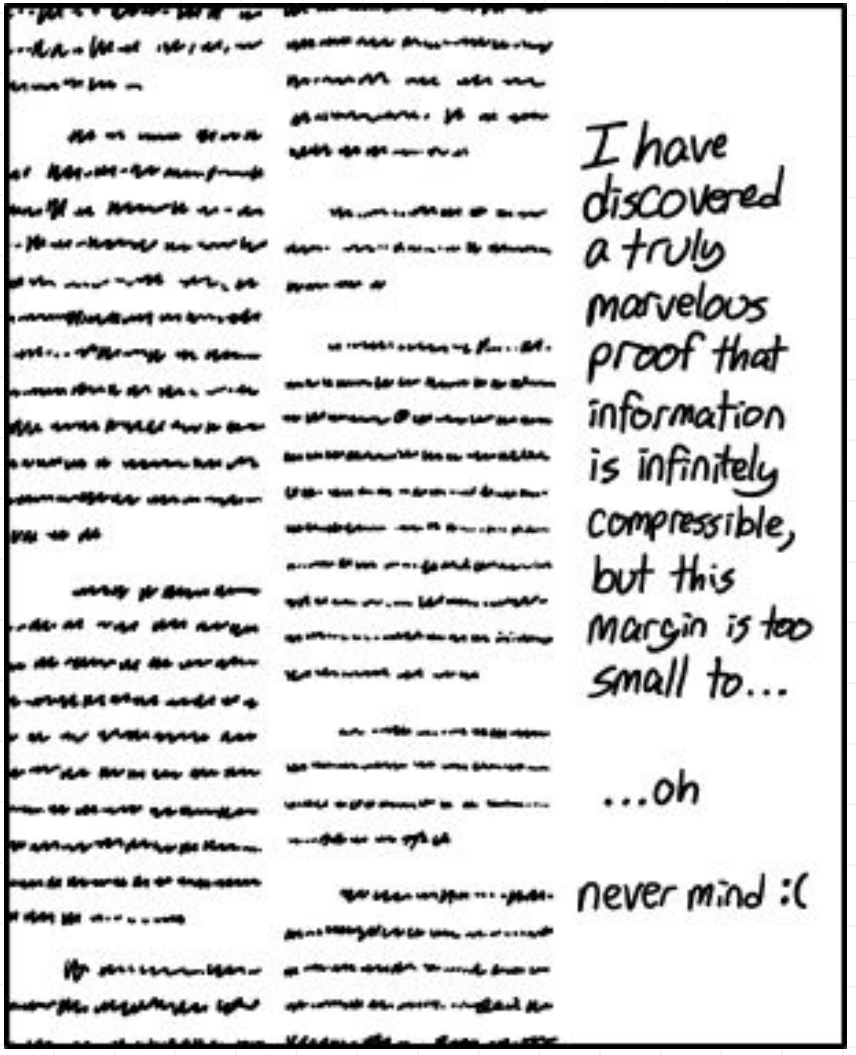


https://xkcd.com/1381/

Not the same kind of margin either ... :)

Announcements

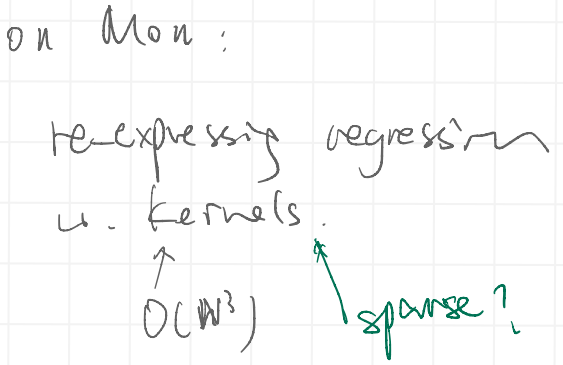
BREAK COMING UP :)

Watch out for survey

Week 7 Mon – Easter

Week 8 Mon – ANZAC Day

Kernel machines



Lagrange multipliers, take 2

Maximum-margin classifiers (Chap 7.1)

- The intuition of margins
- Constructing the max-margin classifier
- The dual representation (cultural exposure)
- Support vectors and their geometric intuitions

SVM for overlapping class distributions

Relations to logistic regression

SVM for regression

The kernel trick.
More exposure to optimisation (lagrangian multipliers, KKT conditions, transforming a problem ...)

From kernels to sparse kernel machines

- Nonlinear kernels extended our toolbox of methods considerably
  - Wherever an inner product was used in an algorithm, we can replace it with a kernel.
  - Kernels act as a kind of 'similarity' measure and can be defined over graphs, sets, strings, and documents.
- But the kernel matrix is a square matrix with dimensions equal to the number of data points N
  - In order to calculate it, the kernel function must be evaluated for all pairs of training inputs.
- Sparse Kernel Machines implement learning algorithms where, for prediction, the kernels are only evaluated at a subset of the training data.
- Today we introduce the famous Support Vector Machine — a non-probabilistic, non-parametric classifier, related to Kernel Logistic Regression
  - partially alleviates the time complexity problems, but in general approximations are needed for a scalable algorithm.

For linear regression, we go from

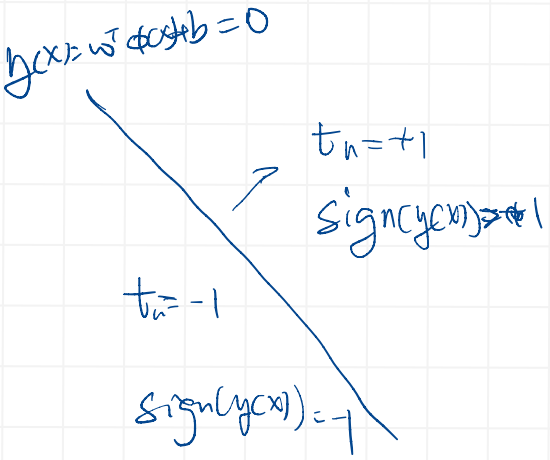
$$y(\mathbf{x}) = \boldsymbol{\phi}(\mathbf{x})^\top (\lambda \mathbf{I} + \boldsymbol{\Phi}^\top \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^\top \mathbf{t}$$

to

$$y(\mathbf{x}) = \mathbf{k}(\mathbf{x})^\top (\mathbf{K} + \lambda \mathbf{I}_N)^{-1} \mathbf{t}$$

Separating two classes with a linear model

$$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}) + b \qquad (7.1)$$



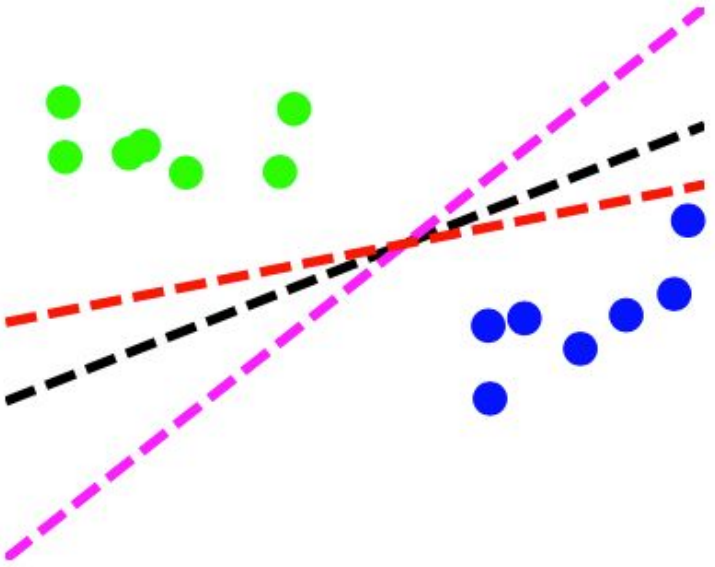
- Training data are N input vectors $\mathbf{x}_1, \ldots, \mathbf{x}_N$ with corresponding targets $t_1, \ldots, t_N$ where $t_n \in \{-1, +1\}$.
- The class of a new point is predicted as $\mathrm{sign}\,(y(\mathbf{x}))$.

Assume: training dataset linearly separable, i.e. there exist **w** and b such that:

$$t_n\ \mathrm{sign}\,(y(\mathbf{x}_n)) > 0 \qquad n = 1, \ldots, N.$$

The multiple separator problem

There may exist many solutions $\mathbf{w}$ and b for which the linear classifier perfectly separates the two classes!



"far-away" form either class seems better

The perceptron algorithm can find a solution, but this depends on the initial choice of parameters.

What is the decision boundary which results in the best generalisation (smallest generalisation error)?

Maximum margin classifier

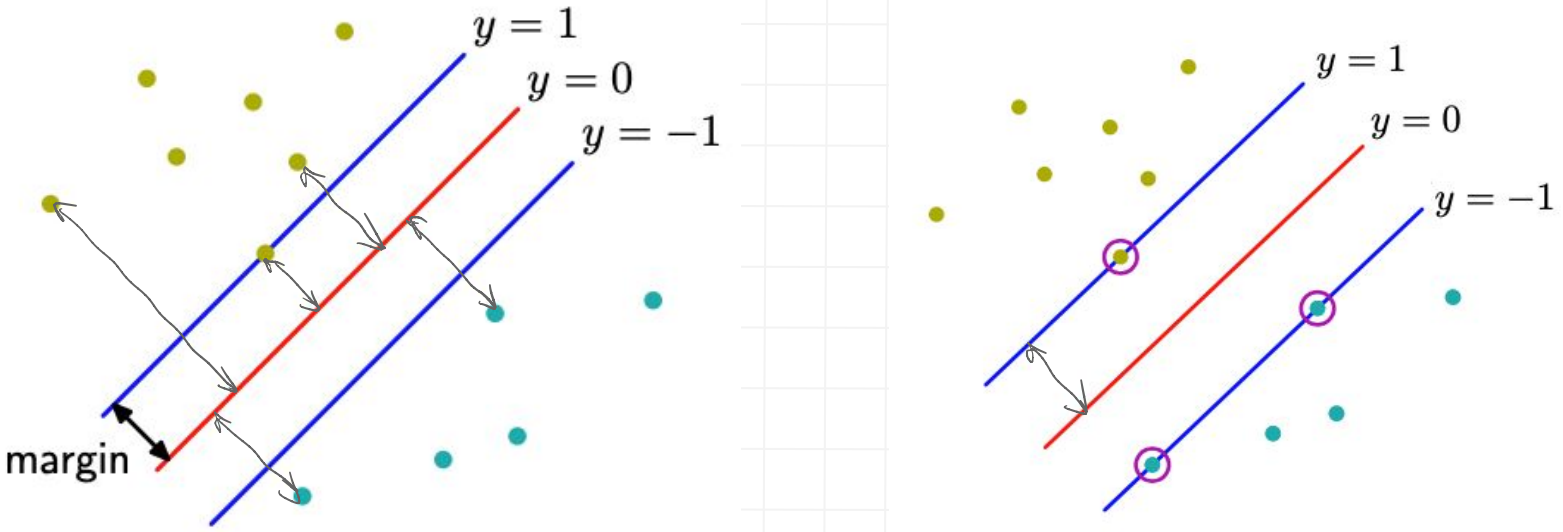


Figure 7.1 The margin is defined as the perpendicular distance between the decision boundary and the closest of the data points, as shown on the left figure. Maximizing the margin leads to a particular choice of decision boundary, as shown on the right. The location of this boundary is determined by a subset of the data points, known as support vectors, which are indicated by the circles.

Figure incomplete in pdf version of the book.

Flashback: Discriminant for two classes

$$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0 \tag{4.4}$$

$y(\mathbf{x})$ gives a **signed** measure of the perpendicular distance r **from** the decision surface **to** $\mathbf{x}$, that is $r = y(\mathbf{x})/\|w\|$.

$$y(\mathbf{x}) = \mathbf{w}^T\left(\overbrace{\mathbf{x}_\perp + r\frac{\mathbf{w}}{\|w\|}}^{\mathbf{x}}\right) + w_0 = r\frac{\mathbf{w}^\top\mathbf{w}}{\|w\|} + \overbrace{\mathbf{w}^\top\mathbf{x}_\perp + w_0}^{0} = r\|w\|$$

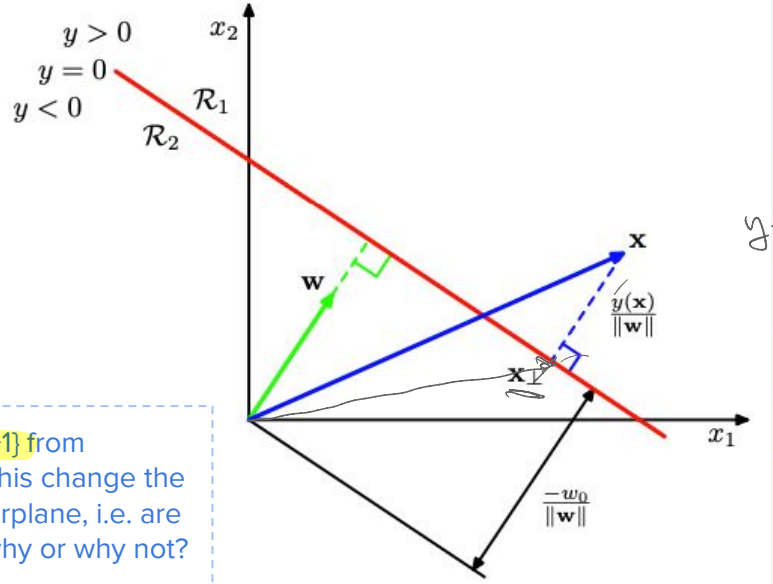
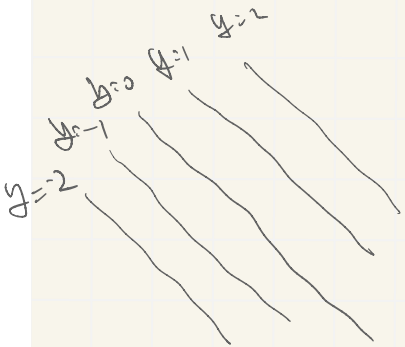


Figure 4.1 Illustration of the geometry of a linear discriminant function in two dimensions. The decision surface, shown in red, is perpendicular to $\mathbf{w}$, and its displacement from the origin is controlled by the bias parameter w_0. Also, the signed orthogonal distance of a general point $\mathbf{x}$ from the decision surface is given by $y(\mathbf{x})/\|\mathbf{w}\|$.

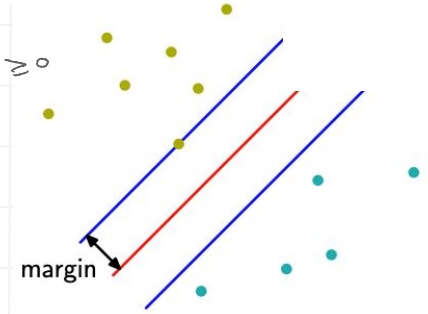


Q: The target encoding is changed to {-1, +1} from one-of-K encoding from Chapter 4. Does this change the meaning and value of the separating hyperplane, i.e. are the two classes still separated by y(x)=0, why or why not?

opt. problem

max f(x)
s.t g(x) ≥ 0

Maximum margin solution: objective function



$$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}) + b \tag{7.1}$$

Separating hyperplane: y(x) = 0

Distance of y(x) from they separating hyperplane y(x)=0: $|y(\mathbf{x})|/\|\mathbf{w}\|.$

$$\frac{t_n y(\mathbf{x}_n)}{\|\mathbf{w}\|} = \frac{t_n(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n) + b)}{\|\mathbf{w}\|}. \tag{7.2}$$

$t_n \operatorname{sgn}(y(x)) > 0$

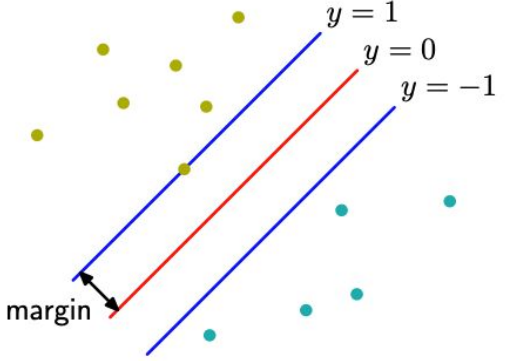
unsigned distance.

Solve the following to obtain maximum margin solution:

$$\arg\max_{\mathbf{w},b} \left\{ \frac{1}{\|\mathbf{w}\|} \min_n \left[t_n \left(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n) + b \right) \right] \right\} \tag{7.3}$$

$\mathbf{w}$ unconstrained $\in \mathbb{R}^p$.

Normalising the margin



non-linear obj in $\mathbf{w}$

$$\arg\max_{\mathbf{w},b} \left\{ \frac{1}{\|\mathbf{w}\|} \min_n \left[t_n \left(\mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_n) + b \right) \right] \right\} \qquad (7.3)$$

Rescaling doesn't change the margin of each data point $\{\mathbf{x}_n, t_n\}$:

If $\mathbf{w} \to \kappa\mathbf{w}$ and $b \to \kappa b$, $t_n y(\mathbf{x}_n)/\|\mathbf{w}\|$ unchanged.

For points closest to decision boundary, set:

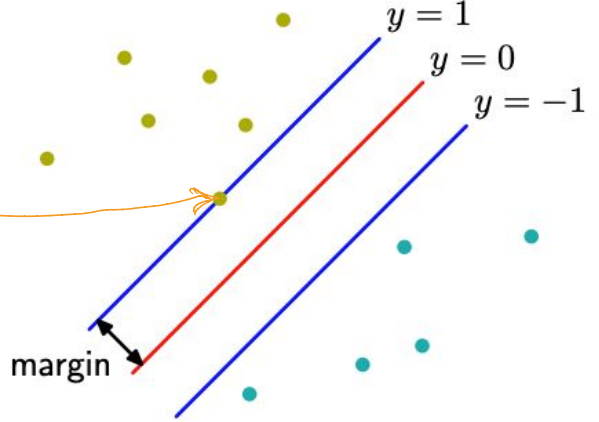
$$t_n \left(\mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_n) + b \right) = 1 \qquad (7.4)$$

This implies that all data points lie "outside the margin"

$$t_n \left(\mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_n) + b \right) \geqslant 1, \qquad n = 1, \ldots, N. \qquad (7.5)$$

→ linear constraints in $\mathbf{w}$.

Equivalent objective function



from last page,

$$\arg\max_{\mathbf{w},b}\left\{\frac{1}{\|\mathbf{w}\|}\min_{n}\left[t_n\left(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)+b\right)\right]\right\} \tag{7.3}$$

"the point"

For points closest to decision boundary, set:

$$t_n\left(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)+b\right)=1 \tag{7.4}$$

margin $\frac{t(w^T\phi(x)+b)}{\|w\|}$

This implies that all data points lie "on or outside the margin"

$$t_n\left(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)+b\right)\geqslant 1, \qquad n=1,\ldots,N. \tag{7.5}$$

$t\,\mathrm{sign}(y(x)) = 1$

$+1/-1$ & $+1/-1$

(7.4 holds)

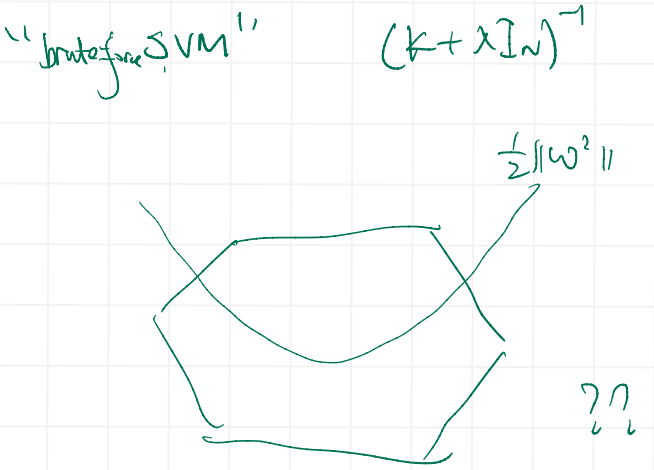
there is at least one constraint active, where $t_n y(x_n)=1$ → maximize $\|\mathbf{w}\|^{-1}$

$$\arg\min_{\mathbf{w},b}\frac{1}{2}\|\mathbf{w}\|^2 \tag{7.6}$$

min $\|w\|$

quadratic, obj.

Outline



Lagrange multipliers, take 2

Maximum-margin classifiers (Chap 7.1)

- The intuition of margins
- Constructing the max-margin classifier
- The dual representation
- Support vectors and their geometric intuitions

SVM for overlapping class distributions

Relations to logistic regression

SVM for regression

Flashback: Lagrange multipliers (appendix E)

The first encounter in SML - we'll see it again in kernel methods.

objective function
equality constraint

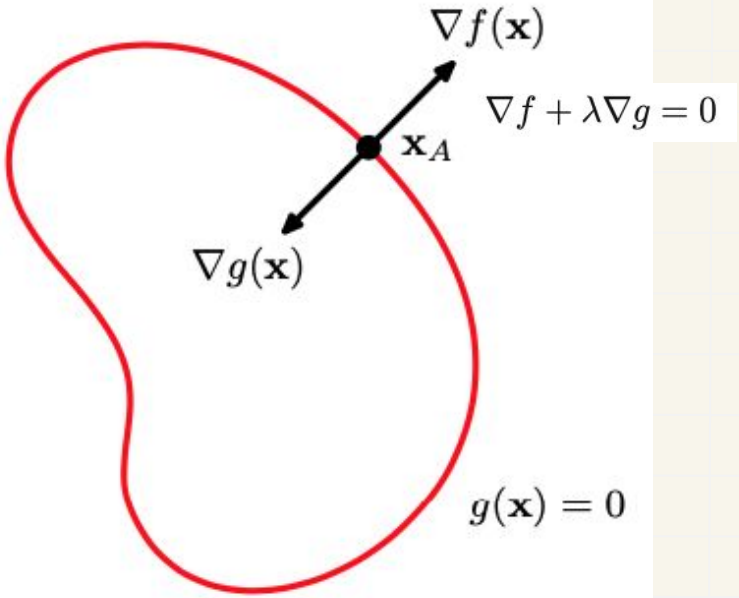
maximize f(x)
subject to g(x)=0

$$\mathcal{L}(x, \lambda) = f(x) + \lambda g(x)$$

$\mathcal{L}$ = Lagrangian

λ = Lagrange multiplier

Figure E.1 A geometrical picture of the technique of Lagrange multipliers in which we seek to maximize a function $f(\mathbf{x})$, subject to the constraint $g(\mathbf{x}) = 0$. If $\mathbf{x}$ is D dimensional, the constraint $g(\mathbf{x}) = 0$ corresponds to a subspace of dimensionality $D - 1$, indicated by the red curve. The problem can be solved by optimizing the Lagrangian function $L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \lambda g(\mathbf{x})$.



Lagrange multipliers with inequality constraints

objective function
inequality constraint

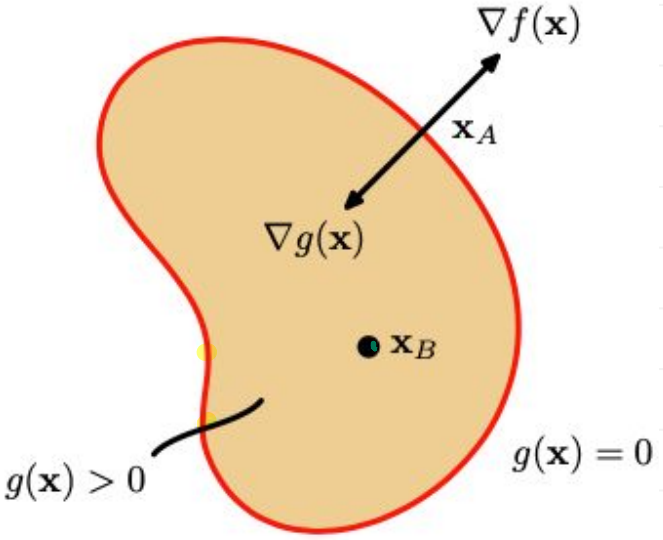
maximize f(x)
subject to g(x) >= 0

max. $L(x, \lambda) = f(x) + \lambda g(x)$

Karush-Kuhn-Tucker (KKT) conditions

$$g(\mathbf{x}) \geqslant 0 \quad \text{(E.9)}$$
$$\lambda \geqslant 0 \quad \text{(E.10)}$$
$$\lambda g(\mathbf{x}) = 0 \quad \text{(E.11)}$$

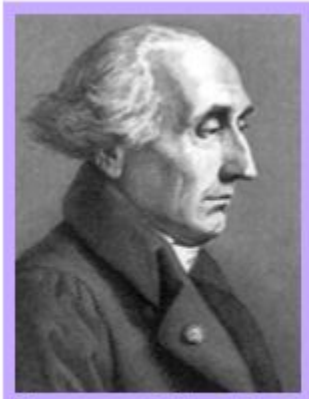
Figure E.3 Illustration of the problem of maximizing $f(\mathbf{x})$ subject to the inequality constraint $g(\mathbf{x}) \geqslant 0$.



two cases (stationary point of $L(x,\lambda)$)

Constraint *inactive*: g(x)>0 → $\nabla f(x) = 0, \ \lambda = 0$

Constraint *active*: g(x)=0 → $\nabla f(x) = -\lambda \nabla g(x), \ \lambda > 0$



Joseph-Louis Lagrange
1736–1813

Although widely considered to be a French mathematician, Lagrange was born in Turin in Italy. By the age of nineteen, he had already made important contributions mathematics and had been appointed as Professor at the Royal Artillery School in Turin. For many years, Euler worked hard to persuade Lagrange to move to Berlin, which he eventually did in 1766 where he succeeded Euler as Director of Mathematics at the Berlin Academy. Later he moved to Paris, narrowly escaping with his life during the French revolution thanks to the personal intervention of Lavoisier (the French chemist who discovered oxygen) who himself was later executed at the guillotine. Lagrange made key contributions to the calculus of variations and the foundations of dynamics.

The KKT conditions were originally named after Harold W. Kuhn and Albert W. Tucker, who first published the conditions in 1951. Later scholars discovered that the necessary conditions for this problem had been stated by William Karush in his master's thesis in 1939.
https://en.wikipedia.org/wiki/Karush%E2%80%93Kuhn%E2%80%93Tucker_conditions

The max-margin problem and its Lagrangian (primal)

$$\arg\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \qquad (7.6)$$

s.t. $$t_n\left(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)+b\right) \geqslant 1, \qquad n=1,\ldots,N. \qquad (7.5)$$

The Lagrangian, define one a_n >=0 for each constraint in (7.5)

$$L(\mathbf{w},b,\mathbf{a}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N} a_n\left\{t_n(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)+b)-1\right\} \qquad (7.7)$$

Set its derivative w.r.t **w** and b to 0

$$\mathbf{w} = \sum_{n=1}^{N} a_n t_n \boldsymbol{\phi}(\mathbf{x}_n) \qquad (7.8)$$

$$0 = \sum_{n=1}^{N} a_n t_n. \qquad (7.9)$$

← $\{a_n\}$ are still left!

The dual representation

→ opt problem expressed in Lagrange multipliers

Here: cultural exposure

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N} a_n \left\{ t_n(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n) + b) - 1 \right\} \tag{7.7}$$

Set the derivative of L w.r.t. **w** and b to 0

$$\mathbf{w} = \sum_{n=1}^{N} a_n t_n \boldsymbol{\phi}(\mathbf{x}_n) \tag{7.8}$$

$$0 = \sum_{n=1}^{N} a_n t_n. \tag{7.9}$$

from last page.

Substitute **w** with (7.8) and use (7.9) - see notes.

$$\widetilde{L}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m) \tag{7.10}$$

→ quadratic

training data.

Hope: lots of $a_n = 0$ → sparsity.

subject to the constraints

$$a_n \geqslant 0, \qquad n = 1, \ldots, N, \tag{7.11}$$

$$\sum_{n=1}^{N} a_n t_n = 0. \tag{7.12}$$

total weights of class +1 & class −1 are equal.

Quadratic program: optimize a quadratic function of **a** subject to a set of linear inequality constraints.

$$\widetilde{L}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m) \tag{7.10}$$

subject to the constraints

$$a_n \geqslant 0, \qquad n = 1, \ldots, N, \tag{7.11}$$

$$\sum_{n=1}^{N} a_n t_n = 0. \tag{7.12}$$

obtained using QP solver
↓ OR customized SVM solver

Assume we have solutions for **a**, now solve for b

$$t_n \left(\sum_{m \in \mathcal{S}} a_m t_m k(\mathbf{x}_n, \mathbf{x}_m) + b \right) = 1 \tag{7.17}$$

$$b = \frac{1}{N_{\mathcal{S}}} \sum_{n \in \mathcal{S}} \left(t_n - \sum_{m \in \mathcal{S}} a_m t_m k(\mathbf{x}_n, \mathbf{x}_m) \right) \tag{7.18}$$

SVM derivation - dual form

$$L(\vec{w}, b, \vec{a}) = \frac{1}{2}\|w\|^2 - \sum_{n=1}^{N} a_n \{ t_n(w^T\phi(x_n) + b) - 1 \} \quad (7.7)$$

$$\downarrow \quad \vec{w} = \sum_{n=1}^{N} a_n t_n \phi(x_n) \qquad \frac{\partial L}{\partial w} = w - \sum_{n=1}^{N} a_n t_n \phi(x_n) = 0 \quad (7.8)$$

$$L(b, \vec{a}) = \frac{1}{2} \sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m \phi(x_n)^T \phi(x_m)$$

$$- \sum_{n=1}^{N} a_n \{ t_n \sum_{m=1}^{N} a_m t_m \phi(x_m)^T \phi(x_n) + t_n b - 1 \} \quad (7.9)$$

$$= -\frac{1}{2} \sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m \phi^T(x_n)\phi(x_m) - \left(\sum_{n=1}^{N} a_n t_n\right) b + \sum_{n=1}^{N} a_n \quad (\text{where } \sum_{n=1}^{N} a_n t_n = 0)$$

$$= \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m k(x_n, x_m) \quad (7.10)$$

skipped, left 4 reading

KKT conditions → support vectors

Make predictions via:

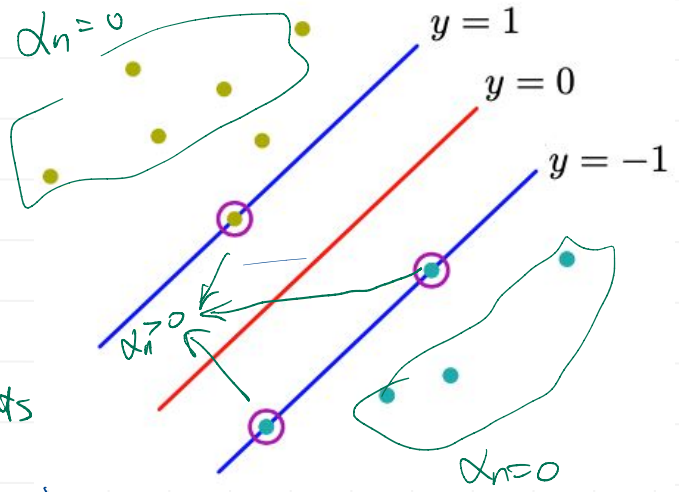
$$\mathbf{w} = \sum_{n=1}^{N} a_n t_n \phi(\mathbf{x}_n) \qquad (7.8)$$

kernel trick

$$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}) + b = \sum_{n=1}^{N} a_n t_n k(\mathbf{x}, \mathbf{x}_n) + b \qquad (7.13)$$

↑ ignore training points w $\alpha_n = 0$:)

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N} a_n \left\{ t_n(\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n) + b) - 1 \right\} \qquad (7.7)$$



in this example, only 3 $\alpha_n > 0$:)

KKT conditions

$$a_n \geqslant 0 \qquad (7.14)$$

$$t_n y(\mathbf{x}_n) - 1 \geqslant 0 \qquad (7.15)$$

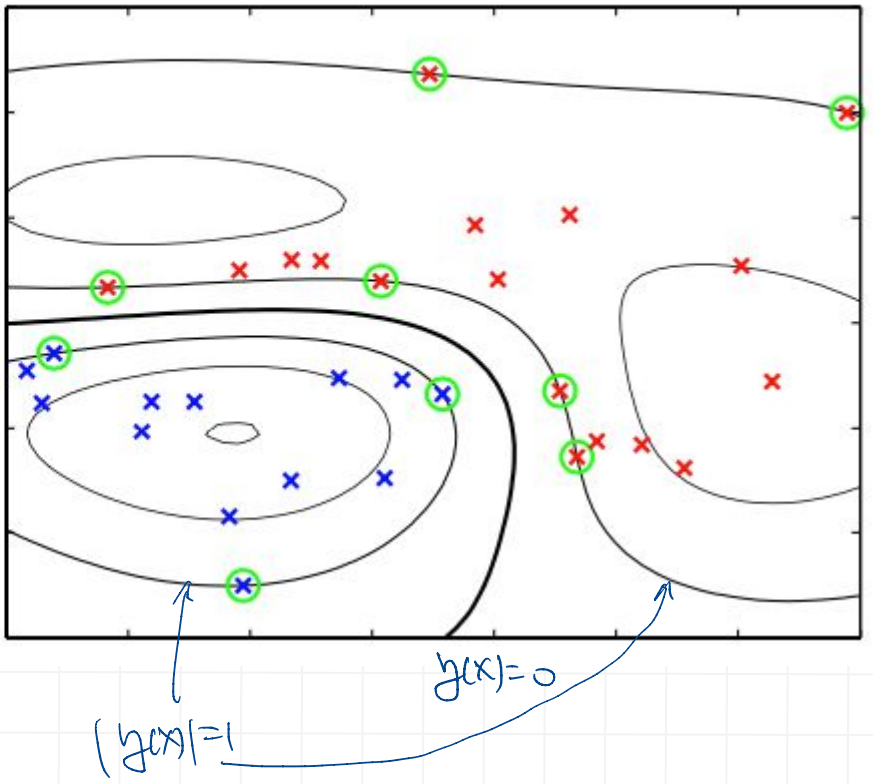
$$a_n \left\{ t_n y(\mathbf{x}_n) - 1 \right\} = 0. \qquad (7.16)$$

either α_n or $t_n(y(t_n)) - 1 = 0$

nice theoretical properties via VC dimensions.

Geometric insight for sparsity

Figure 7.2 Example of synthetic data from two classes in two dimensions showing contours of constant $y(\mathbf{x})$ obtained from a support vector machine having a Gaussian kernel function. Also shown are the decision boundary, the margin boundaries, and the support vectors.



Outline

Lagrange multipliers, take 2

Maximum-margin classifiers (Chap 7.1)

- The intuition of margins
- Constructing the max-margin classifier
- The dual representation
- Support vectors and their geometric intuitions ← KKT conditions.

SVM for overlapping class distributions

Relations to logistic regression

SVM for regression

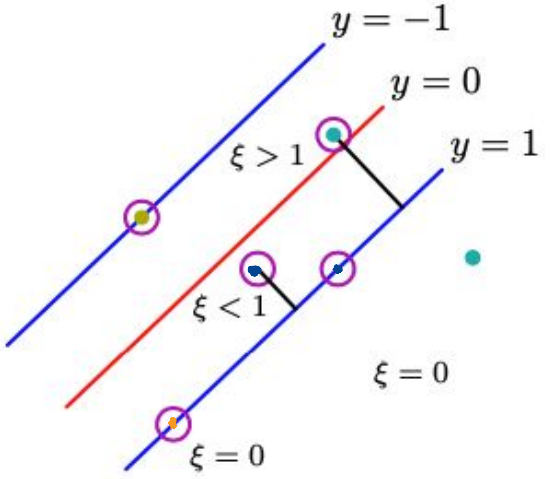
What happens if the classes overlap?

$$\arg\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \qquad (7.6)$$

s.t. $t_n\left(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)+b\right) \geqslant 1, \qquad n=1,\ldots,N. \qquad (7.5)$

$$\sum_{n=1}^{N} E_\infty(y(\mathbf{x}_n)t_n - 1) + \lambda\|\mathbf{w}\|^2 \qquad (7.19)$$

Figure 7.3 Illustration of the slack variables $\xi_n \geqslant 0$. Data points with circles around them are support vectors.



First, replace the constraints as

$$t_n y(\mathbf{x}_n) \geqslant 1 - \xi_n, \qquad n=1,\ldots,N \qquad (7.20)$$

- Allow some data points to be on the 'wrong side' of the decision boundary.
- Increase a penalty with distance from the decision boundary.

The “soft margin” optimisation problem

SVM with “hard margins”

$$\arg\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \qquad (7.6)$$

s.t. $t_n\left(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)+b\right) \geqslant 1, \qquad n=1,\ldots,N. \qquad (7.5)$

total slack — same

minimize $C\sum_{n=1}^{N}\xi_n + \frac{1}{2}\|\mathbf{w}\|^2 \qquad (7.21)$

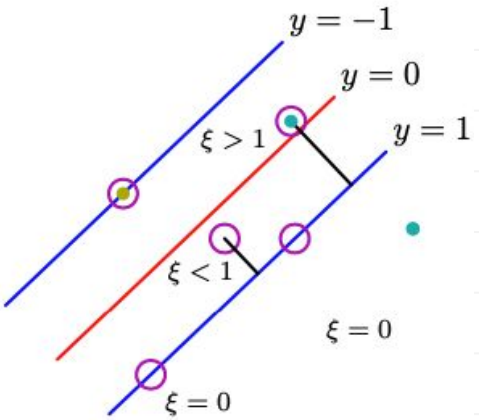
s.t. $\xi_n \geqslant 0$

$t_n y(\mathbf{x}_n) \geqslant 1-\xi_n, \qquad n=1,\ldots,N \qquad (7.20)$

margin violation

$\xi_n > 1$ if the point is mis-classified. Therefore $\boldsymbol{\Sigma}_n\, \xi_n$ is an upper bound on the number of misclassified points.

training



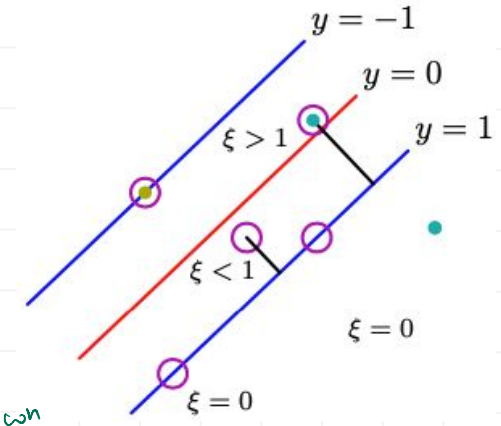
C controls the trade-off between the slack variable penalty and the margin -- the objective tries minimise “total slack” across all training points.

The “soft margin” optimisation problem

minimize $C\sum_{n=1}^{N}\xi_n + \frac{1}{2}\|\mathbf{w}\|^2$ (7.21)

s.t. $\xi_n \geqslant 0$

$t_n y(\mathbf{x}_n) \geqslant 1 - \xi_n, \qquad n = 1, \ldots, N$ (7.20)



obj additive their own L-multipliers.

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\xi_n - \sum_{n=1}^{N} a_n \{t_n y(\mathbf{x}_n) - 1 + \xi_n\} - \sum_{n=1}^{N}\mu_n\xi_n \quad (7.22)$$

KKT conditions

("outside the margin") a_n is 0 when point is

$a_n > 0$ when x_n mis-class or inside margin.

$$a_n \geqslant 0 \quad (7.23)$$

$$t_n y(\mathbf{x}_n) - 1 + \xi_n \geqslant 0 \quad (7.24)$$

$$a_n (t_n y(\mathbf{x}_n) - 1 + \xi_n) = 0 \quad (7.25)$$

$$\mu_n \geqslant 0 \quad (7.26)$$

$$\xi_n \geqslant 0 \quad (7.27)$$

$$\mu_n\xi_n = 0 \quad (7.28)$$

def of L-multipliers

orig constraints

What changed, if any, in its solution?

The dual problem of soft margin SVM

Set derivatives of L to zero to eliminate **w**, b, ξ_n , μ_n

$$\widetilde{L}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m) \tag{7.32}$$

$$0 \leqslant a_n \leqslant C \tag{7.33}$$

$$\sum_{n=1}^{N} a_n t_n = 0 \tag{7.34}$$

The only change from the separable case, is the box constraint via the parameter C.

Prediction function unchanged

$$y(\mathbf{x}) = \mathbf{w}^{\top} \phi(\mathbf{x}) + b = \sum_{n=1}^{N} a_n t_n k(\mathbf{x}, \mathbf{x}_n) + b \tag{7.13}$$

prediction time :
we need $k(x, x_n)$, $a_n > 0$
we don't need k
training time : ? ? ?
e.g. SMO
you may still need to compute k
Similar -

linear
poly
RBF
graph ...

Limitations of support vector machines

- Output are decisions, not posterior probabilities.
- Extension to classification with more than two classes is problematic.
- There is a complexity parameter C (or ν) which must be found (e.g. via cross-validation).
- Predictions are expressed as linear combinations of kernel functions that are centered on the training points.
- Kernel matrix is required to be positive definite.

platt's scaling,

Further Reading (No Assessable)

- Our derivation of the SVM
  - provides a nice example of Lagrangian optimisation
  - yields a neat dual formulation
  - is of historical value
- It is possible to derive an SVM algorithm much more simply however. See *e.g.*:

  Olivier Chapelle
  Training a support vector machine in the primal
  Neural computation, 2007 - MIT Press.

Outline

Lagrange multipliers, take 2

Maximum-margin classifiers (Chap 7.1)

- The intuition of margins
- Constructing the max-margin classifier
- The dual representation
- Support vectors and their geometric intuitions

SVM for overlapping class distributions

Relations to logistic regression

SVM for regression

Re-writing SVM objective with hinge loss

When $y_n t_n > 1$, $\xi_n = 0$; otherwise, $\xi_n = 1 - y_n t_n$

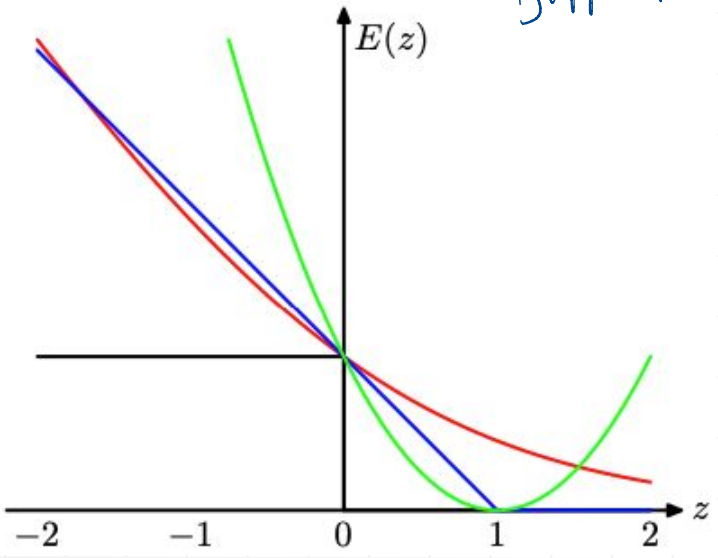
$$C \sum_{n=1}^{N} \xi_n + \frac{1}{2}\|\mathbf{w}\|^2 \tag{7.21}$$

$$\sum_{n=1}^{N} E_{\mathrm{SV}}(y_n t_n) + \lambda \|\mathbf{w}\|^2 \tag{7.44}$$

$$E_{\mathrm{SV}}(y_n t_n) = [1 - y_n t_n]_+$$

$[x]_+ \begin{cases} x, & x > 0 \\ 0, & x < 0 \end{cases}$

Clipped ReLU



$z = y_n t_n$

Figure 7.5 Plot of the 'hinge' error function used in support vector machines, shown in blue, along with the error function for logistic regression, rescaled by a factor of $1/\ln(2)$ so that it passes through the point $(0, 1)$, shown in red. Also shown are the misclassification error in black and the squared error in green.

Loss functions

- Linear Regression (squared loss)

$$\frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^\top \phi(x_n)\}^2 + \frac{\lambda}{2}\sum_{d=1}^{D} w_d^2 = \frac{1}{2}(\mathbf{t}-\mathbf{\Phi w})^T(\mathbf{t}-\mathbf{\Phi w}) + \frac{\lambda}{2}\mathbf{w}^\top \mathbf{w}$$

- Logistic Regression (log loss)

$$-\ln p(\mathbf{t}\,|\,\mathbf{w}) = -\sum_{n=1}^{N} \log\left(1+\exp(-t_n \mathbf{w}^\top \phi(x_n))\right) + \frac{\lambda}{2}\mathbf{w}^\top \mathbf{w}$$

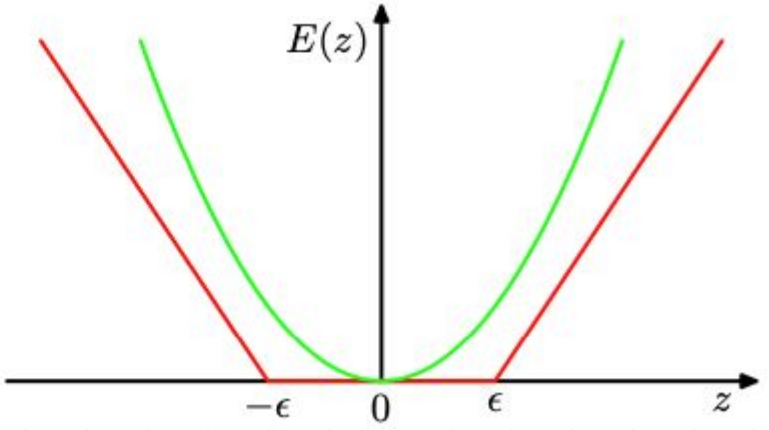
- Support Vector Machine (hinge loss)

$$C\sum_{n=1}^{N}[1 - t_n \mathbf{w}^\top \phi(x_n)]_+ + \frac{1}{2}\mathbf{w}^\top \mathbf{w}$$

← covered in IML?

ε-insensitive error function

Figure 7.6 Plot of an ϵ-insensitive error function (in red) in which the error increases linearly with distance beyond the insensitive region. Also shown for comparison is the quadratic error function (in green).



Sq. loss

$$\frac{1}{2}\sum_{n=1}^{N}\{y_n - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2. \tag{7.50}$$

SV regression

$$C\sum_{n=1}^{N} E_\epsilon(y(\mathbf{x}_n) - t_n) + \frac{1}{2}\|\mathbf{w}\|^2 \tag{7.52}$$

$$E_\epsilon(y(\mathbf{x}) - t) = \begin{cases} 0, & \text{if } |y(\mathbf{x}) - t| < \epsilon; \\ |y(\mathbf{x}) - t| - \epsilon, & \text{otherwise} \end{cases} \tag{7.51}$$

SVM regression with two set of slack variables

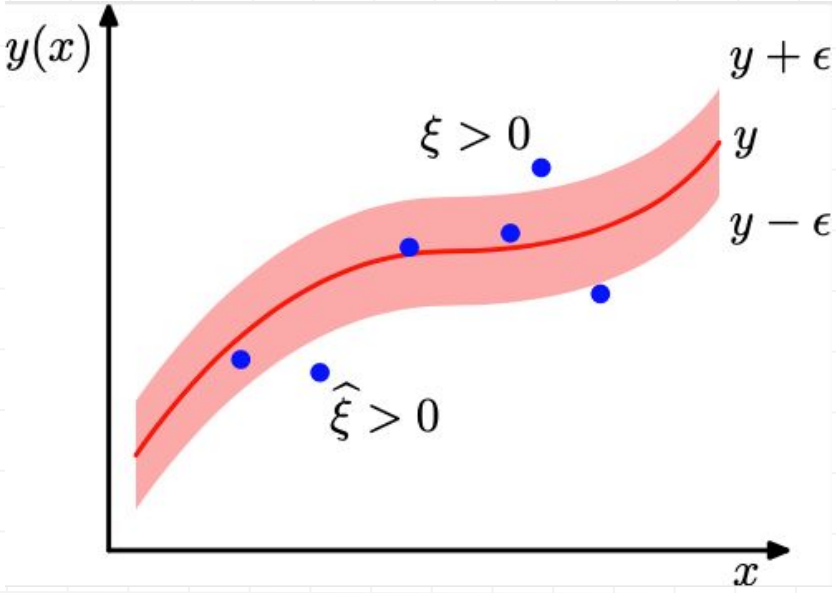


Figure 7.7 Illustration of SVM regression, showing the regression curve together with the ϵ-insensitive 'tube'. Also shown are examples of the slack variables ξ and $\widehat{\xi}$. Points above the ϵ-tube have $\xi > 0$ and $\widehat{\xi} = 0$, points below the ϵ-tube have $\xi = 0$ and $\widehat{\xi} > 0$, and points inside the ϵ-tube have $\xi = \widehat{\xi} = 0$.

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

total slack

see book

$$\widetilde{L}(\mathbf{a}, \widehat{\mathbf{a}}) = -\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}(a_n - \widehat{a}_n)(a_m - \widehat{a}_m)k(\mathbf{x}_n, \mathbf{x}_m) - \epsilon\sum_{n=1}^{N}(a_n + \widehat{a}_n) + \sum_{n=1}^{N}(a_n - \widehat{a}_n)t_n \qquad (7.61)$$

still uses kernels.

Prediction function

$$y(\mathbf{x}) = \sum_{n=1}^{N} (a_n - \widehat{a}_n) k(\mathbf{x}, \mathbf{x}_n) + b \tag{7.64}$$

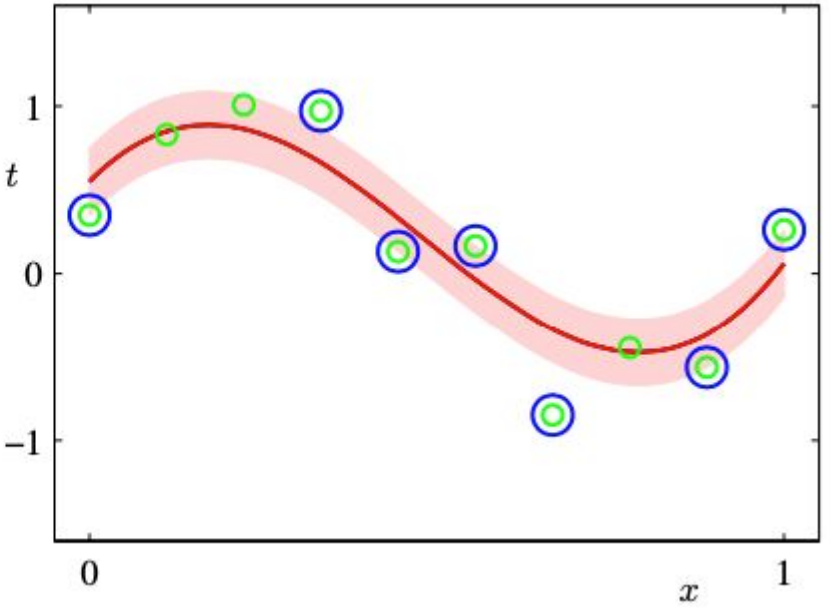


Figure 7.8 Illustration of the ν-SVM for regression applied to the sinusoidal synthetic data set using Gaussian kernels. The predicted regression curve is shown by the red line, and the ϵ-insensitive tube corresponds to the shaded region. Also, the data points are shown in green, and those with support vectors are indicated by blue circles.

Kernel machines

Lagrange multipliers, take 2

Maximum-margin classifiers (Chap 7.1)

- The intuition of margins
- Constructing the max-margin classifier
- The dual representation
- Support vectors and their geometric intuitions

SVM for overlapping class distributions

Relations to logistic regression

SVM for regression