Support Vector Machines

Dr. Francis Colas

04.11.2011

Machine learning

Learning:

- adapt algorithm to empirical data;
- learn different things:
  - relationship between variables,
  - clusters,
  - classes...
- in different ways:
  - supervised,
  - unsupervised...

Machine learning

Regression:

- relationship between variables,
- linear regression,
- probabilistic formulation,
- Gaussian processes;

Gaussian process:

- probability distribution over functions,
- specified by a kernel,
- can be applied to regression,
- and classification.

Machine learning

Regression:

- relationship between variables,
- linear regression,
- probabilistic formulation,
- Gaussian processes;

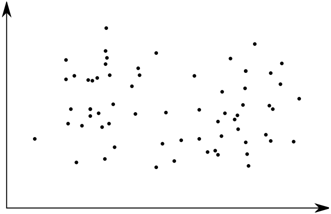
Gaussian process:

- probability distribution over functions,
- specified by a kernel,
- can be applied to regression,
- and classification.

Classification

Clustering:

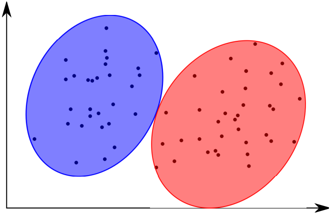
- non supervised,
- groupe points together.



Classification

Clustering:

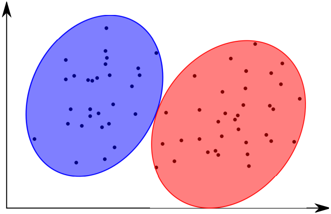
- non supervised,
- groupe points together.



Classification

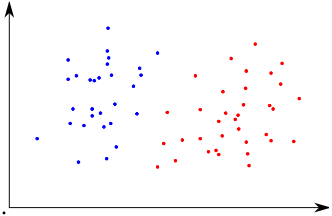
Clustering:

- non supervised,
- groupe points together.



Classification:

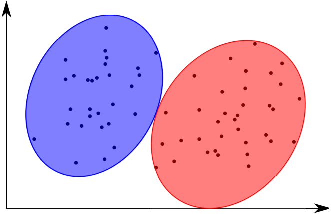
- supervised,
- function to give the class.



Classification

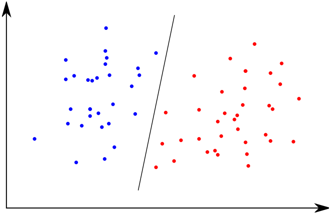
Clustering:

- non supervised,
- groupe points together.



Classification:

- supervised,
- function to give the class.



Approaches for Object Classification

Two phases:

- learning of the classes separation with training data,
- estimating the class for test data.

Many different approaches:

- Support Vector Machines (SVMs),
- Adaboost,
- Voting Techniques,
- Conditional Random Fields,
- Combinations of these methods,
- ...

Approaches for Object Classification

Two phases:

- learning of the classes separation with training data,
- estimating the class for test data.

Many different approaches:

- Support Vector Machines (SVMs),
- Adaboost,
- Voting Techniques,
- Conditional Random Fields,
- Combinations of these methods,
- . . .

Support Vector Machines

Support Vector Machines:

- separate classes with a line or hyperplane,
- optimizing the margin,
- under classification constraints (supervised),
- in a feature space or using a kernel (kernel method),
- quadratic programming formulation.

Support Vector Machines

Support Vector Machines:

- separate classes with a line or hyperplane,
- optimizing the margin,
- under classification constraints (supervised),
- in a feature space or using a kernel (kernel method),
- quadratic programming formulation.

Support Vector Machines

Support Vector Machines:

- separate classes with a line or hyperplane,
- optimizing the margin,
- under classification constraints (supervised),
- in a feature space or using a kernel (kernel method),
- quadratic programming formulation.

Support Vector Machines

Support Vector Machines:

- separate classes with a line or hyperplane,
- optimizing the margin,
- under classification constraints (supervised),
- in a feature space or using a kernel (kernel method),
- quadratic programming formulation.

Support Vector Machines

Support Vector Machines:

- separate classes with a line or hyperplane,
- optimizing the margin,
- under classification constraints (supervised),
- in a feature space or using a kernel (kernel method),
- quadratic programming formulation.

Linear Discriminant Function

Assumptions:

- training data points: $(\mathbf{x}_i, t_i)$,
- binary classification: $t_i \in \{-1, 1\}$,
- optional transformation in feature space: $\phi(\mathbf{x})$,
- linearly separable (in feature space).

Separation: hyperplane:

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

with:

- $\mathbf{w}$: parameters of the hyperplane (normal vector),
- ϕ: feature function vector,
- b: bias parameter.

Classification: $\forall n, y(\mathbf{x}_n) > 0 \Leftrightarrow t_n = 1$

Linear Discriminant Function

Assumptions:

- training data points: $(\mathbf{x}_i, t_i)$,
- binary classification: $t_i \in \{-1, 1\}$,
- optional transformation in feature space: $\phi(\mathbf{x})$,
- linearly separable (in feature space).

Separation: hyperplane:

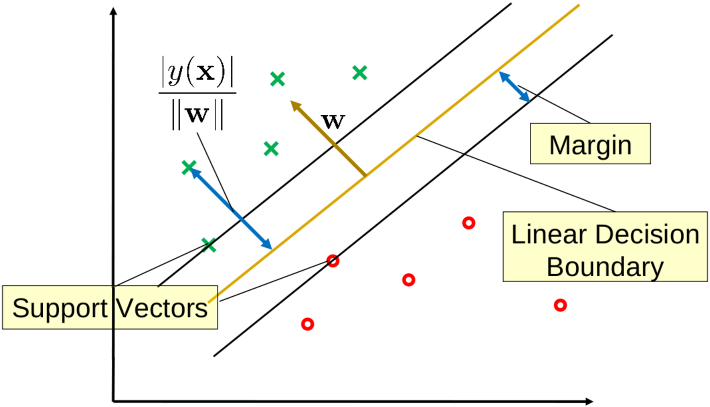
$$y(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b \quad \text{or} \quad y(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$$

with:

- $\mathbf{w}$: parameters of the hyperplane (normal vector),
- ϕ: feature function vector,
- b: bias parameter.

Classification: $\forall n, y(\mathbf{x}_n) > 0 \Leftrightarrow t_n = 1$

Margin



Several solutions:

$\Longrightarrow$ maximize the margin.

Margin: formal definition

Distance of a point $\mathbf{x}$ to the hyperplane:

- $|\mathbf{w}^T\mathbf{x} + b| = |y(\mathbf{x})|$, if $\mathbf{w}$ is normalized;
- $\frac{|y(\mathbf{x})|}{\|\mathbf{w}\|}$, in general.

Margin:

$$\min_n \frac{|y(\mathbf{x}_n)|}{\|\mathbf{w}\|}$$

Maximum margin:

$$\arg\max_{\mathbf{w},b} \left\{ \frac{1}{\|\mathbf{w}\|} min_n |y(\mathbf{x}_n)| \right\}$$

Margin: formal definition

Distance of a point $\mathbf{x}$ to the hyperplane:

- $|\mathbf{w}^T\mathbf{x} + b| = |y(\mathbf{x})|$, if $\mathbf{w}$ is normalized;
- $\frac{|y(\mathbf{x})|}{\|\mathbf{w}\|}$, in general.

Margin:

$$\min_n \frac{|y(\mathbf{x}_n)|}{\|\mathbf{w}\|}$$

Maximum margin:

$$\arg\max_{\mathbf{w},b} \left\{ \frac{1}{\|\mathbf{w}\|} min_n |y(\mathbf{x}_n)| \right\}$$

Margin: formal definition

Distance of a point $\mathbf{x}$ to the hyperplane:

- $|\mathbf{w}^T\mathbf{x} + b| = |y(\mathbf{x})|$, if $\mathbf{w}$ is normalized;
- $\frac{|y(\mathbf{x})|}{||\mathbf{w}||}$, in general.

Margin:

$$\min_n \frac{|y(\mathbf{x}_n)|}{||\mathbf{w}||}$$

Maximum margin:

$$\arg\max_{\mathbf{w},b} \left\{ \frac{1}{||\mathbf{w}||} min_n |y(\mathbf{x}_n)| \right\}$$

Restating the problem

Classification constraints:

$$\forall n, sign(y(\mathbf{x}_n)) = t_n$$

Target classes:

$$\begin{array}{rrcl} & sign(y(\mathbf{x}_n)) & = & t_n \\ \iff & |y(\mathbf{x}_n)| & = & t_n.y_n(\mathbf{x}_n) \\ \iff & |y(\mathbf{x}_n)| & = & t_n(\mathbf{w}^T\mathbf{x}_n + b) \end{array}$$

Maximum margin solution:

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

Restating the problem

Classification constraints:

$$\forall n, sign(y(\mathbf{x}_n)) = t_n$$

Target classes:

$$\begin{array}{rrcl} & sign(y(\mathbf{x}_n)) & = & t_n \\ \iff & |y(\mathbf{x}_n)| & = & t_n.y_n(\mathbf{x}_n) \\ \iff & |y(\mathbf{x}_n)| & = & t_n(\mathbf{w}^T\mathbf{x}_n + b) \end{array}$$

Maximum margin solution:

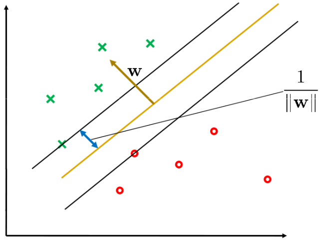
$$\arg\max_{\mathbf{w},b} \left\{ \frac{1}{\|\mathbf{w}\|} \min_n \left[t_n(\mathbf{w}^T\mathbf{x}_n + b) \right] \right\}$$

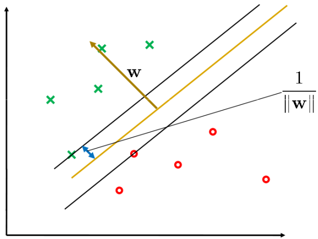
Rescaling (1/2)

We have:

$$\begin{aligned} sign(y(\mathbf{x}_n)) &= t_n \\ \iff \quad t_n(\mathbf{w}^T\mathbf{x}_n + b) &> 0 \end{aligned}$$

$\frac{t_n(\mathbf{w}^T\mathbf{x}_n+b)}{\|\mathbf{w}\|}$ is scaled but $t_n(\mathbf{w}^T\mathbf{x}_n) + b)$ is not.



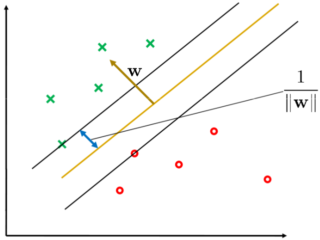


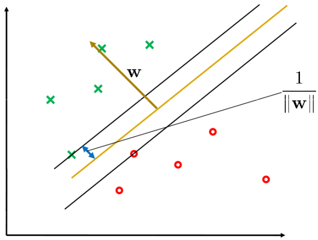
Rescaling (1/2)

We have:

$$\begin{aligned} sign(y(\mathbf{x}_n)) &= t_n \\ \iff t_n(\mathbf{w}^T\mathbf{x}_n + b) &> 0 \end{aligned}$$

$\frac{t_n(\mathbf{w}^T\mathbf{x}_n+b)}{\|\mathbf{w}\|}$ is scaled but $t_n(\mathbf{w}^T\mathbf{x}_n) + b)$ is not.





Rescaling (2/2)

For support vectors, we decide:

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

That is (canonical representation):

$$\forall n, t_n(\mathbf{w}^t\mathbf{x}_n + b) \geq 1 \quad (1)$$

With these constraints, maximum margin is:

$$\arg\max_{\mathbf{w},b} \left\{ \frac{1}{\|\mathbf{w}\|} \right\}$$

Rescaling (2/2)

For support vectors, we decide:

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

That is (canonical representation):

$$\forall n, t_n(\mathbf{w}^t\mathbf{x}_n + b) \geq 1 \qquad (1)$$

With these constraints, maximum margin is:

$$\arg\max_{\mathbf{w},b}\left\{\frac{1}{\|\mathbf{w}\|}\right\}$$

Rescaling (2/2)

For support vectors, we decide:

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

That is (canonical representation):

$$\forall n, t_n(\mathbf{w}^t \mathbf{x}_n + b) \geq 1 \qquad (1)$$

With these constraints, maximum margin is:

$$\arg\max_{\mathbf{w},b} \left\{ \frac{1}{\|\mathbf{w}\|} \right\}$$

Quadratic Formulation

Maximize $\frac{1}{\|\mathbf{w}\|}$ under constraints (1).
$\iff$ Maximizing any decreasing function of $\|\mathbf{w}\|$ under (1).
$\iff$ Minimizing any increasing function of $\|\mathbf{w}\|$ under (1).
Finally:

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

$$\iff \quad \arg\min_{\mathbf{w},b}\left\{\frac{1}{2}\|\mathbf{w}\|^2\right\} \quad \text{under (1)}$$

M variables, N constraints.

Quadratic Formulation

Maximize $\frac{1}{\|\mathbf{w}\|}$ under constraints (1).

$\iff$ Maximizing any decreasing function of $\|\mathbf{w}\|$ under (1).

$\iff$ Minimizing any increasing function of $\|\mathbf{w}\|$ under (1).

Finally:

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

$$\iff \quad \arg\min_{\mathbf{w},b} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 \right\} \quad \text{under (1)}$$

M variables, N constraints.

Summary

SVM:

- maximize the margin,
- under classification constraints;

First formulation:

$$\begin{cases} \forall n, t_n(\mathbf{w}^T\mathbf{x}_n + b) - 1 \geq 0 \\ \arg\min_{\mathbf{w},b} \left\{\frac{1}{2}\|\mathbf{w}\|^2\right\} \end{cases}$$

Solution:

- linear constraints,
- optimization of a quadratic function,
- $\Longrightarrow$ *quadratic programming* $(O(M^3))$.

Summary

SVM:

- maximize the margin,
- under classification constraints;

First formulation:

$$\begin{cases} \forall n, t_n(\mathbf{w}^T \mathbf{x}_n + b) - 1 \geq 0 \\ \arg\min_{\mathbf{w},b} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 \right\} \end{cases}$$

Solution:

- linear constraints,
- optimization of a quadratic function,
- $\Longrightarrow$ *quadratic programming* ($O(M^3)$).

Summary

SVM:

- maximize the margin,
- under classification constraints;

First formulation:

$$\begin{cases} \forall n, t_n(\mathbf{w}^T \mathbf{x}_n + b) - 1 \geq 0 \\ \arg\min_{\mathbf{w},b} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 \right\} \end{cases}$$

Solution:

- linear constraints,
- optimization of a quadratic function,
- $\Longrightarrow$ *quadratic programming* ($O(M^3)$).

Dual Formulation

Introducing Lagrange Multipliers $a_n \geq 0$:

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

- colinear gradient for each constraint,
- first term should be smallest $\Longrightarrow$ minimize over $\mathbf{w}$, b,
- second term should be largest $\Longrightarrow$ maximize over $\mathbf{a}$.

Minimizing over $\mathbf{w}$, b:

- $\mathbf{w} = \sum_{n=1}^{N} a_n t_n \mathbf{x}_n$,
- $0 = \sum_{n=1}^{N} a_n t_n$.

Dual Formulation

Introducing Lagrange Multipliers $a_n \geq 0$:

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

- colinear gradient for each constraint,
- first term should be smallest $\Longrightarrow$ minimize over $\mathbf{w}$, b,
- second term should be largest $\Longrightarrow$ maximize over $\mathbf{a}$.

Minimizing over $\mathbf{w}$, b:

- $\mathbf{w} = \sum_{n=1}^{N} a_n t_n \mathbf{x}_n$,
- $0 = \sum_{n=1}^{N} a_n t_n$.

Autonomous Systems Lab

Dual Formulation

Then $L \to \tilde{L}$:

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

And we have to solve:

$$\begin{cases} \arg\max_{\mathbf{a}} \tilde{L}(\mathbf{a}) \\ \forall n, a_n \geq 0 \\ \sum_{n=1}^{N} a_n t_n = 0 \end{cases}$$

N variables, $N+1$ constraints: Quadratic Programming $O(N^3)$.
Equivalent solutions but complexity in N instead of M.

Dual Formulation

Then $L \rightarrow \tilde{L}$:

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

And we have to solve:

$$\begin{cases} \arg\max_{\mathbf{a}} \tilde{L}(\mathbf{a}) \\ \forall n, a_n \geq 0 \\ \sum_{n=1}^{N} a_n t_n = 0 \end{cases}$$

N variables, $N+1$ constraints: Quadratic Programming $O(N^3)$.
Equivalent solutions but complexity in N instead of M.

Dual Formulation

Then $L \to \tilde{L}$:

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

And we have to solve:

$$\begin{cases} \arg\max_{\mathbf{a}} \tilde{L}(\mathbf{a}) \\ \forall n, a_n \geq 0 \\ \sum_{n=1}^{N} a_n t_n = 0 \end{cases}$$

N variables, $N+1$ constraints: Quadratic Programming $O(N^3)$.
Equivalent solutions but complexity in N instead of M.

Support Vectors

Once solved, it can be shown that:

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

$\implies$ either $a_n = 0$ or $t_n y(\mathbf{x}_n) = 1$.

- $a_n = 0$: inactive constraint,
- $t_n y(\mathbf{x}_n) = 1$: support vector ($\mathcal{S}$).

Support Vector Machines

Normal vector:

$$\mathbf{w} = \sum_{n \in \mathcal{S}} a_n t_n \mathbf{x}_n$$

Bias:

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

Classification:

$$y(\mathbf{x}) = \sum_{n \in \mathcal{S}} a_n t_n \mathbf{x}^T \mathbf{x_n} + b$$

Need *only* support vectors after training.

Support Vector Machines

Normal vector:

$$\mathbf{w} = \sum_{n \in \mathcal{S}} a_n t_n \mathbf{x}_n$$

Bias:

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

Classification:

$$y(\mathbf{x}) = \sum_{n \in \mathcal{S}} a_n t_n \mathbf{x}^T \mathbf{x_n} + b$$

Need *only* support vectors after training.

Support Vector Machines

Normal vector:

$$\mathbf{w} = \sum_{n \in \mathcal{S}} a_n t_n \mathbf{x}_n$$

Bias:

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

Classification:

$$y(\mathbf{x}) = \sum_{n \in \mathcal{S}} a_n t_n \mathbf{x}^T \mathbf{x_n} + b$$

Need *only* support vectors after training.

Summary

SVM:

- binary classifier,
- linear separation,
- maximum margin;

Compact representation:

- active constraints,
- support vectors.

Feature Space

Back to our assumptions:

- training data points: $(\mathbf{x}_i, t_i)$,
- binary classification: $t_i \in -1, 1$,
- optional transformation in *feature space*: $\mathbf{x} \rightarrow \phi(\mathbf{x})$,
- *linearly separable* in feature space.

Why use a feature space? What if data is not linearly separable?
If the data is not linearly separable, use a feature space in which they are!

$< Video >$

Kernel Trick

The dual representation is:

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

Kernel Trick

The dual representation is:

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

Kernel Trick

The dual representation is:

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

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

Kernel Trick: replace a dot product by a kernel.
Kernel: symmetric positive semi-definite function.
Mercer's theorem: any (continuous) such kernel $\rightarrow$ dot product in a high-dimensional space.

Constructing a Kernel

- define ϕ,
- define $k(\mathbf{x}, \mathbf{x}')$ directly (check that it is symmetric positive semi-definite),
- $\mathbf{x}^T A \mathbf{x}'$ with A a symmetric positive semi-definite matrix,
- adapt an existing kernel:
  - $ck(\mathbf{x}, \mathbf{x}')$ if $c > 0$,
  - $f(x)k(\mathbf{x}, \mathbf{x}')f(\mathbf{x}')$ for any f,
  - $q(k(\mathbf{x}, \mathbf{x}'))$ if q is a polynomial with positive coefficients,
  - $\exp(k(\mathbf{x}, \mathbf{x}'))$,
  - $k_1(\mathbf{x}, \mathbf{x}') + k_2(\mathbf{x}, \mathbf{x}')$, $k_1(\mathbf{x}, \mathbf{x}')k_2(\mathbf{x}, \mathbf{x}')$,
  - $k(\phi(\mathbf{x}), \phi(\mathbf{x}')$,
  - ...

Properties and Examples

Stationary kernel:

$$k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x} - \mathbf{x}')$$

Homogeneous:

$$k(\mathbf{x}, \mathbf{x}') = k(\|\mathbf{x} - \mathbf{x}'\|)$$

Gaussian Kernel:

$$k(\mathbf{x}, \mathbf{x}') = \exp - \frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}$$

Radial Basis Functions:

$$f(\mathbf{x}) = \sum_{n=1}^{N} w_n h(\|\mathbf{x} - \mathbf{x_n}\|)$$

.

Summary

Classes not directly separable:

- usage of a kernel,
- change of feature space,
- change in dimensionality,
- can be in a infinite dimension (Gaussian kernel).

Overlapping Class Distributions

In practice:

- outlier, noise...
- overlapping classes;
- no separability in the feature space.

Introduction of slack variables ξ_n:

$$\begin{cases} \xi_n = 0 & \text{if data inside margin boundary} \\ \xi_n = |t_n - y(\mathbf{x}_n)| & \text{otherwise} \end{cases}$$

Handling Slack Variables

Classification constraints:

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

Relaxation of the constraint: soft margin.
Maximize the margin while penalizing points on the wrong side:

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

with $C > 0$: trade-off between margin and slack penalty.
The dual formulation is similar:

$$\begin{cases} \arg\max_{\mathbf{a}} \tilde{L}(\mathbf{a}) \\ \forall n, C \geq a_n \geq 0 \quad \text{(box constraints)} \\ \sum_{n=1}^{N} a_n t_n = 0 \end{cases}$$

Multiclass SVMs

SVMs: two class classifier.
For $K > 2$ classes:

- K classifiers: one vs rest;
  - but one point could be assigned to several classifiers
- chaging objective function to learn all classes together;
- $K(K-1)/2$ classifiers: one versus one
  - more costly in learning, and classification
- pairwise classifiers organized on directed acyclic graphs (DAGSVN);
- ...

Summary

SVMs:

- binary classifier,
- linear separation;

Kernel trick:

- change space to have non linear boundary;

Generalization to:

- overlapping classes,
- multiple classes.