
PATTERN RECOGNITION AND MACHINE LEARNING

CHAPTER 7: SPARSE KERNEL MACHINES

Learning Objectives

- 1、 What are support vector machines?
 - 2、 What are maximum (soft) margin classifiers?
 - 3、 What the relation between SVMs and logistic regression?
 - 4、 How to use SVMs for regression?
 - 5、 What are relevance vector machines?
 - 6、 How to use RVMs for regression?
 - 7、 How to use RVMs for classification?
 - 8、 What is the mechanism for RVMs to have sparse solutions?
-

Outlines

- Support Vector Machines
 - SVMs and Logistic Regression
 - SVMs for Regression
 - Relevance Vector Machines
 - RVMs for Regression
 - RVMs for Classification
-

Support Vector Machines

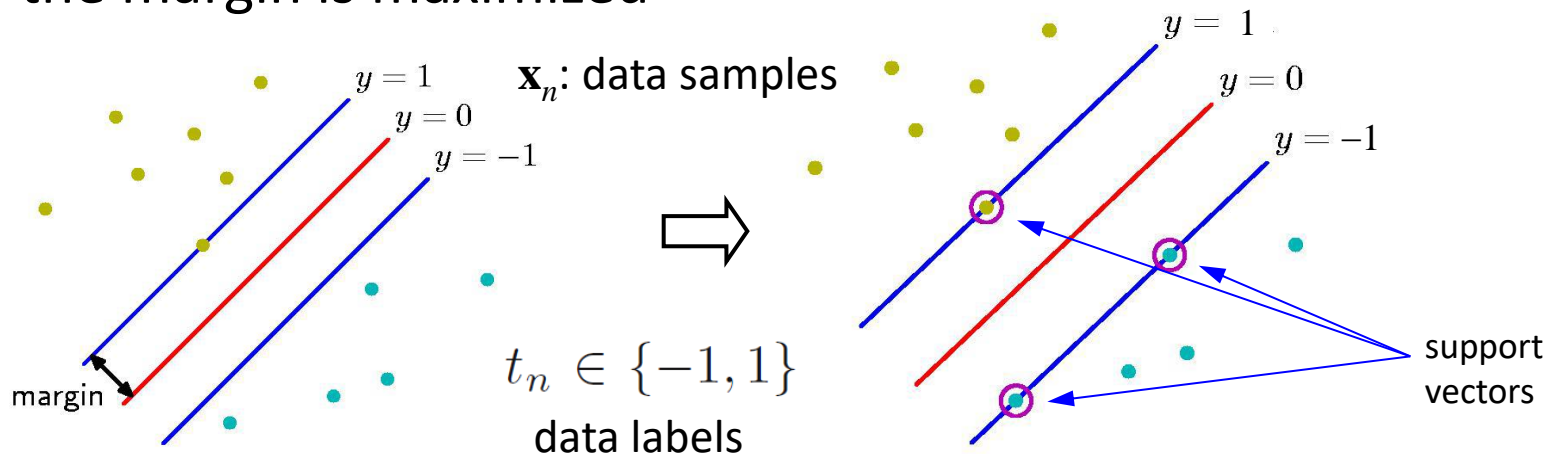
□ Problem settings

$$y(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b$$

- ✓ Two-class classification using linear models
- ✓ Assume that training data set is linearly separable

□ Support vector machine approaches

- ✓ The decision boundary is chosen to be the one for which the margin is maximized



Maximum Margin Classifier I

For all data points, $t_n y(\mathbf{x}_n) > 0$

The distance of a point to the decision surface

$$\frac{t_n y(\mathbf{x}_n)}{\|\mathbf{w}\|} = \frac{t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b)}{\|\mathbf{w}\|}$$

The maximum margin solution

$$\arg \max_{\mathbf{w}, b} \left\{ \frac{1}{\|\mathbf{w}\|} \min_n \left[t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b) \right] \right\}$$

support vectors

A difficult problem !

Maximum Margin Classifier II

- After rescaling w and b , the point closest to the surface becomes

$$t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b) = 1$$

- The constraints for all points become

$$t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b) \geq 1, \quad n = 1, \dots, N$$

“=” means active constraints, “>” means inactive constraints

$$\Rightarrow \boxed{\arg \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 \text{ subject to } t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b) \geq 1, \quad n = 1, \dots, N}$$

An easier problem !

Lagrange Method

- Minimize an object function s. t. constraints of inequality

$$\min_x f(x) \quad \text{s. t.} \quad g(x) \geq 0$$

- By Introducing a Lagrange multiplier $\lambda \geq 0$, then we will have

$$\min_x \max_{\lambda \geq 0} \{\mathcal{L}(x, \lambda) = f(x) - \lambda g(x)\}$$

- When certain conditions are satisfied, its dual problem is

$$\max_{\lambda \geq 0} \min_x \{\mathcal{L}(x, \lambda) = f(x) - \lambda g(x)\}$$

- By setting derivatives of \mathcal{L} w.r.t. x equal to 0, we will have

$$x = h(\lambda), \text{ and then the problem becomes } \lambda^* = \max_{\lambda \geq 0} Q(\lambda)$$

- Finally, $x^* = h(\lambda^*)$ is the solution
-

Dual Representation I

□ Introducing Lagrange multipliers $a_n \geq 0$, then we will have

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2} \|\mathbf{w}\|^2 - \underbrace{\sum_{n=1}^N a_n \{ \overbrace{t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b) - 1}^{\geq 0} \}}_{\leq 0}$$

which minimizes the first part and maximizes the second part:

either $a_n = 0$ or $a_n \neq 0 \cap t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b) = 1$

inactive constraint

active constraint

: support vectors

□ By setting derivatives of L w.r.t. \mathbf{w} and b equal 0, we will have

$$\mathbf{w} = \sum_{n=1}^N a_n t_n \phi(\mathbf{x}_n), \quad 0 = \sum_{n=1}^N a_n t_n$$

Dual Representation II

□ Eliminating \mathbf{w} and b with $\mathbf{w} = \sum_{n=1}^N a_n t_n \phi(\mathbf{x}_n)$ $t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b) = 1$
from L , we will have the dual representation

$$\tilde{L}(\mathbf{a}) = \sum_{n=1}^N a_n - \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m)$$

subject to $a_n \geq 0, n = 1, \dots, N$

$$\sum_{n=1}^N a_n t_n = 0$$

quadratic programming

⇒ solving a_n

$a_n \neq 0$: support vectors

where

$$k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}')$$

Classifier Parameters

- The classifier can be rewritten as

$$y(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b \Rightarrow y(\mathbf{x}) = \sum_{n=1}^N a_n t_n k(\mathbf{x}, \mathbf{x}_n) + b \quad \mathbf{x}_n: \text{support vectors}$$

- After finding a by solving the quadratic programming problem, we need to estimate b . For support vectors, $a_n \neq 0$, we will have

$$t_n \underbrace{\left(\sum_{m \in S} a_m t_m k(\mathbf{x}_n, \mathbf{x}_m) + b \right)}_{y(\mathbf{x}_n)} = 1 \quad t_n^2 = 1$$

$$b = \frac{1}{N_S} \sum_{n \in S} \left(t_n - \sum_{m \in S} a_m t_m k(\mathbf{x}_n, \mathbf{x}_m) \right)$$

where S is the set of support vectors.

Maximum Margin Classifier

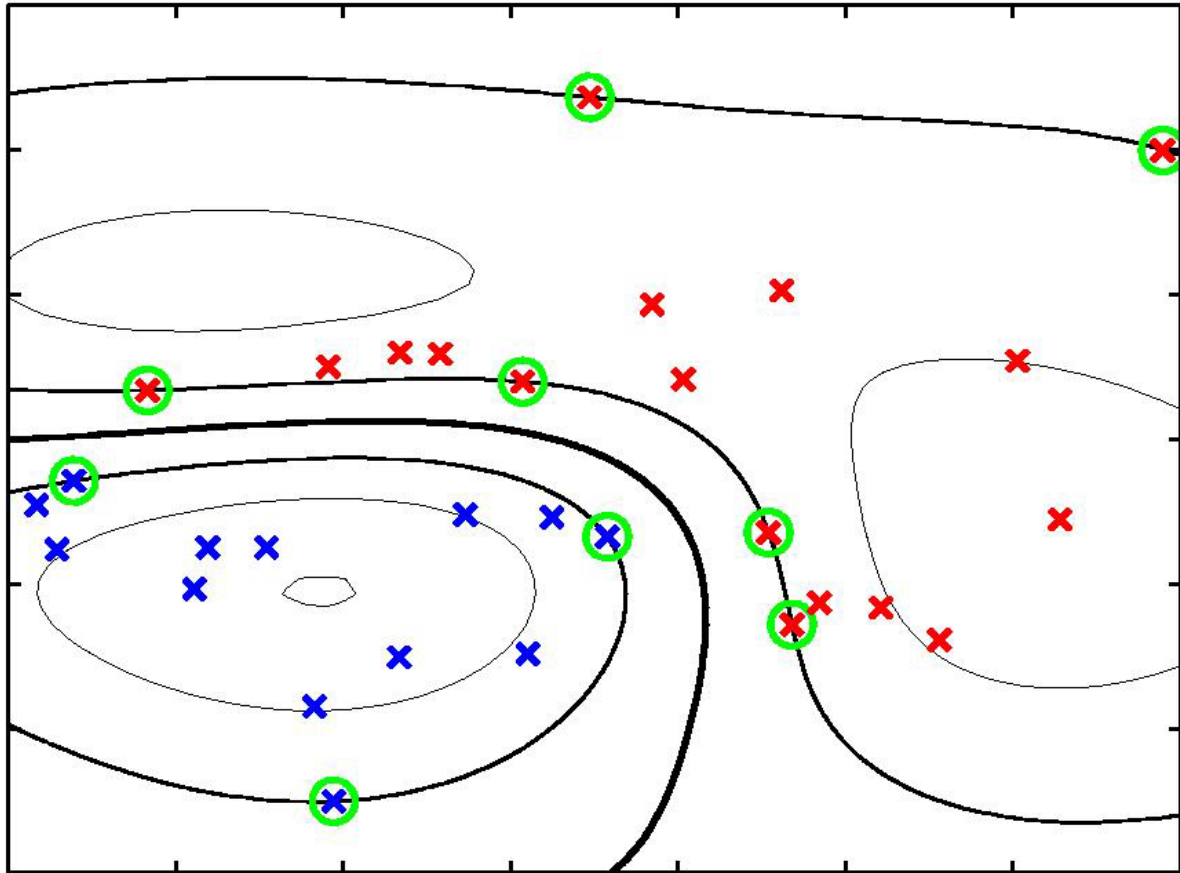
- Training maximum margin classifiers can be generalized as

$$\arg \min_{\mathbf{w}, b} \sum_{n=1}^N E_{\infty}(y(\mathbf{x}_n)t_n - 1) + \lambda \|\mathbf{w}\|^2 \quad \lambda > 0$$

where $E_{\infty}(z)$ is a function that is zero if $z \geq 0$ and ∞ otherwise.

- Such that only support vectors will be selected to optimize model parameters
-

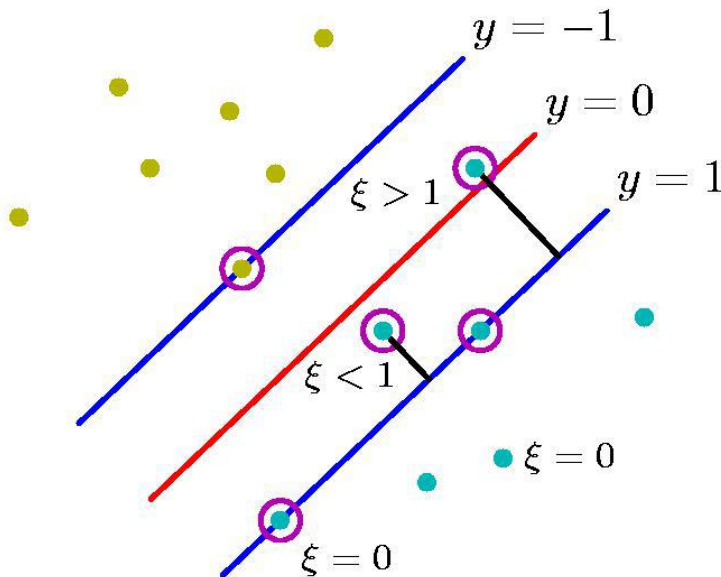
Example of Separable Data Classification



Overlapping Class Distributions

Allow some misclassified examples \rightarrow soft margin

Introduce slack variables $\xi_n \geq 0, n = 1, \dots, N$



$$t_n y(\mathbf{x}_n) = t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b) \geq 1 \Rightarrow t_n y(\mathbf{x}_n) \geq 1 - \xi_n$$

Soft Margin Classifier

Minimize $C \sum_{n=1}^N \xi_n + \frac{1}{2} \|\mathbf{w}\|^2$

$C > 0$: trade-off between minimizing training errors and controlling model complexity

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{n=1}^N \xi_n - \sum_{n=1}^N a_n \{t_n y(\mathbf{x}_n) - 1 + \xi_n\} - \sum_{n=1}^N \mu_n \xi_n$$

$$a_n \geq 0$$

KKT conditions: $t_n y(\mathbf{x}_n) - 1 + \xi_n \geq 0$

$$a_n (t_n y(\mathbf{x}_n) - 1 + \xi_n) = 0$$

: support vectors

$$a_n = 0$$

or

$$t_n y(\mathbf{x}_n) = 1 - \xi_n$$

$$\mu_n \geq 0$$

$$\xi_n \geq 0$$

$$\mu_n \xi_n = 0$$

Dual Representation

- By setting derivatives of L w.r.t. \mathbf{w} , b , and $\{\xi_n\}$ equal 0, we will have

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{n=1}^N a_n t_n \phi(\mathbf{x}_n)$$

$$\frac{\partial L}{\partial b} = 0 \quad \Rightarrow \quad \sum_{n=1}^N a_n t_n = 0$$

$$\frac{\partial L}{\partial \xi_n} = 0 \quad \Rightarrow \quad a_n = C - \mu_n.$$

Dual Representation

□ Dual representation

$$\tilde{L}(\mathbf{a}) = \sum_{n=1}^N a_n - \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m)$$

subject to $0 \leq a_n \leq C, n = 1, \dots, N$

$$\sum_{n=1}^N a_n t_n = 0$$

□ Estimating b

(→ same as hard maximum margin classifiers)

Alternative Formulation


ν -SVM (Schölkopf *et al.*, 2000)

$$\tilde{L}(\mathbf{a}) = -\frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m)$$

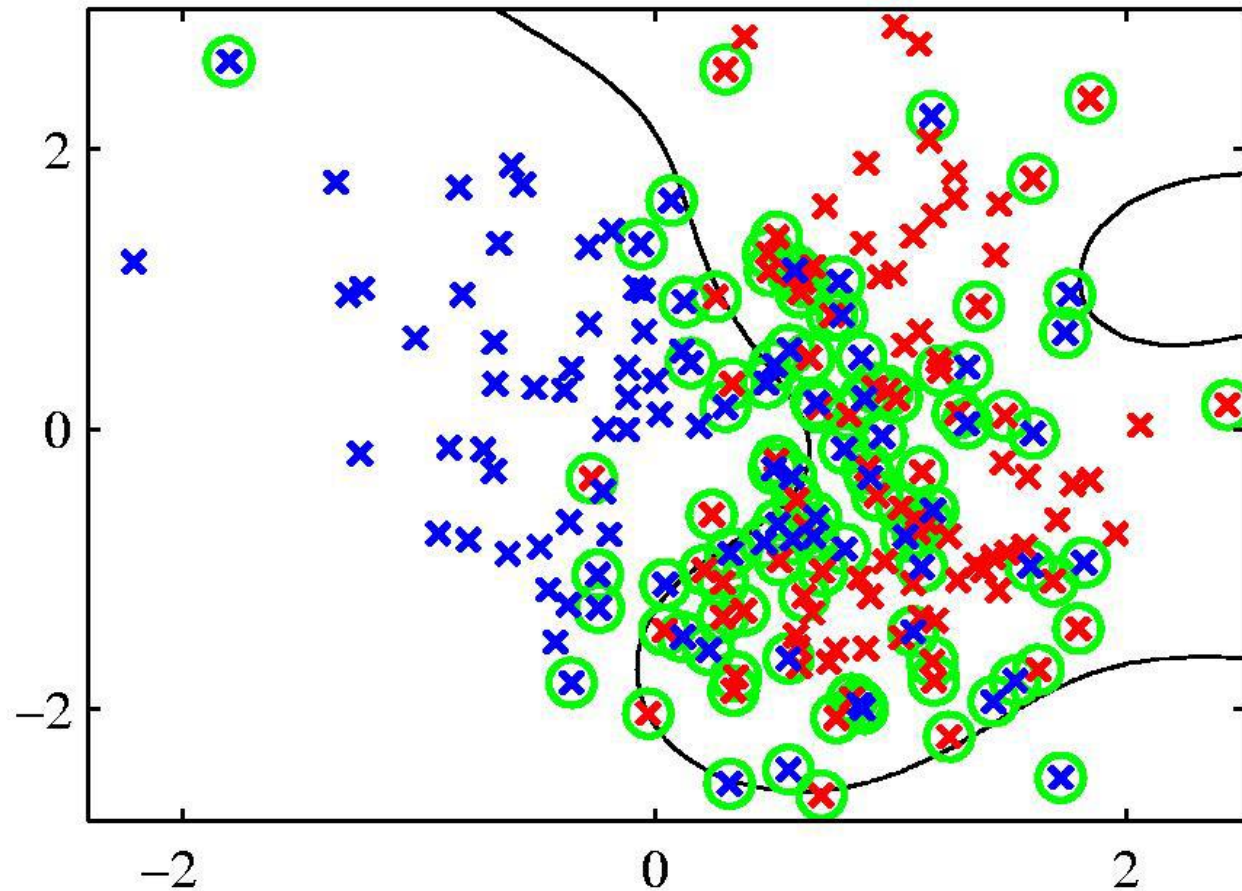
subject to $0 \leq a_n \leq 1/N$

$$\sum_{n=1}^N a_n t_n = 0$$

$$\sum_{n=1}^N a_n \geq \nu$$

- 
- Upper bound on the fraction of margin errors
 - Lower bound on the fraction of support vectors

Nonseparable Data Classification (ν -SVM)



Solutions of the QP Problem

❑ Chunking (Vapnik, 1982)

Idea: the value of Lagrangian is unchanged if we remove the rows and columns of the kernel matrix corresponding to Lagrange multipliers that have value zero

❑ Decomposition methods (Osuna *et al.*, 1996)

❑ Protected conjugate gradients (Burges, 1998)

❑ Sequential minimal optimization (Platt, 1999)

Outlines

- Support Vector Machines
 - SVMs and Logistic Regression
 - SVMs for Regression
 - Relevance Vector Machines
 - RVMs for Regression
 - RVMs for Classification
-

Relation to Logistic Regression I

For data points on the correct side, $\xi = 0$

For the remaining points, $\xi = 1 - y_n t_n$

$$C \sum_{n=1}^N \xi_n + \frac{1}{2} \|\mathbf{w}\|^2 \Leftrightarrow \sum_{n=1}^N E_{SV}(y_n, t_n) + \lambda \|\mathbf{w}\|^2$$

where $\lambda = (2C)^{-1}$

$E_{SV}(y_n, t_n) = [1 - y_n t_n]_+$: hinge error function

where $[\cdot]_+$ denotes the positive part

Relation to Logistic Regression II

□ From maximum likelihood logistic regression

$$p(t = 1 | y) = \sigma(y)$$

$$p(t = -1 | y) = 1 - \sigma(y) = \sigma(-y)$$

$$\Rightarrow p(t | y) = \sigma(yt)$$

□ Error function with quadratic regularization

$$\sum_{n=1}^N E_{LR}(y_n t_n) + \lambda \|\mathbf{w}\|^2$$

$$\text{where } E_{LR}(yt) = \ln(1 + \exp(-yt))$$

Relation to Logistic Regression III

□ Cross-Entropy

y : Bernoulli parameter
 a : natural parameter

$$-\ln p(t|y) = -t \ln y - (1 - t) \ln(1 - y)$$

$$-\ln p(t|a) = -\ln \sigma(at) = \ln(1 + e^{-at}) \quad y = \sigma(a)$$

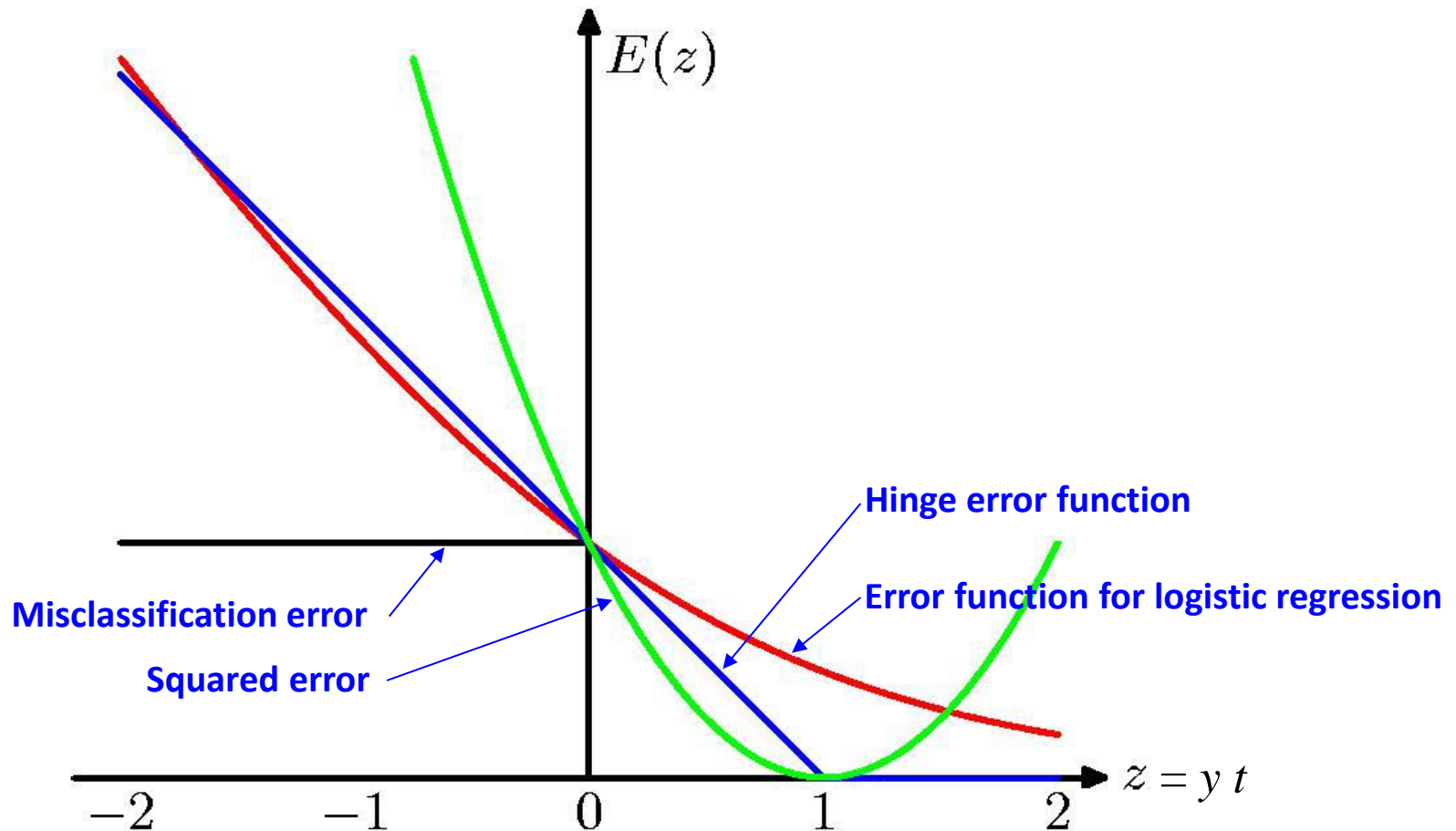
□ Cross-Entropy with prior

$$-\ln p(t|y) + \alpha^{-1} \mathbf{w}^T \mathbf{w} = -t \ln y - (1 - t) \ln(1 - y) + \alpha^{-1} \mathbf{w}^T \mathbf{w}$$

$$-\ln p(t|a) + \alpha^{-1} \mathbf{w}^T \mathbf{w} = -\ln \sigma(at) = \ln(1 + e^{-at}) + \alpha^{-1} \mathbf{w}^T \mathbf{w}$$

softplus: $\ln(1 + e^{-x})$

Comparison of Error Functions



Multiclass SVMs

- ❑ *One-versus-the-rest*: K separate SVMs

Can lead inconsistent results (Figure 4.2)

Imbalanced training sets

Positive class: $+1$, negative class: $-1/(K-1)$

- ❑ An objective function for training all SVMs simultaneously

- ❑ *One-versus-one*: $K(K-1)/2$ SVMs

- ❑ Error-correcting output codes

Generalization of the voting scheme of the *one-versus-one*

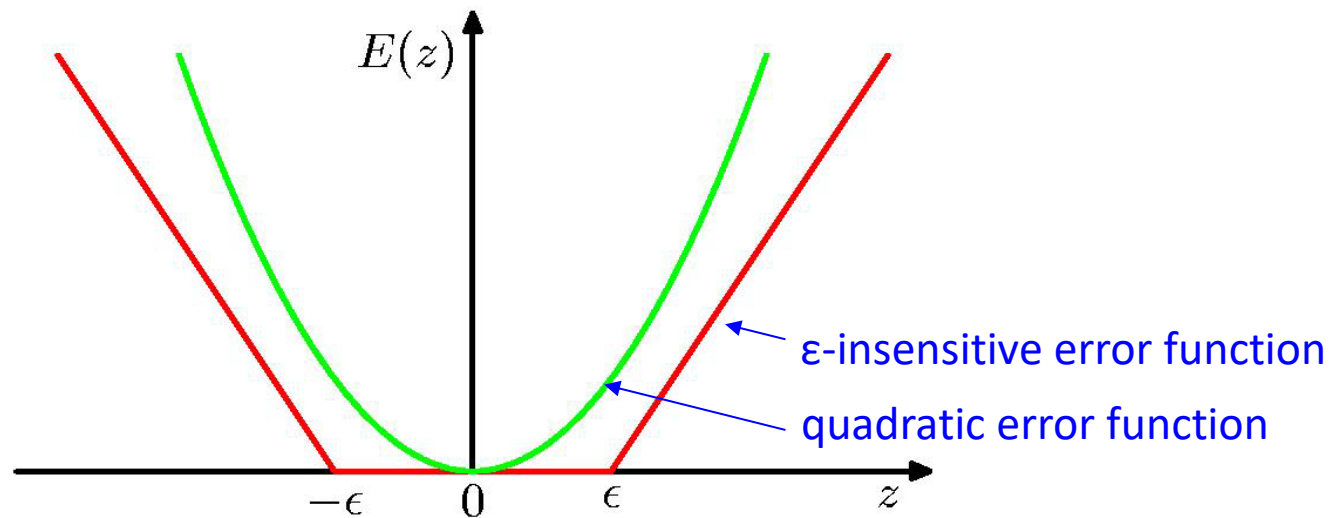
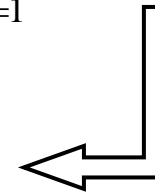
Outlines

- Support Vector Machines
 - SVMs and Logistic Regression
 - SVMs for Regression
 - Relevance Vector Machines
 - RVMs for Regression
 - RVMs for Classification
-

SVMs for Regression I

Simple linear regression: minimize $\frac{1}{2} \sum_{n=1}^N \boxed{\{y_n - t_n\}^2} + \frac{\lambda}{2} \|\mathbf{w}\|^2$
 ϵ -insensitive error function

$$E_{\epsilon}(y(\mathbf{x}) - t) = \begin{cases} 0, & \text{if } |y(\mathbf{x}) - t| < \epsilon \\ |y(\mathbf{x}) - t| - \epsilon, & \text{otherwise} \end{cases}$$



SVMs for Regression II

Minimize

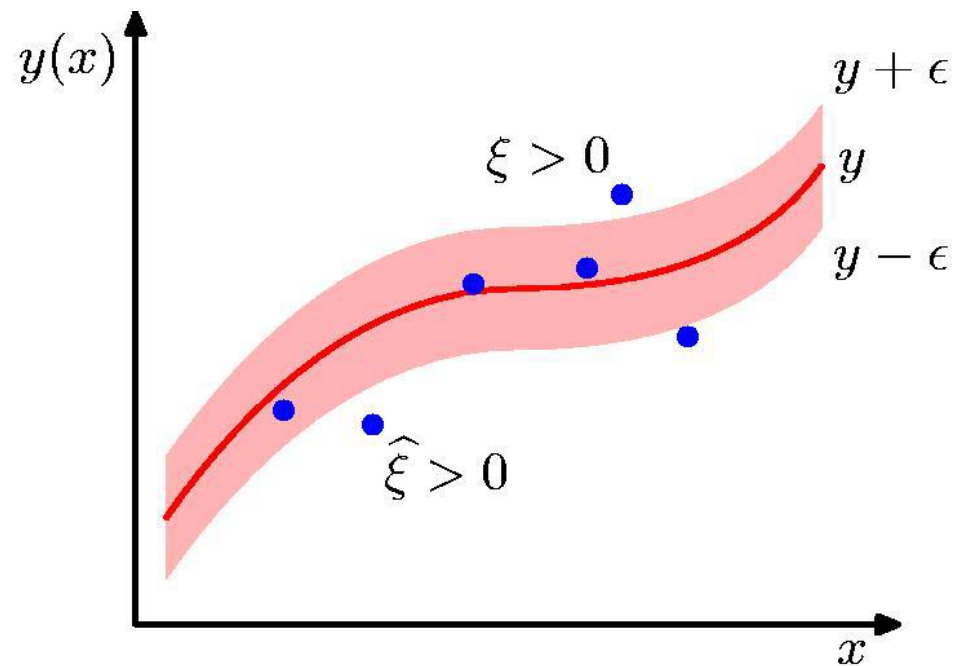
$$C \sum_{n=1}^N E_{\varepsilon}(y(\mathbf{x}_n) - t_n) + \frac{1}{2} \|\mathbf{w}\|^2$$

$$C \sum_{n=1}^N (\xi_n + \hat{\xi}_n) + \frac{1}{2} \|\mathbf{w}\|^2$$

$$\text{where } t_n \leq y(\mathbf{x}_n) + \varepsilon + \xi_n$$

$$t_n \geq y(\mathbf{x}_n) - \varepsilon - \hat{\xi}_n$$

$$\xi_n \geq 0, \hat{\xi}_n \geq 0$$



Dual Problem

$$L = C \sum_{n=1}^N (\xi_n + \hat{\xi}_n) + \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{n=1}^N (\mu_n \xi_n + \hat{\mu}_n \hat{\xi}_n) \\ - \sum_{n=1}^N a_n (\varepsilon + \xi_n + y_n - t_n) - \sum_{n=1}^N \hat{a}_n (\varepsilon + \hat{\xi}_n - y_n + t_n)$$

$$\tilde{L}(\mathbf{a}, \hat{\mathbf{a}}) = -\frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N (a_n - \hat{a}_n)(a_m - \hat{a}_m) k(\mathbf{x}_n, \mathbf{x}_m) \\ - \varepsilon \sum_{n=1}^N (a_n + \hat{a}_n) + \sum_{n=1}^N (a_n - \hat{a}_n) t_n$$

subject to $0 \leq a_n, \hat{a}_n \leq C$

$$\sum_{n=1}^N (a_n - \hat{a}_n) = 0$$

Predictions

$$\mathbf{w} = \sum_{n=1}^N (a_n - \hat{a}_n) \phi(\mathbf{x}_n) \quad (\text{from derivatives of the Lagrangian cost equal 0})$$

$$y(\mathbf{x}) = \sum_{n=1}^N (a_n - \hat{a}_n) k(\mathbf{x}, \mathbf{x}_n) + b$$

KKT conditions:

$$\begin{aligned} a_n (\varepsilon + \xi_n + y_n - t_n) &= 0 \\ \hat{a}_n (\varepsilon + \hat{\xi}_n - y_n + t_n) &= 0 \\ (C - a_n) \xi_n &= 0 \\ (C - \hat{a}_n) \hat{\xi}_n &= 0 \end{aligned}$$

$$b = t_n - \varepsilon - \mathbf{w}^T \phi(\mathbf{x}_n) = t_n - \varepsilon - \sum_{m=1}^N (a_m - \hat{a}_m) k(\mathbf{x}_n, \mathbf{x}_m)$$

Alternative Formulation

ν -SVM (Schölkopf *et al.*, 2000)

$$\begin{aligned}\tilde{L}(\mathbf{a}, \hat{\mathbf{a}}) = & -\frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N (a_n - \hat{a}_n)(a_m - \hat{a}_m) k(\mathbf{x}_n, \mathbf{x}_m) \\ & + \sum_{n=1}^N (a_n - \hat{a}_n) t_n\end{aligned}$$

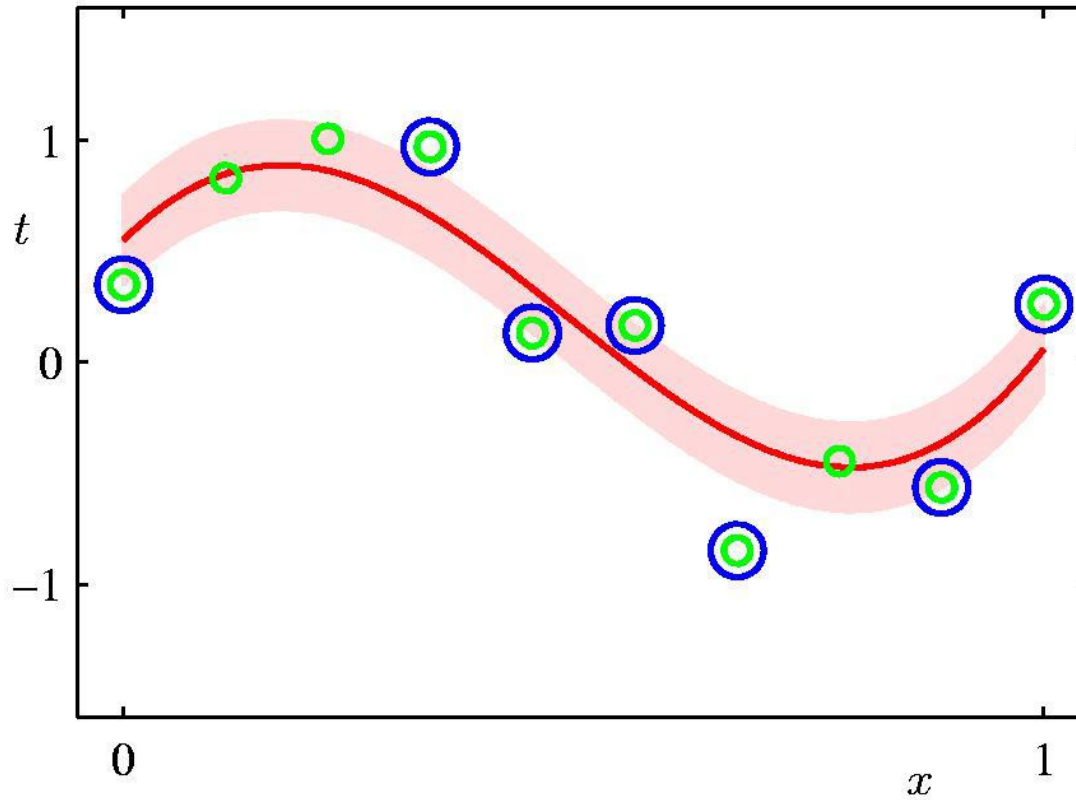
subject to $0 \leq a_n, \hat{a}_n \leq C/N$

$$\sum_{n=1}^N (a_n - \hat{a}_n) = 0$$

$$\sum_{n=1}^N (a_n + \hat{a}_n) \leq \nu C$$

fraction of points lying outside the tube

Example of ν -SVM Regression



Outlines

- Support Vector Machines
 - SVM and Logistic Regression
 - SVM for Regression
 - Relevance Vector Machines
 - RVMs for Regression
 - RVMs for Classification
-

Relevance Vector Machines

□ SVM

- ✓ Outputs are decisions rather than posterior probabilities
- ✓ The extension to $K > 2$ classes is problematic
- ✓ There is a complexity parameter
- ✓ Kernel functions are centered on training data points and required to be positive definite

□ RVM

- ✓ Bayesian regression and classification frameworks
 - ✓ Bayesian sparse kernel technique
 - ✓ Much sparser models
 - ✓ Faster performance on test data
-

Outlines

- Support Vector Machines
 - SVMs and Logistic Regression
 - SVMs for Regression
 - Relevance Vector Machines
 - RVMs for Regression
 - RVMs for Classification
-

RVM for Regression I

□ RVM is a linear form with a modified prior

$$p(t | \mathbf{x}, \mathbf{w}, \beta) = N(t | y(\mathbf{x}), \beta^{-1})$$

$$\text{where } y(\mathbf{x}) = \sum_{i=1}^M w_i \phi_i(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}) \quad \Leftrightarrow \quad y(\mathbf{x}) = \sum_{i=1}^N w_n k(\mathbf{x}, \mathbf{x}_n) + b$$

$$\beta = \sigma^{-2}$$

$$p(\mathbf{t} | \mathbf{X}, \mathbf{w}, \beta) = \prod_{n=1}^N p(t_n | \mathbf{x}_n, \mathbf{w}, \beta^{-1})$$

$$p(\mathbf{w} | \boldsymbol{\alpha}) = \prod_{i=1}^N \mathcal{N}(w_i | 0, \alpha_i^{-1})$$

Each data sample has a weight

RVM for Regression II

$$p(\mathbf{w} | \mathbf{t}, \mathbf{X}, \boldsymbol{\alpha}, \beta) = N(\mathbf{w} | \mathbf{m}, \Sigma)$$

$$\text{where } \mathbf{m} = \beta \Sigma \Phi^T \mathbf{t}$$

$$\Sigma = (\mathbf{A} + \beta \Phi^T \Phi)^{-1}$$

$$\text{where } \Phi : N \times M \text{ matrix with elements } \Phi_{ni} = \phi_i(\mathbf{x}_n)$$

$$\mathbf{A} = \text{diag}(\alpha_i)$$

→ From the result (3.49)
for linear regression models

α and β are determined using *evidence approximation*

$$p(\mathbf{t} | \mathbf{X}, \boldsymbol{\alpha}, \beta) = \int p(\mathbf{t} | \mathbf{X}, \mathbf{w}, \beta) p(\mathbf{w} | \boldsymbol{\alpha}) d\mathbf{w}$$

$$\ln p(\mathbf{t} | \mathbf{X}, \boldsymbol{\alpha}, \beta) = \ln N(\mathbf{t} | \mathbf{0}, \mathbf{C})$$

Prior Predictive Distribution

$$= -\frac{1}{2} \left\{ N \ln(2\pi) + \ln|\mathbf{C}| + \mathbf{t}^T \mathbf{C}^{-1} \mathbf{t} \right\} \quad \Rightarrow \text{Maximize}$$

$$\text{where } \mathbf{t} = (t_1, \dots, t_N)^T, \quad \mathbf{C} = \beta^{-1} \mathbf{I} + \Phi \mathbf{A}^{-1} \Phi^T$$

RVM for Regression III

□ Two steps

- ① From derivatives of the marginal likelihood, we have

$$\alpha_i^{new} = \frac{\gamma_i}{m_i^2}, \quad (\beta^{new})^{-1} = \frac{\|\mathbf{t} - \Phi \mathbf{m}\|^2}{N - \sum_i \gamma_i}$$

where $\gamma_i = 1 - \alpha_i \sum_{ii}$

\sum_{ii} : i^{th} diagonal element of Σ

- ② Predictive distribution

Posterior Predictive Distribution

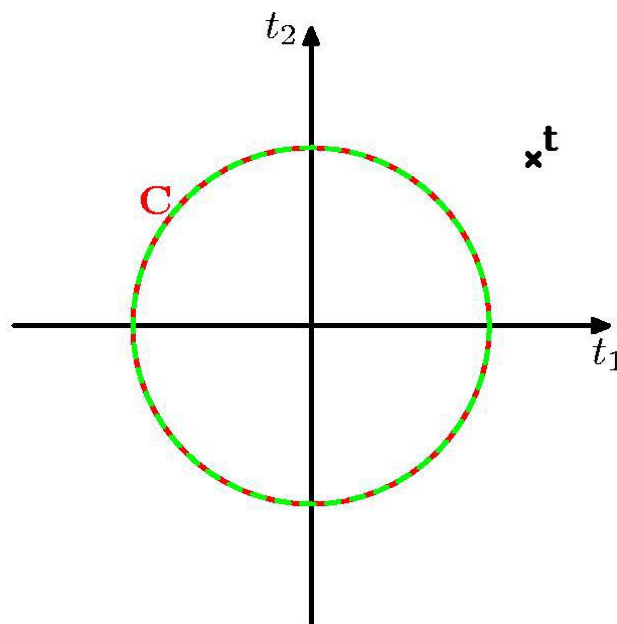
$$\begin{aligned} p(t | \mathbf{x}, \mathbf{X}, \mathbf{t}, \boldsymbol{\alpha}^*, \beta^*) &= \int p(t | \mathbf{x}, \mathbf{w}, \beta^*) p(\mathbf{w} | \mathbf{x}, \mathbf{X}, \mathbf{t}, \boldsymbol{\alpha}^*, \beta^*) d\mathbf{w} \\ &= N(t | \mathbf{m}^T \phi(\mathbf{x}), \sigma^2(\mathbf{x})) \end{aligned}$$

where $\sigma^2(\mathbf{x}) = (\beta^*)^{-1} + \phi(\mathbf{x})^T \Sigma \phi(\mathbf{x})$

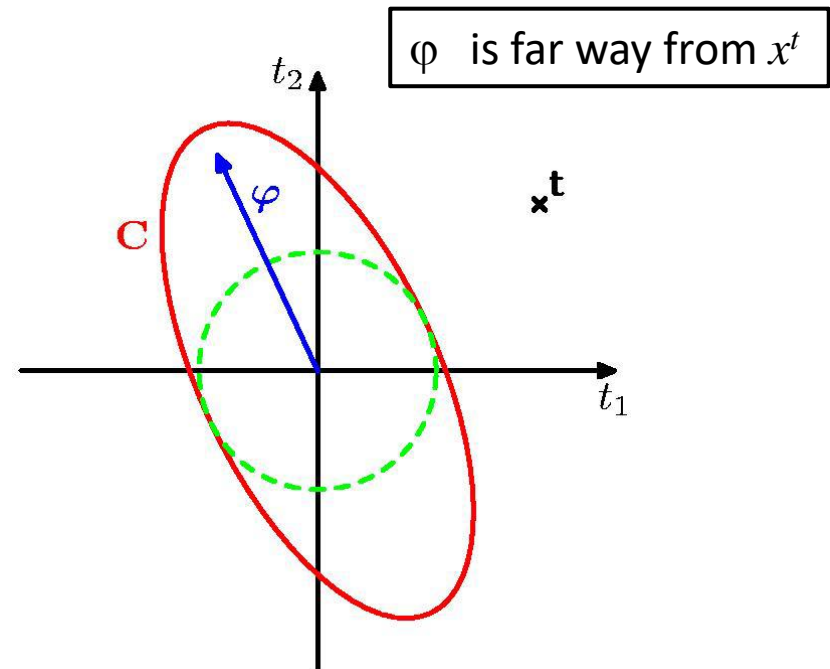
Mechanism for Sparsity

$$p(\mathbf{t} \mid \alpha, \beta) = N(\mathbf{t} \mid \mathbf{0}, \mathbf{C})$$

$$\text{where } \mathbf{t} = (t_1, t_2)^T, \quad \mathbf{C} = \beta^{-1} \mathbf{I} + \alpha^{-1} \boldsymbol{\varphi} \boldsymbol{\varphi}^T$$



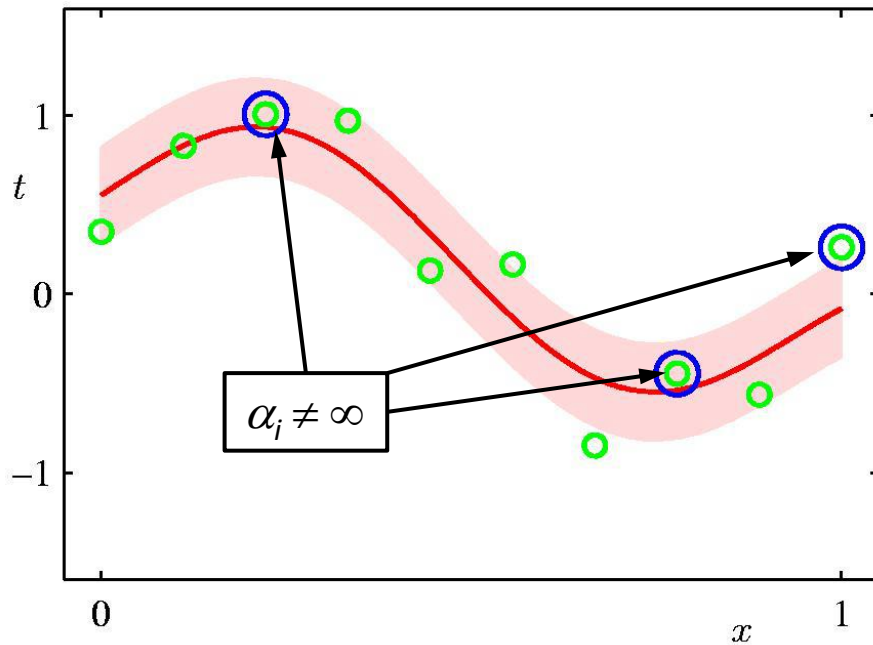
only isotropic noise, $\alpha = \infty$



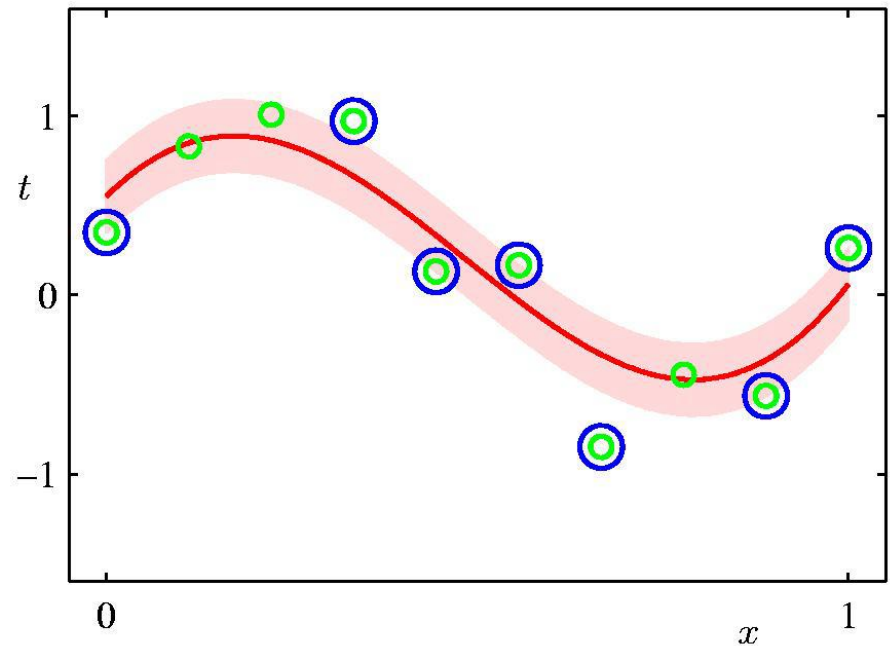
a finite value of α

Examples of RVM Regression

RVM regression



ν -SVM regression



More compact than SVM (3 relevance vectors v.s. 7 support vectors)

Parameters are determined automatically

Require more training time than SVM

Sparse Solution I

Pull out the contribution from α_i in

$$\mathbf{C} = \beta^{-1} \mathbf{I} + \Phi \mathbf{A}^{-1} \Phi^T$$

$$\begin{aligned} \mathbf{C} &= \beta^{-1} \mathbf{I} + \sum_{j \neq i} \alpha_j^{-1} \varphi_j \varphi_j^T + \alpha_i^{-1} \varphi_i \varphi_i^T \\ &= \mathbf{C}_{-i} + \alpha_i^{-1} \varphi_i \varphi_i^T \end{aligned}$$

where φ_i : i th column of Φ

$$|\mathbf{C}| = |\mathbf{C}_{-i}| \left| 1 + \alpha_i^{-1} \varphi_i^T \mathbf{C}_{-i}^{-1} \varphi_i \right|$$

→ Using (C.7), (C.15) in Appendix C

$$\mathbf{C}^{-1} = \mathbf{C}_{-i}^{-1} - \frac{\mathbf{C}_{-i}^{-1} \varphi_i \varphi_i^T \mathbf{C}_{-i}^{-1}}{\alpha_i + \varphi_i^T \mathbf{C}_{-i}^{-1} \varphi_i}$$

Sparse Solution II

□ Then log marginal likelihood function L becomes,

$$L(\boldsymbol{\alpha}) = L(\boldsymbol{\alpha}_{-i}) + \lambda(\alpha_i)$$

$L(\boldsymbol{\alpha}_{-i})$: omitting α_i

$$\lambda(\alpha_i) = \frac{1}{2} \left[\ln \alpha_i - \ln(\alpha_i + s_i) + \frac{q_i^2}{\alpha_i + s_i} \right]$$

where $s_i = \boldsymbol{\varphi}_i^T \mathbf{C}_{-i}^{-1} \boldsymbol{\varphi}_i$

$$q_i = \boldsymbol{\varphi}_i^T \mathbf{C}_{-i}^{-1} \mathbf{t}$$

→ Sparsity: measures the extent to which $\boldsymbol{\varphi}_i$ overlaps with the other basis vectors

→ Quality of $\boldsymbol{\varphi}_i$: represents a measure of the alignment of the basis vector with the error between \mathbf{t} and \mathbf{y}_{-i}

□ Stationary points of the marginal likelihood w.r.t. α_i

$$\Rightarrow \frac{d\lambda(\alpha_i)}{d\alpha_i} = \frac{\alpha_i^{-1} s_i^2 - (q_i^2 - s_i)}{2(\alpha_i + s_i)^2} = 0$$

Sequential Sparse Bayesian Learning

1. Initialize β
2. Initialize using φ_1 , with $\alpha_1 = s_1^2 / (q_1^2 - s_1)$, with the remaining $\alpha_{j(j \neq 1)} = \infty$
3. Evaluate Σ and \mathbf{m} for all basis functions
4. Select a candidate φ_i
5. If $q_i^2 > s_i$, $\alpha_i < \infty$ (φ_i is already in the model), update $\alpha_i = s_i^2 / (q_i^2 - s_i)$
6. If $q_i^2 > s_i$, $\alpha_i = \infty$, add φ_i to the model, and evaluate $\alpha_i = s_i^2 / (q_i^2 - s_i)$
7. If $q_i^2 \leq s_i$, $\alpha_i < \infty$, remove φ_i from the model, and set $\alpha_i = \infty$
8. Update β
9. Go to 3 until converged

Outlines

- Support Vector Machines
 - SVMs and Logistic Regression
 - SVMs for Regression
 - Relevance Vector Machines
 - RVMs for Regression
 - RVMs for Classification
-

RVM for Classification

- Probabilistic linear classification model with Gaussian prior

$$y(\mathbf{x}, \mathbf{w}) = \sigma(\mathbf{w}^T \phi(\mathbf{x})) \quad p(\mathbf{w} | \boldsymbol{\alpha}) = \prod_{n=1}^N N(w_n | 0, \alpha_n^{-1})$$

- Initialize $\boldsymbol{\alpha}$
 - Build a Gaussian approximation to the posterior distribution
 - Obtain an approximation to the marginal likelihood
 - Maximize the marginal likelihood (re-estimate $\boldsymbol{\alpha}$) until converged
-

RVM for Classification (Cont'd)

□ The posterior distribution is obtained by maximizing

$$\begin{aligned}\ln p(\mathbf{w} | \mathbf{t}, \boldsymbol{\alpha}) &= \ln \{p(\mathbf{t} | \mathbf{w}) p(\mathbf{w} | \boldsymbol{\alpha})\} - \ln p(\mathbf{t} | \boldsymbol{\alpha}) \\ &= \sum_{n=1}^N \{t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\} - \frac{1}{2} \mathbf{w}^T \mathbf{A} \mathbf{w} + \text{const}\end{aligned}$$

where $\mathbf{A} = \text{diag}(\alpha_i)$

⇒ Iterative reweighted least squares (IRLS)

$$\nabla \ln p(\mathbf{w} | \mathbf{t}, \boldsymbol{\alpha}) = \boldsymbol{\Phi}^T (\mathbf{t} - \mathbf{y}) - \mathbf{A} \mathbf{w}$$

$$\nabla \nabla \ln p(\mathbf{w} | \mathbf{t}, \boldsymbol{\alpha}) = -(\boldsymbol{\Phi}^T \mathbf{B} \boldsymbol{\Phi} + \mathbf{A})$$

where $\mathbf{B} : N \times N$ diagonal matrix, $b_n = y_n(1 - y_n)$,

$\boldsymbol{\Phi}$: design matrix, $\Phi_{ni} = \phi_i(\mathbf{x}_n)$

⇒ Resulting Gaussian approximation to the posterior distribution

$$\mathbf{w}^* = \mathbf{A}^{-1} \boldsymbol{\Phi}^T (\mathbf{t} - \mathbf{y}), \quad \Sigma = (\boldsymbol{\Phi}^T \mathbf{B} \boldsymbol{\Phi} + \mathbf{A})^{-1} \quad \Leftarrow \nabla \ln p(\mathbf{w} | \mathbf{t}, \boldsymbol{\alpha}) = 0$$

RVM for Classification (Cont'd)

- Marginal likelihood using Laplace approximation

$$\begin{aligned} p(\mathbf{t} | \boldsymbol{\alpha}) &= \int p(\mathbf{t} | \mathbf{w}) p(\mathbf{w} | \boldsymbol{\alpha}) d\mathbf{w} \\ &= p(\mathbf{t} | \mathbf{w}^*) p(\mathbf{w}^* | \boldsymbol{\alpha}) (2\pi)^{M/2} |\Sigma|^{1/2} \end{aligned}$$

- Set the derivative of the marginal likelihood equal to zero, and rearranging then gives

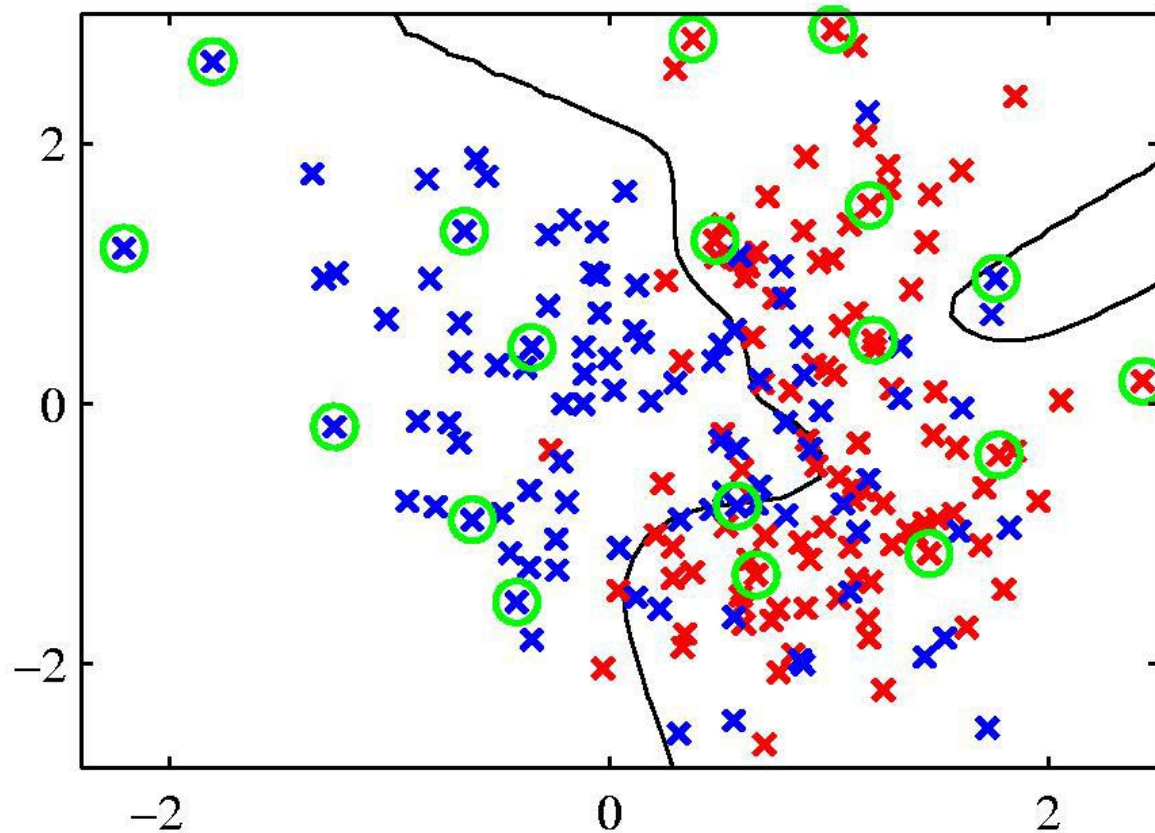
If we define $\hat{\mathbf{t}} = \Phi \mathbf{w}^* + \mathbf{B}^{-1}(\mathbf{t} - \mathbf{y})$

$$\alpha_i^{new} = \frac{\gamma_i}{(w_i^*)^2} \quad \text{where } \gamma_i = 1 - \alpha_i \sum_{ii}$$

$$\ln p(\mathbf{t} | \boldsymbol{\alpha}, \beta) = -\frac{1}{2} \left\{ N \ln(2\pi) + \ln |\mathbf{C}| + (\hat{\mathbf{t}})^T \mathbf{C}^{-1} \hat{\mathbf{t}} \right\} \Rightarrow \boxed{\text{Same in the regression case}}$$

where $\mathbf{C} = \mathbf{B} + \Phi \mathbf{A} \Phi^T$

Example of RVM Classification



Summary

- Support Vector Machines
 - SVMs and Logistic Regression
 - SVMs for Regression
 - Relevance Vector Machines
 - RVMs for Regression
 - RVMs for Classification
-