



# Statistical Machine Learning

Christian Walder

Machine Learning Research Group  
CSIRO Data61

and

College of Engineering and Computer Science  
The Australian National University

Canberra  
Semester One, 2020.

## Outlines

Overview  
Introduction  
Linear Algebra  
Probability  
Linear Regression 1  
Linear Regression 2  
Linear Classification 1  
Linear Classification 2  
Kernel Methods  
Sparse Kernel Methods  
Mixture Models and EM 1  
Mixture Models and EM 2  
Neural Networks 1  
Neural Networks 2  
Principal Component Analysis  
Autoencoders  
Graphical Models 1  
Graphical Models 2  
Graphical Models 3  
Sampling  
Sequential Data 1  
Sequential Data 2

(Many figures from C. M. Bishop, "Pattern Recognition and Machine Learning")



## Part XIV

# *Sparse Kernel Machines*

*Sparse Kernel Machines*

*SVMs and Maximum  
Margin Classifiers*

*Lagrange Multipliers*

*SVMs and Maximum  
Margin Classifiers:  
Redux*

*Soft Margin SVMs:  
Non-Separable Case*

*Loss functions*

# Sparse Kernel Machines - Motivation

- 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.





- Return to the two-class classification with a linear model

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

where  $\phi(\mathbf{x})$  is some fixed feature space mapping, and  $b$  is the bias.

- Training data are  $N$  input vectors  $\mathbf{x}_1, \dots, \mathbf{x}_N$  with corresponding targets  $t_1, \dots, t_N$  where  $t_n \in \{-1, +1\}$ .
- The class of a new point is predicted as  $\text{sign}(y(\mathbf{x}))$ .
- Assume there exists a linear decision boundary. That means, there exist  $\mathbf{w}$  and  $b$  such that

$$t_n \text{sign}(y(\mathbf{x}_n)) > 0 \quad n = 1, \dots, N.$$

*Sparse Kernel Machines*

*SVMs and Maximum Margin Classifiers*

*Lagrange Multipliers*

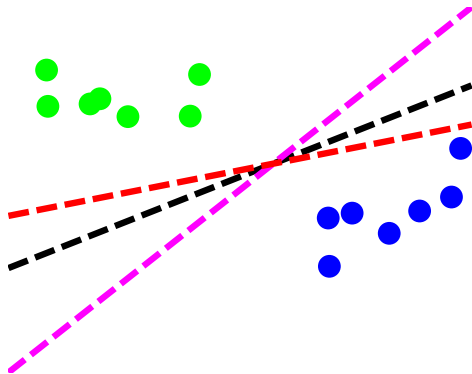
*SVMs and Maximum Margin Classifiers: Redux*

*Soft Margin SVMs: Non-Separable Case*

*Loss functions*

# The Multiple Separator problem

- There may exist many solutions  $w$  and  $b$  for which the linear classifier perfectly separates the two classes!





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

*Sparse Kernel Machines*

*SVMs and Maximum  
Margin Classifiers*

*Lagrange Multipliers*

*SVMs and Maximum  
Margin Classifiers:  
Redux*

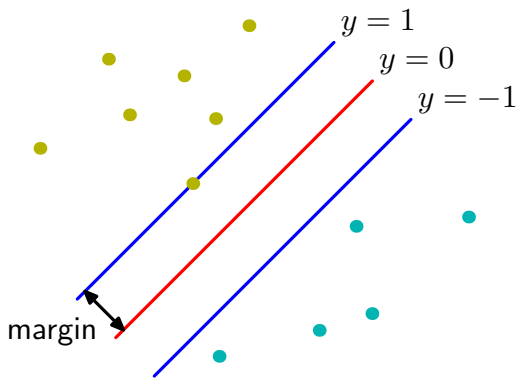
*Soft Margin SVMs:  
Non-Separable Case*

*Loss functions*

# Maximum Margin Classifiers



- The **margin** is the **smallest** distance between the decision boundary and any of the samples. (We will see later why  $y = \pm 1$  on the margin.)



*Sparse Kernel Machines*

*SVMs and Maximum Margin Classifiers*

*Lagrange Multipliers*

*SVMs and Maximum Margin Classifiers: Redux*

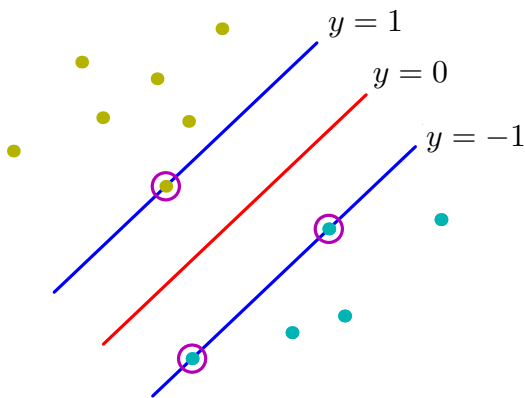
*Soft Margin SVMs: Non-Separable Case*

*Loss functions*

# Maximum Margin Classifiers



- Support Vector Machines (SVMs) choose the decision boundary which maximises the margin.



*Sparse Kernel Machines*

*SVMs and Maximum Margin Classifiers*

*Lagrange Multipliers*

*SVMs and Maximum Margin Classifiers: Redux*

*Soft Margin SVMs: Non-Separable Case*

*Loss functions*



# Reminder - Linear Classification

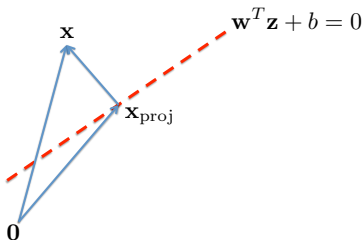
- For a linear model with weights  $\mathbf{w}$  and bias  $b$ , the decision boundary is  $\{\mathbf{x} : \mathbf{w}^\top \mathbf{x} + b = 0\}$
- Any  $\mathbf{x}$  has projection  $\mathbf{x}_{\text{proj}}$  on the boundary, and

$$\mathbf{x} - \mathbf{x}_{\text{proj}} = \lambda \cdot \mathbf{w}$$

since  $\mathbf{w}$  is orthogonal to the decision boundary

- But we also have

$$\mathbf{w}^\top \mathbf{x} - \mathbf{w}^\top \mathbf{x}_{\text{proj}} = \mathbf{w}^\top \mathbf{x} + b = \lambda \cdot \mathbf{w}^\top \mathbf{w}$$



# Reminder - Linear Classification

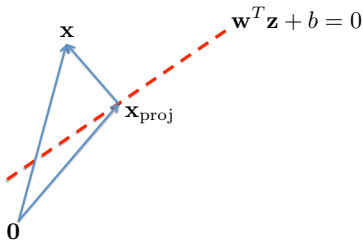


- So,

$$\begin{aligned}\|\mathbf{x} - \mathbf{x}_{\text{proj}}\| &= |\lambda| \cdot \|\mathbf{w}\| \\ &= \frac{|\mathbf{w}^T \mathbf{x} + b|}{\|\mathbf{w}\|}\end{aligned}$$

- The distance of  $\mathbf{x}$  from the decision boundary is therefore

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

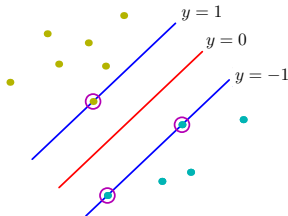


# Maximum Margin Classifiers



- **Support Vector Machines** choose the decision boundary which maximises the smallest distance to samples in both classes.
- We calculated the distance of a point  $\mathbf{x}$  from the hyperplane  $y(\mathbf{x}) = 0$  as  $|y(\mathbf{x})|/\|\mathbf{w}\|$ .
- Perfect classification means  $t_n y(\mathbf{x}_n) > 0$  for all  $n$ .
- Thus, the distance of  $\mathbf{x}_n$  from the decision boundary is

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



*Sparse Kernel Machines*

*SVMs and Maximum  
Margin Classifiers*

*Lagrange Multipliers*

*SVMs and Maximum  
Margin Classifiers:  
Redux*

*Soft Margin SVMs:  
Non-Separable Case*

*Loss functions*



- **Support Vector Machines** choose the decision boundary which maximises the smallest distance to samples in both classes.
- For the **maximum margin solution**, solve

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

- How can we solve this?

*Sparse Kernel Machines*

*SVMs and Maximum  
Margin Classifiers*

*Lagrange Multipliers*

*SVMs and Maximum  
Margin Classifiers:  
Redux*

*Soft Margin SVMs:  
Non-Separable Case*

*Loss functions*



- Maximum margin solution

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

or

$$\arg \max_{\mathbf{w}, b, \gamma} \left\{ \frac{\gamma}{\|\mathbf{w}\|} \right\} \quad \text{s.t.} \quad t_n (\mathbf{w}^\top \phi(\mathbf{x}_n) + b) \geq \gamma \quad n = 1, \dots, N.$$

- By rescaling any  $(\mathbf{w}, b)$  by  $1/\gamma$ , we don't change the answer!
  - We can assume that  $\gamma = 1$
- The **canonical representation** for the decision hyperplane is therefore

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

*Sparse Kernel Machines*

*SVMs and Maximum Margin Classifiers*

*Lagrange Multipliers*

*SVMs and Maximum Margin Classifiers: Redux*

*Soft Margin SVMs: Non-Separable Case*

*Loss functions*



- Maximum margin solution

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

- Transformed once

$$\arg \max_{\mathbf{w}, b} \left\{ \frac{1}{\|\mathbf{w}\|} \right\} \quad \text{s.t. } t_n (\mathbf{w}^\top \phi(\mathbf{x}_n) + b) \geq 1.$$

- Transformed again

$$\arg \min_{\mathbf{w}, b} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 \right\} \quad \text{s.t. } t_n (\mathbf{w}^\top \phi(\mathbf{x}_n) + b) \geq 1.$$

*Sparse Kernel Machines*

*SVMs and Maximum  
Margin Classifiers*

*Lagrange Multipliers*

*SVMs and Maximum  
Margin Classifiers:  
Redux*

*Soft Margin SVMs:  
Non-Separable Case*

*Loss functions*



# Lagrange Multipliers

*Sparse Kernel Machines*

*SVMs and Maximum  
Margin Classifiers*

*Lagrange Multipliers*

*SVMs and Maximum  
Margin Classifiers:  
Redux*

*Soft Margin SVMs:  
Non-Separable Case*

*Loss functions*



- Find the stationary points for a function  $f(x_1, x_2)$  subject to one or more constraints on the variables  $x_1$  and  $x_2$  written in the form  $g(x_1, x_2) = 0$ .
- Direct approach
  - 1 Solve  $g(x_1, x_2) = 0$  for one of the variables to get  $x_2 = h(x_1)$ .
  - 2 Insert the result into  $f(x_1, x_2)$  to get a function of one variable  $f(x_1, h(x_1))$ .
  - 3 Find the stationary point(s)  $x_1^*$  of  $f(x_1, h(x_1))$  with corresponding value  $x_2^* = h(x_1^*)$ .
- Finding  $x_2 = h(x_1)$  may be hard.
- Symmetry in the variables  $x_1$  and  $x_2$  is lost.

*Sparse Kernel Machines*

*SVMs and Maximum  
Margin Classifiers*

*Lagrange Multipliers*

*SVMs and Maximum  
Margin Classifiers:  
Redux*

*Soft Margin SVMs:  
Non-Separable Case*

*Loss functions*





- To solve

$$\max_{\mathbf{x}} f(\mathbf{x}) \text{ s.t. } g(\mathbf{x}) = 0$$

introduce the **Lagrangian** function

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

from which we get the constraint stationary conditions

$$\nabla_{\mathbf{x}} L(\mathbf{x}, \lambda) = \nabla f(\mathbf{x}) + \lambda \nabla g(\mathbf{x}) = 0$$

and the constraint itself

$$\frac{\partial L(\mathbf{x}, \lambda)}{\partial \lambda} = g(\mathbf{x}) = 0.$$

- This are  $D$  equations resulting from  $\nabla_{\mathbf{x}} L(\mathbf{x}, \lambda)$  and one equation from  $\frac{\partial L(\mathbf{x}, \lambda)}{\partial \lambda}$ , together determining  $\mathbf{x}^*$  and  $\lambda$ .

*Sparse Kernel Machines*

*SVMs and Maximum  
Margin Classifiers*

**Lagrange Multipliers**

*SVMs and Maximum  
Margin Classifiers:  
Redux*

*Soft Margin SVMs:  
Non-Separable Case*

*Loss functions*

# Lagrange Multipliers - Example



- Given  $f(x_1, x_2) = 1 - x_1^2 - x_2^2$  subject to the constraint  $g(x_1, x_2) = x_1 + x_2 - 1 = 0$ .
- Define the Lagrangian function

$$L(\mathbf{x}, \lambda) = 1 - x_1^2 - x_2^2 + \lambda(x_1 + x_2 - 1).$$

- A stationary solution with respect to  $x_1$ ,  $x_2$ , and  $\lambda$  must satisfy

$$-2x_1 + \lambda = 0$$

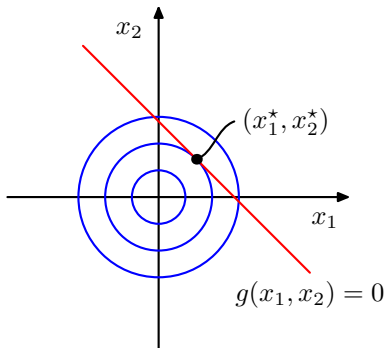
$$-2x_2 + \lambda = 0$$

$$x_1 + x_2 - 1 = 0.$$

- Therefore  $(x_1^*, x_2^*) = (\frac{1}{2}, \frac{1}{2})$  and  $\lambda = 1$ .

# Lagrange Multipliers - Example

- Given  $f(x_1, x_2) = 1 - x_1^2 - x_2^2$  subject to the constraint  $g(x_1, x_2) = x_1 + x_2 - 1 = 0$ .
- Lagrangian  $L(\mathbf{x}, \lambda) = 1 - x_1^2 - x_2^2 + \lambda(x_1 + x_2 - 1)$
- $(x_1^*, x_2^*) = (\frac{1}{2}, \frac{1}{2})$ .



# Lagrange Multipliers for Inequality Constraints



- To solve

$$\max_{\mathbf{x}} f(\mathbf{x}) \text{ s.t. } g(\mathbf{x}) \geq 0$$

define the Lagrangian

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

- Solve for  $\mathbf{x}$  and  $\lambda$  subject to the constraints  
(**Karush-Kuhn-Tucker** or KKT conditions )

$$g(\mathbf{x}) \geq 0$$

$$\lambda \geq 0$$

$$\lambda g(\mathbf{x}) = 0$$

*Sparse Kernel Machines*

*SVMs and Maximum  
Margin Classifiers*

*Lagrange Multipliers*

*SVMs and Maximum  
Margin Classifiers:  
Redux*

*Soft Margin SVMs:  
Non-Separable Case*

*Loss functions*

# Lagrange Multipliers for General Case



- Maximise  $f(\mathbf{x})$  subject to the constraints  $g_j(\mathbf{x}) = 0$  for  $j = 1, \dots, J$ , and  $h_k(\mathbf{x}) \geq 0$  for  $k = 1, \dots, K$ .
- Define the Lagrange multipliers  $\{\lambda_j\}$  and  $\{\mu_k\}$ , and the Lagrangian

$$L(\mathbf{x}, \{\lambda_j\}, \{\mu_k\}) = f(\mathbf{x}) + \sum_{j=1}^J \lambda_j g_j(\mathbf{x}) + \sum_{k=1}^K \mu_k h_k(\mathbf{x}).$$

- Solve for  $\mathbf{x}$ ,  $\{\lambda_j\}$ , and  $\{\mu_k\}$  subject to the constraints (Karush-Kuhn-Tucker or KKT conditions)

$$\begin{aligned} \mu_k &\geq 0 \\ \mu_k h_k(\mathbf{x}) &= 0 \end{aligned}$$

for  $k = 1, \dots, K$ .

- For minimisation of  $f(\mathbf{x})$ , change the sign in front of the Lagrange multipliers.



# SVMs and Maximum Margin Classifiers: Redux

*Sparse Kernel Machines*

*SVMs and Maximum  
Margin Classifiers*

*Lagrange Multipliers*

*SVMs and Maximum  
Margin Classifiers:  
Redux*

*Soft Margin SVMs:  
Non-Separable Case*

*Loss functions*



- **Quadratic Programming** (QP) problem: minimise a quadratic function subject to linear constraints.

$$\arg \min_{\mathbf{w}, b} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 \right\} \quad \text{s.t. } t_n (\mathbf{w}^\top \phi(\mathbf{x}_n) + b) \geq 1.$$

- Introduce Lagrange multipliers  $\{a_n \geq 0\}$  to get the Lagrangian

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

- note negative sign since minimisation problem

*Sparse Kernel Machines*

*SVMs and Maximum  
Margin Classifiers*

*Lagrange Multipliers*

*SVMs and Maximum  
Margin Classifiers:  
Redux*

*Soft Margin SVMs:  
Non-Separable Case*

*Loss functions*



- Lagrangian

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{n=1}^N a_n \{t_n (\mathbf{w}^\top \phi(\mathbf{x}_n) + b) - 1\}.$$

- Derivatives with respect to  $\mathbf{w}$  and  $b$  are

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

- Dual representation (again QP problem): maximise

$$\begin{aligned} \tilde{L}(\mathbf{a}) &= \sum_{n=1}^N a_n - \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m) \\ \sum_{n=1}^N a_n t_n &= 0 \\ a_n &\geq 0 \qquad n = 1, \dots, N. \end{aligned}$$



# Maximum Margin Classifiers



- Dual representation (again QP problem): maximise

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

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

$$a_n \geq 0 \quad n = 1, \dots, N.$$

- Vector form: maximise

$$\tilde{L}(\mathbf{a}) = \mathbf{1}^\top \mathbf{a} - \frac{1}{2} \mathbf{a}^\top \mathbf{Q} \mathbf{a}$$

$$\mathbf{a}^\top \mathbf{t} = 0$$

$$a_n \geq 0 \quad n = 1, \dots, N.$$

where

$$\mathbf{Q}_{nm} = t_n \cdot t_m \cdot k(\mathbf{x}_n, \mathbf{x}_m)$$

*Sparse Kernel Machines*

*SVMs and Maximum Margin Classifiers*

*Lagrange Multipliers*

*SVMs and Maximum Margin Classifiers: Redux*

*Soft Margin SVMs: Non-Separable Case*

*Loss functions*

# Maximum Margin Classifiers - Prediction



- Predict via

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

- The **Karush-Kuhn-Tucker** (KKT) conditions are

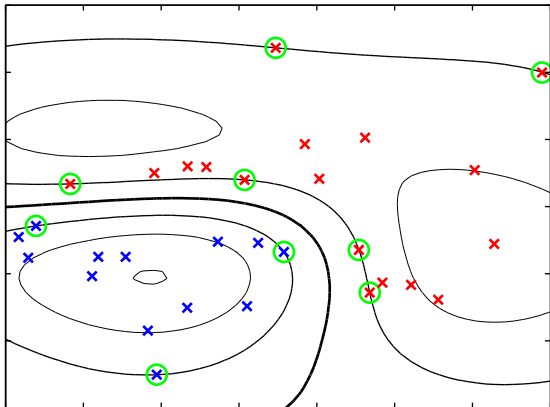
$$\begin{aligned} a_n &\geq 0 \\ t_n y(\mathbf{x}_n) - 1 &\geq 0 \\ a_n \{t_n y(\mathbf{x}_n) - 1\} &= 0. \end{aligned}$$

- Therefore, either  $a_n = 0$  or  $t_n y(\mathbf{x}_n) - 1 = 0$ .
- If  $a_n = 0$ , no contribution to the prediction!
- After training, use only the set  $\mathcal{S}$  of points for which  $a_n > 0$  (and therefore  $t_n y(\mathbf{x}_n) = 1$ ) holds.
- $\mathcal{S}$  contains only the **support vectors**.

# Maximum Margin Classifiers - Support Vectors



- $S$  contains only the support vectors.



Decision and margin boundaries for a two-class problem using Gaussian kernel functions.

# Maximum Margin Classifiers - Support Vectors



- Now for the bias  $b$ .
- Observe that for the support vectors,  $t_n y(\mathbf{x}_n) = 1$ .
- Therefore, we find

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

and can use any support vector to calculate  $b$ .

- Numerically more stable: Multiply by  $t_n$ , observe  $t_n^2 = 1$ , and average over all support vectors.

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



- Maximise the dual objective

$$\tilde{L}(\mathbf{a}) = \mathbf{1}^\top \mathbf{a} - \frac{1}{2} \mathbf{a}^\top \mathbf{Q} \mathbf{a}$$

$$\mathbf{a}^\top \mathbf{t} = 0$$

$$a_n \geq 0 \quad n = 1, \dots, N.$$

- Make predictions via

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

- Dependence only on data points with  $a_n > 0$

*Sparse Kernel Machines*

*SVMs and Maximum  
Margin Classifiers*

*Lagrange Multipliers*

*SVMs and Maximum  
Margin Classifiers:  
Redux*

*Soft Margin SVMs:  
Non-Separable Case*

*Loss functions*



# Soft Margin SVMs: Non-Separable Case

*Sparse Kernel Machines*

*SVMs and Maximum  
Margin Classifiers*

*Lagrange Multipliers*

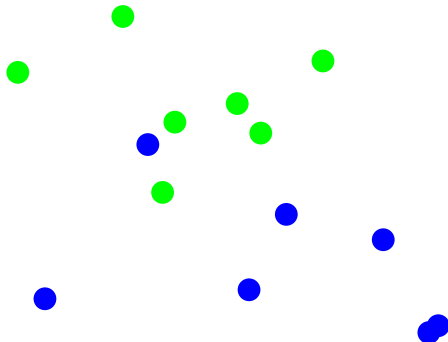
*SVMs and Maximum  
Margin Classifiers:  
Redux*

*Soft Margin SVMs:  
Non-Separable Case*

*Loss functions*



- What if the training data in the feature space are not linearly separable?
  - Can we find a separator that makes fewest possible errors?



*Sparse Kernel Machines*

*SVMs and Maximum Margin Classifiers*

*Lagrange Multipliers*

*SVMs and Maximum Margin Classifiers: Redux*

*Soft Margin SVMs: Non-Separable Case*

*Loss functions*



- What if the training data in the feature space are not linearly separable?
  - Allow some data points to be on the 'wrong side' of the decision boundary.
  - Increase a penalty with distance from the decision boundary.
- Assume, the penalty increases linearly with distance.
- Introduce **slack** variable  $\xi_n \geq 0$  for each data point  $n$ .

$$\xi_n \begin{cases} = 0, & \text{data point is correctly classified and} \\ & \text{on margin boundary or beyond} \\ < 1, & \text{data point is in correct margin} \\ = 1, & \text{data point is on the decision boundary} \\ > 1, & \text{data point is misclassified} \end{cases}$$

*Sparse Kernel Machines*

*SVMs and Maximum  
Margin Classifiers*

*Lagrange Multipliers*

*SVMs and Maximum  
Margin Classifiers:  
Redux*

*Soft Margin SVMs:  
Non-Separable Case*

*Loss functions*





- Constraints of the separable classification

$$t_n y(\mathbf{x}_n) \geq 1, \quad n = 1, \dots, N$$

change now to

$$t_n y(\mathbf{x}_n) \geq 1 - \xi_n, \quad n = 1, \dots, N$$

- In sum, we have the constraints

$$\xi_n \geq 0$$

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

*Sparse Kernel Machines*

*SVMs and Maximum  
Margin Classifiers*

*Lagrange Multipliers*

*SVMs and Maximum  
Margin Classifiers:  
Redux*

*Soft Margin SVMs:  
Non-Separable Case*

*Loss functions*



- Find now

$$\min_{\mathbf{w}, \xi} C \sum_{n=1}^N \xi_n + \frac{1}{2} \|\mathbf{w}\|^2 \text{ s.t. } \xi_n \geq 0$$

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

where  $C$  controls the trade-off between the slack variable penalty and the margin.

- Minimise the total slack associated with all data points
- Any misclassified point contributes  $\xi_n > 1$ 
  - therefore  $\sum_{n=1}^N \xi_n$  is an upper bound on the number of misclassified points.

*Sparse Kernel Machines*

*SVMs and Maximum  
Margin Classifiers*

*Lagrange Multipliers*

*SVMs and Maximum  
Margin Classifiers:  
Redux*

*Soft Margin SVMs:  
Non-Separable Case*

*Loss functions*



- The Lagrangian with the Lagrange multipliers  $\mathbf{a} = (a_1, \dots, a_N)^\top$  and  $\boldsymbol{\mu} = (\mu_1, \dots, \mu_N)^\top$  is now

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

- The KKT conditions for  $n = 1, \dots, N$  are

$$\begin{aligned} a_n &\geq 0 \\ t_n y(\mathbf{x}_n) - 1 + \xi_n &\geq 0 \\ a_n(t_n y(\mathbf{x}_n) - 1 + \xi_n) &= 0 \\ \mu_n &\geq 0 \\ \xi_n &\geq 0 \\ \mu_n \xi_n &= 0. \end{aligned}$$

*Sparse Kernel Machines*

*SVMs and Maximum  
Margin Classifiers*

*Lagrange Multipliers*

*SVMs and Maximum  
Margin Classifiers:  
Redux*

*Soft Margin SVMs:  
Non-Separable Case*

*Loss functions*



- The dual Lagrangian after eliminating  $\mathbf{w}$ ,  $b$ ,  $\xi$ , and  $\mu$  is then

$$\tilde{L}(\mathbf{a}) = \sum_{n=1}^N a_n - \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m)$$
$$\sum_{n=1}^N a_n t_n = 0$$
$$0 \leq a_n \leq C \quad n = 1, \dots, N.$$

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

*Sparse Kernel Machines*

*SVMs and Maximum  
Margin Classifiers*

*Lagrange Multipliers*

*SVMs and Maximum  
Margin Classifiers:  
Redux*

*Soft Margin SVMs:  
Non-Separable Case*

*Loss functions*



- Equivalent formulation by Schölkopf et. al. (2000)

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

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

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

$$\sum_{n=1}^N a_n \geq \nu \quad n = 1, \dots, N.$$

where  $\nu$  is both

- 1 an upper bound on the fraction of margin errors, and,
- 2 a lower bound on the fraction of support vectors.

*Sparse Kernel Machines*

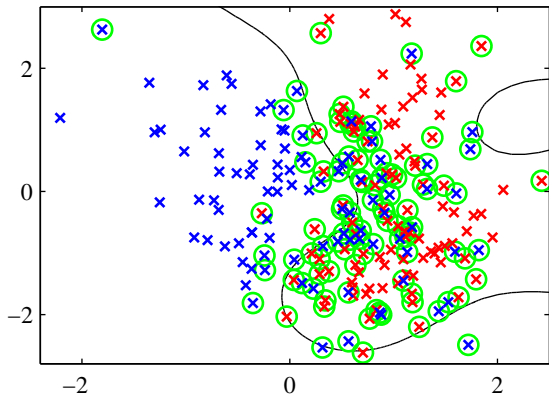
*SVMs and Maximum  
Margin Classifiers*

*Lagrange Multipliers*

*SVMs and Maximum  
Margin Classifiers:  
Redux*

*Soft Margin SVMs:  
Non-Separable Case*

*Loss functions*



The  $\nu$ -SVM algorithm using Gaussian kernels  $\exp(-\gamma\|\mathbf{x} - \mathbf{x}'\|^2)$  with  $\gamma = 0.45$  applied to a nonseparable data set in two dimensions. Support vectors are indicated by circles.

# Support Vector Machines - Limitations



- Output are decisions, not posterior probabilities.
- Extension to classification with more than two classes is problematic.
- There is a complexity parameter  $C$  (or  $\nu$ ) 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.

*Sparse Kernel Machines*

*SVMs and Maximum  
Margin Classifiers*

*Lagrange Multipliers*

*SVMs and Maximum  
Margin Classifiers:  
Redux*

*Soft Margin SVMs:  
Non-Separable Case*

*Loss functions*



- Recall from decision theory that we want to minimize the expected loss

$$\begin{aligned} p(\text{mistake}) &= p(\mathbf{x} \in \mathcal{R}_1, \mathcal{C}_2) + p(\mathbf{x} \in \mathcal{R}_2, \mathcal{C}_1) \\ &= \int_{\mathcal{R}_1} p(\mathbf{x}, \mathcal{C}_2) \, d\mathbf{x} + \int_{\mathcal{R}_2} p(\mathbf{x}, \mathcal{C}_1) \, d\mathbf{x} \end{aligned}$$

- For finite data, we minimise the negative log likelihood, appropriately regularized.
- More generally, we minimise the empirical risk (defined by a loss function).

$$\mathcal{R}_{\text{emp}}(\mathbf{w}, \mathbf{X}, \mathbf{t}) + \Omega(\mathbf{w}) = \sum_{n=1}^N \underbrace{\ell(\mathbf{x}_n, t_n; \mathbf{w})}_{\text{loss per example}} + \underbrace{\Omega(\mathbf{w})}_{\text{regulariser}}$$

*Sparse Kernel Machines*

*SVMs and Maximum  
Margin Classifiers*

*Lagrange Multipliers*

*SVMs and Maximum  
Margin Classifiers:  
Redux*

*Soft Margin SVMs:  
Non-Separable Case*

*Loss functions*





# Logistic Regression Loss Function?

- Consider the labels  $t_n \in \{-1, +1\}$ .
- Recall the definition of the sigmoid  $\sigma(\cdot)$ , then

$$p(t_n = 1|x_n) = \sigma(\mathbf{w}^\top \phi(x_n)) = \frac{1}{1 + \exp(-\mathbf{w}^\top \phi(x_n))}$$

$$p(t_n = -1|x_n) = 1 - \sigma(\mathbf{w}^\top \phi(x_n)) = \frac{1}{1 + \exp(+\mathbf{w}^\top \phi(x_n))}$$

which can be compactly written as

$$p(t_n|x_n) = \sigma(\mathbf{w}^\top \phi(x_n)) = \frac{1}{1 + \exp(-t_n \mathbf{w}^\top \phi(x_n))}$$

- Assuming independence, the negative log likelihood

$$\sum_{n=1}^N \log (1 + \exp(-t_n \mathbf{w}^\top \phi(x_n)))$$

# SVM Loss Function?



- Soft margin SVM

$$\begin{aligned} \min_{\mathbf{w}, \xi} C \sum_{n=1}^N \xi_n + \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{s.t. } \xi_n \geq 0 \\ \xi_n \geq 1 - t_n y(\mathbf{x}_n). \end{aligned}$$

- Observe the equivalent unconstrained loss:

$$\begin{aligned} \min_{\xi, y} \xi \\ \text{s.t. } \xi \geq 0 \\ \xi \geq 1 - y. \end{aligned}$$

is equivalent to

$$\min_y \max\{0, 1 - y\}$$

- We denote the **hinge loss** by  $[1 - y]_+ = \max\{0, 1 - y\}$ .

*Sparse Kernel Machines*

*SVMs and Maximum Margin Classifiers*

*Lagrange Multipliers*

*SVMs and Maximum Margin Classifiers: Redux*

*Soft Margin SVMs: Non-Separable Case*

*Loss functions*



- Linear Regression (squared loss)

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

- Logistic Regression (log loss)

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

- Support Vector Machine (hinge loss)

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

*Sparse Kernel Machines*

*SVMs and Maximum  
Margin Classifiers*

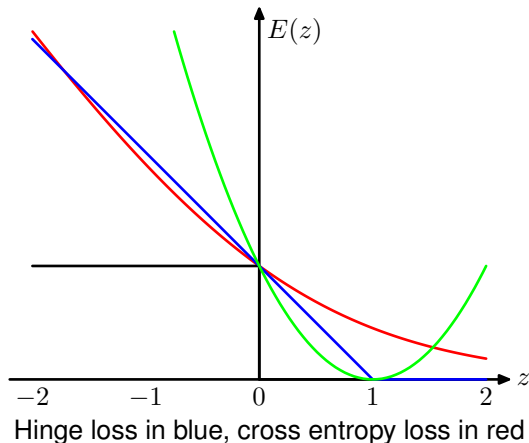
*Lagrange Multipliers*

*SVMs and Maximum  
Margin Classifiers:  
Redux*

*Soft Margin SVMs:  
Non-Separable Case*

*Loss functions*

# Loss functions



*Sparse Kernel Machines*

*SVMs and Maximum Margin Classifiers*

*Lagrange Multipliers*

*SVMs and Maximum Margin Classifiers: Redux*

*Soft Margin SVMs: Non-Separable Case*

*Loss functions*

## 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.

*Sparse Kernel Machines*

*SVMs and Maximum  
Margin Classifiers*

*Lagrange Multipliers*

*SVMs and Maximum  
Margin Classifiers:  
Redux*

*Soft Margin SVMs:  
Non-Separable Case*

*Loss functions*