{\boldsymbol{\beta}} \max_{\boldsymbol{\alpha}\in[0,C]^N} \sum_{n=1}^{N} \alpha_n(1 - y_n \widetilde{\boldsymbol{\phi}}_n^T \boldsymbol{\beta}) + \frac{1}{2}\sum_{j=1}^{M} \beta_j^2$$

This is differentiable, convex in $\boldsymbol{\beta}$ and concave in $\boldsymbol{\alpha}$.

Q2: When is it OK to switch max and min? Using a minimax theorem, it is OK to do so when $G(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is convex in $\boldsymbol{\beta}$ and concave in $\boldsymbol{\alpha}$, and the sets over which $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are optimized are convex. In ~~this~~ that case, we have:

$$\min_{\beta} \max_{\alpha} G(\boldsymbol{\beta}, \boldsymbol{\alpha}) = \max_{\alpha} \min_{\beta} G(\boldsymbol{\beta}, \boldsymbol{\alpha})$$

See Bertsekas' "Nonlinear Programming" for many more variants of this type of duality.



Switch the min and max and optimize w.r.t. $\boldsymbol{\beta}$ to get the dual.

$$\max_{\alpha \in [0,C]^N} \min_{\beta} \sum_{n=1}^{N} \alpha_n (1 - y_n \widetilde{\boldsymbol{\phi}}_n^T \boldsymbol{\beta}) + \frac{1}{2} \sum_{j=1}^{M} \beta_j^2$$

Take derivative w.r.t. $\boldsymbol{\beta}$: Derive this at home!

$$\frac{\partial G}{\partial \boldsymbol{\beta}} = - \left[\sum_{n=1}^{N} \alpha_n y_n \widetilde{\boldsymbol{\phi}}_n \right] + \begin{bmatrix} 0 \\ \boldsymbol{\beta}_{1:M} \end{bmatrix}$$

where $\boldsymbol{\beta}_{1:M}$ is a vector of all β_j except β_0.

Equating this to 0, we get:

$$\boldsymbol{\beta}^*_{1:M} = \sum_{n=1}^{N} \alpha_n y_n \boldsymbol{\phi}_n = \boldsymbol{\Phi}^T \text{diag}(\mathbf{y}) \boldsymbol{\alpha} = \boldsymbol{\Phi}^T \mathbf{Y} \boldsymbol{\alpha}$$

$$\boldsymbol{\alpha}^T \mathbf{y} = 0$$

where $\mathbf{Y} := \text{diag}(\mathbf{y})$.

Plugging $\boldsymbol{\beta}^*$ back in, we get the dual problem:

$$\max_{\alpha \in [0,C]^N} \boldsymbol{\alpha}^T \mathbf{1} - \tfrac{1}{2} \boldsymbol{\alpha}^T \mathbf{Y} \boldsymbol{\Phi} \boldsymbol{\Phi}^T \mathbf{Y} \boldsymbol{\alpha}$$

quadratic

$$\text{subject to } \boldsymbol{\alpha}^T \mathbf{y} = 0$$

Q3: When is the dual better than the primal and why?

(1) The dual is a differentiable (but constrained) least-squares problem.

$$\max_{\alpha \in [0,C]^N} \boldsymbol{\alpha}^T \mathbf{1} - \tfrac{1}{2} \boldsymbol{\alpha}^T \mathbf{Q} \boldsymbol{\alpha},$$

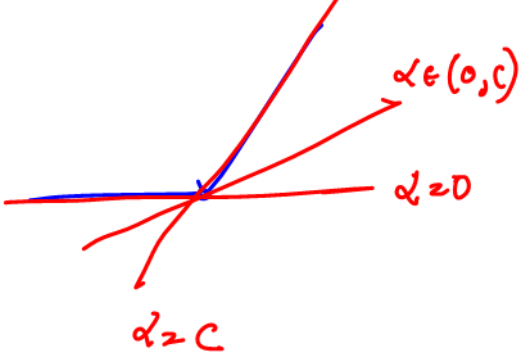
where $\mathbf{Q} := \text{diag}(\mathbf{y}) \boldsymbol{\Phi} \boldsymbol{\Phi}^T \text{diag}(\mathbf{y})$. Optimization is super easy using Sequential Minimal Optimization (SMO). See Wikipedia for details.

Read the details on your own.

Summary: Take two variables α_1 and α_2 and fix others. This gives rise to a 1-D quadratic problem. Minimize and repeat by choosing two different elements of α.

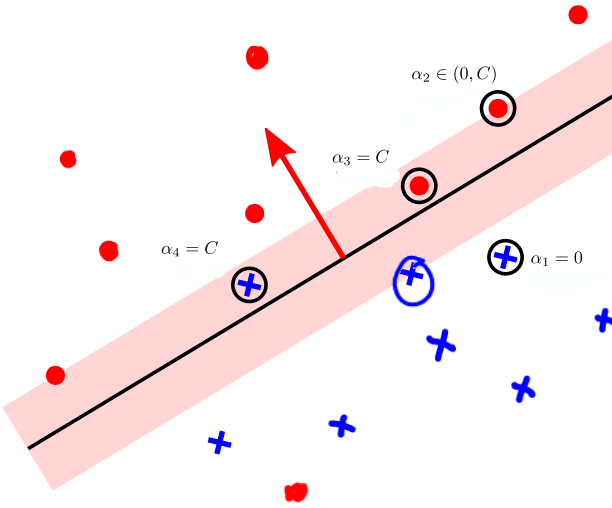
(2) The dual is naturally kernelized (just like the kernelized ridge) with $\mathbf{K} := \boldsymbol{\Phi} \boldsymbol{\Phi}^T$.

(3) The solution $\boldsymbol{\alpha}$ is sparse, and is non-zero only for the training examples that are instrumental in determining the decision boundary.



Recall that α_n is the slope of lines that are lower bounds to the Hinge loss.

$$C[1 - y_n f_n]_+ = \max_{\alpha_n \in [0,C]} \alpha_n (1 - y_n f_n)$$

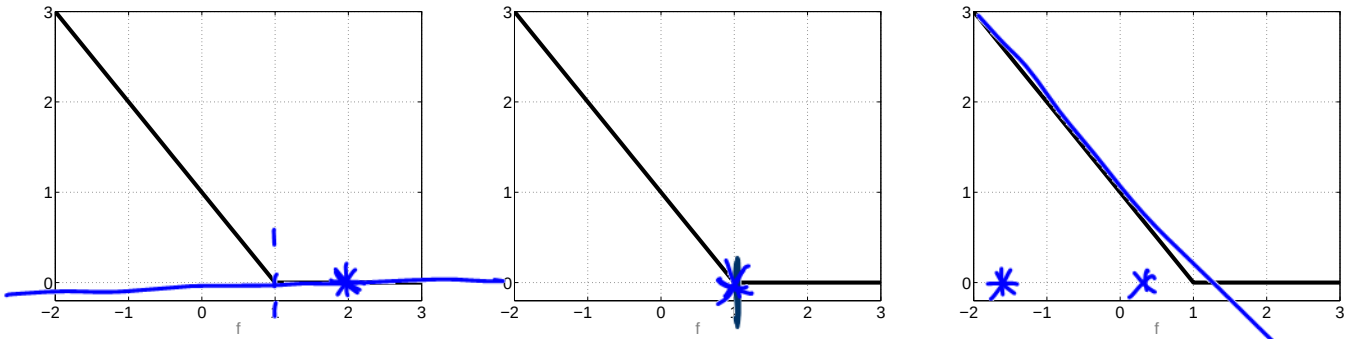


There are 3 kinds of data vectors $\widetilde{\phi}_n$.

1. Not support vectors. Examples that lie outside the margin, therefore $\alpha_n = 0$.

2. Essential support vectors. Examples that lie right on the margin, therefore $\alpha_n \in (0, C)$.

3. Bound support vectors. Examples that lie ~~inside~~ away from the margin, therefore $\alpha_n = C$.
   ↳ on the other side.



(c) Not SV (d) Essential SV (e) Bound SV

$\alpha_n = 0$

Q: Why don't we use a step function instead of the hinge loss?

A: Convexity.

Issues with SVM

- There are no obvious probabilistic interpretation of SVM.
- Extension to multiclass is difficult (see Section 14.5.2.4 of KPM book).
- Choosing C is difficult in the presence of Kernels.
- The method does not work for positive semidefinite Kernels.

To do

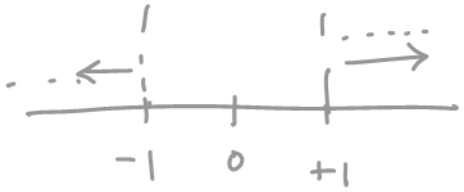
1. Understand and visualize hinge loss and the margin.
2. Get comfortable with duality. Work out the derivation for SVM.
3. Clearly understand the reasons why dual is better than the primal.
4. What does "support vector" mean? Why do they arise? Where do they lie in the data space?
5. Read about SMO algorithm from Wikipedia and implement it.
6. Read about SVM for regression (section 14.5.1 of KPM).
7. Read Section 14.5.2.4 of KPM book and understand why extension of SVM to multiclass is difficult.
8. Read about maximum-margin methods in section 14.5.2.2 of KPM book.
9. Resource: SVM tutorial by Christopher J.C. Burges at http://research.microsoft.com/pubs/67119/svmtutorial.pdf

10. Read SVM from HTF.

$$\sum_{n=1}^{N} \left[1 - y_n \tilde{\varphi}_n^T \beta\right]_+ + \frac{\lambda}{2} \sum_{j=1}^{M} \beta_j^2$$

For linearly separable case, all n are correctly classified

$$\Rightarrow \quad y_n \tilde{\varphi}_n^T \beta \geq 1 \quad , \forall n$$

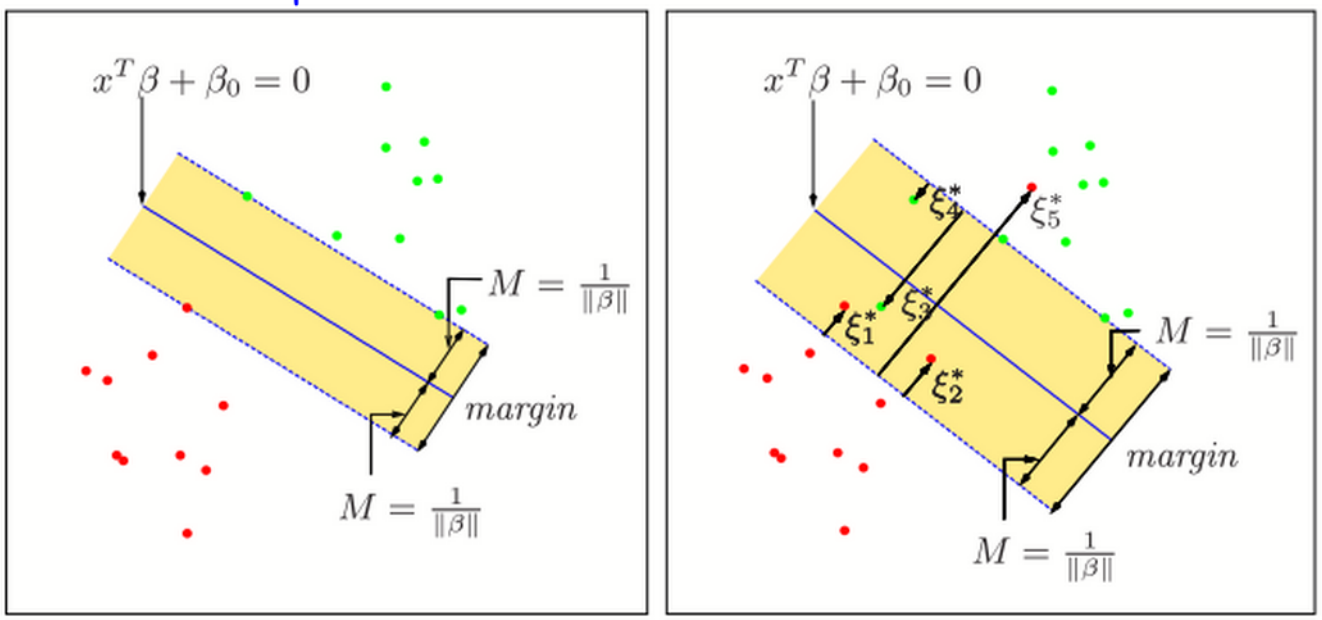


$y_n = -1$: $\tilde{\varphi}_n^T \beta \leq 0 - 1$

$y_n = +1$: $\tilde{\varphi}_n^T \beta \geq 0 + 1$

$\Rightarrow$ An equivalent problem

$$\left[\min_{\beta} \frac{\lambda}{2} \sum_j \beta_j^2 \;, \text{ s.t. } y_n(\tilde{\varphi}_n^T \beta) \geq 1\right]$$



This is important

FIGURE 12.1. *Support vector classifiers. The left panel shows the separable case. The decision boundary is the solid line, while broken lines bound the shaded maximal margin of width* $2M = 2/\|\beta\|$. *The right panel shows the nonseparable (overlap) case. The points labeled* ξ_j^* *are on the wrong side of their margin by an amount* $\xi_j^* = M\xi_j$; *points on the correct side have* $\xi_j^* = 0$. *The margin is maximized subject to a total budget* $\sum \xi_i \leq$ *constant. Hence* $\sum \xi_j^*$ *is the total distance of points on the wrong side of their margin.*

When the problem is not linearly separable,

$$\min_{\beta} \frac{\lambda}{2} \sum_j \beta_j^2 \;, \text{ s.t. } y_n(\tilde{\varphi}_n^T \beta) \geq 1 - e_n$$

$$+ \sum_n e_n^2$$