MACHINE LEARNING

Classifiers: Support Vector Machine and extensions

$\nu$-SVM.

Relevance Vector Machine

Transductive Support Vector Machine

Support Vector Clustering
What is Classification?

Training set must be as unambiguous as possible. Not easy, especially as members of different classes may share similar attributes.

Learning implies generalization; which of the features of each member of class makes the class most distinguishable from the other classes.
Multi-Class Classification

Whenever possible, the classes should be balanced

**Garbage model:**
Male adult versus anything that is neither a female adult nor a child

*Classes can no longer be balanced!*
Classifiers

There is a plethora of classifiers, e.g:

- Neural networks (feed-forward with backpropagation, multi-layer perceptron)
- Decision trees (C4.5, random forest)
- Kernel methods (support vector machine, gaussian process classifier)
- Mixtures of linear classifiers (boosting)

*In this class, we will see only SVM and Boosting for mixture of classifiers*

Each classifier type has its pros and cons:

- Complex model: embed non-linearity but heavy computation
- Simple models: often high number of models, hence high stack memory
- Number of hyperparameters: high \(\rightarrow\) extensive crossvalidation to determine the optimal classifier
- Some of the classifiers come with guarantees for global optimal solution; other have only local optimality guarantee
Support Vector Machine (SVM)

Brief history:
SVM was invented by Vladimir Vapnik. It started with the invention of the statistical learning theory (Vapnik 1979). The current form of SVM was presented in (Boser, Guyon and Vapnik 1992) and Cortes and Vapnik (1995).

Textbooks:

A good survey of the theory behind SVM is given in *Support Vector Machines and other Kernel Based Learning methods* by Nello Cristianini and John-Shawe Taylor. An easy introduction to SVM is given in *Learning with Kernels* by Bernhard Scholkopf and Alexander Smola.
Support Vector Machine (SVM)

SVM was applied to numerous classification problems:
- Computer vision (face detection, object recognition, feature categorization, etc)
- Bioinformatics (categorization of gene expression, of microarray data)
- WWW (categorization of websites)
- Production (control of quality, detection of defaults)
- Robotics (categorization of sensor readings)
- Finance (bankruptcy prediction)

The success of SVM is mainly due to:
- Its ease of use (lots of software available, good documentation)
- Excellent performance on variety of datasets
- Good solvers making optimization (learning phase) very quick
- Very fast at retrieval time – does not hinder practical applications
Linear Classifiers

Class label $y=\{-1;1\}$

- denotes -1
- denotes +1

$y = f(x; w, b) = \text{sgn} (\langle w, x \rangle + b)$

$\langle w, x \rangle$: Inner product, i.e. $w^T x$

How would you classify this data?
Linear Classifiers

Class label $y=\{-1; 1\}$

- denotes -1
- denotes +1

How would you classify this data?

$y = f(x; w, b) = \text{sgn}\left(\langle w, x \rangle + b \right)$
Linear Classifiers

Class label $y=\{-1;1\}$

- denotes -1
- denotes +1

$$y = f(x; w, b) = \text{sgn}\left(\langle w, x \rangle + b\right)$$
Linear Classifiers

Class label $y=\{-1;1\}$

- denotes -1
- denotes +1

$$y = f(x; w, b) = \text{sgn} \left( \langle w, x \rangle + b \right)$$

How would you classify this data?
Linear Classifiers

Class label $y = \{-1; 1\}$

- denotes -1
- denotes +1

$$y = f(x; w, b) = \text{sgn}(\langle w, x \rangle + b)$$

Any of these would be fine..

..but which is best?
Class label $y = \{-1; 1\}$
- denotes -1
- denotes +1

Define the margin of a linear classifier as the width that the boundary could be increased by before hitting a datapoint.

$$y = f(x; w, b) = \text{sgn}(\langle w, x \rangle + b)$$
The maximum margin linear classifier is the linear classifier with the maximum margin.

This is the simplest kind of SVM (Called an LSVM).
Class label $y = \{-1; 1\}$

- denotes -1
- denotes +1

Support Vectors are those datapoints that the margin pushes up against.

The maximum margin linear classifier is the linear classifier with the, um, maximum margin.

This is the simplest kind of SVM (Called an LSVM)
Learning optimal classifier

Can be formulated as a convex optimization

\[
\min_{w,b} \frac{1}{2} \|w\|^2
\]

\[
\langle w, x^i \rangle + b \geq 1 \text{ when } y^i = +1
\]

\[
\langle w, x^i \rangle + b \leq -1 \text{ when } y^i = -1
\]

\[
\|x^1 - x^2\| = \frac{2}{\|w\|}
\]

Maximizes the margin

Norm of \( w \) inversely proportional to distance of SV on either side of the margin.
Learning optimal classifier

Can be formulated as a convex optimization

\[
\begin{align*}
\min_{w,b} & \frac{1}{2} \|w\|^2 \\
\langle w, x^i \rangle + b & \geq 1 \text{ when } y^i = +1 \\
\langle w, x^i \rangle + b & \leq -1 \text{ when } y^i = -1
\end{align*}
\]

Class label \( y = \{-1; 1\} \)

- denotes -1
- denotes +1

\( M \) (nm of datapoints) constraints to ensure that the points are well classified.
Determining the Optimal Separating Hyperplane

The decision function is then expressed in terms of the support vectors:

\[ f(x) = \text{sgn}\left(\langle w, x \rangle + b \right) \]

\[ = \text{sgn}\left(\sum_{i=1}^{M} \alpha_i y_i \langle x, x^i \rangle + b \right) \]

\[ \alpha_1, \ldots, \alpha_M, b \in \mathbb{R} \]

Parameters learned during optimization
Non-Linear Classification

What if the points in the input space cannot be separated by a linear hyperplane?
From linear to non-linear classification

The decision function is then expressed in terms of the support vectors:

\[ f(x) = \text{sgn} \left( \langle w, x \rangle + b \right) \]

\[ = \text{sgn} \left( \sum_{i=1}^{M} \alpha_i y_i \langle x, x^i \rangle + b \right) \quad \alpha_1, \ldots, \alpha_M, b \in \mathbb{R} \]
From linear to non-linear classification

The decision function is then expressed in terms of the support vectors:

\[ f(x) = \text{sgn}(\langle w, x \rangle + b) \]

\[ = \text{sgn} \left( \sum_{i=1}^{M} \alpha_i y_i \langle \phi(x), \phi(x^i) \rangle + b \right) \]
From linear to non-linear classification

The decision function is then expressed in terms of the support vectors:

\[ f(x) = \text{sgn}(\langle w, x \rangle + b) \]

\[ = \text{sgn}\left(\sum_{i=1}^{M} \alpha_i y_i k(x, x^i) + b\right) \]

Replace with the kernel

\[ k : X \times X \rightarrow \mathbb{R} \]

\[ k(x^i, x^j) \rightarrow \langle \phi(x^i), \phi(x^j) \rangle. \]
How to read out the result of SVM

Color gradient = distance to the hyperplane

Hyperplane

Support vectors

The margin: region of equal distance on both side of hyperplane
Effect of the width of Gaussian kernel

RBF kernel width = 0.001; C = 1000; 113 support vectors out of 345 total nm of datapoints
RBF kernel width = 0.008; C=1000; 64 support vectors out of 345 total nm of datapoints
Effect of the width of Gaussian kernel

RBF kernel width = 0.02; C=1000; 33 support vectors out of 345 total nm of datapoints
Different optimization runs end up with different solutions.

Several combination of the support vectors yield the same optimum.
Different optimization runs end up with different solutions

Several combination of the support vectors yield the same optimum
SVM: Exercises

1) Consider two datapoints (one for each class)
   • How many support vectors do you have?
   • Discuss the shape of the separating hyperplane for the RBF and polynomial kernel considering $p=1$ and $p=2$.

2) Consider the set of 4 datapoints (2 per class) in the drawing
   • How many support vectors do you have with RBF kernel?
   • Discuss the shape of the separating hyperplane for the RBF and polynomial kernels considering $p=2$.
   • Discuss the effect of the choice of coordinate for the points in the polynomial kernel.
Extensions to traditional SVM
The hyperparameters of SVM

\[
\min_{w, \xi} \left( \frac{1}{2} \|w\|^2 + \frac{C}{M} \sum_{j=1}^{M} \xi_j \right)
\]

u.c.

\[
y^j \left( w^T \cdot x^j + b \right) \geq 1 - \xi_j,
\]

\[
\xi_j \geq 0 \quad \forall \ j = 1, \ldots, M
\]

C that determines the costs associated to incorrectly classifying datapoints is an open parameter of the optimization function.
Effect of the penalty factor C

RBF kernel width=0.20; C=1000; several misclassified datapoints
Effect of the penalty factor C

RBF kernel width=0.20; C=2000; fewer misclassified datapoints
Hyperparameters for SVM

The original objective function:

\[
\min_{w, \xi} \left( \frac{1}{2} \| w \|^2 + \frac{C}{M} \sum_{j=1}^{M} \xi_j \right)
\]

Determining C may be difficult in practice.
\( \nu \text{-SVM} \)

Introduce a new variable \( \rho \) to control for the lower bound on \( \|w\| \) and add a hyperparameter \( 0 \leq \nu \leq 1 \) to control for its effect in the objective function. The optimization problem becomes:

\[
\min_{w, \xi, \rho} \left( \|w\|^2 - \nu \rho + \frac{1}{M} \sum_{i=1}^{M} \xi_i \right),
\]

subject to \( y^i \left( \langle w, x^i \rangle + b \right) \geq \rho - \xi_i \)

and \( \xi_i \geq 0, \rho \geq 0 \).
$\nu$-SVM

$\nu$ is an upper bound on the fraction of margin error (i.e. the number of datapoints misclassified in the margin)

$\nu$ is a lower bound on the ratio: support vectors / number of datapoints.

$$\nu \leq \frac{p}{M}, \quad p: \text{number of SV}$$
\( \nu \)-SVM: Exercise

Show that \( \nu \) is a lower bound on the ratio:

\[
\nu \leq \frac{p}{M}, \quad p: \text{number of SV, } M: \text{number of datapoints}
\]

Hint: The dual problem is

\[
\min_{\alpha_1, \ldots, \alpha_M} \left( L(\alpha_1, \ldots, \alpha_M) = -\frac{1}{2} \sum_{i,j=1}^{M} \alpha_i \alpha_j k(x^i, x^j) \right)
\]

subject to

\[
\sum_{i=1}^{M} \alpha_i y_i = 0; \quad 0 \leq \alpha_i \leq 1/M; \quad \sum_{i=1}^{M} \alpha_i \geq \nu
\]
\( \nu\)-SVM: Example of effect of choice of \( \nu \)

\( \nu\)-svm \( \nu=0.001 \), rbf kernel width 0.1
$\nu$-SVM: Example of effect of choice of $\nu$

Increase in the number of SV-s with $\nu = 0.2$
\( \nu \text{-SVM: Example of effect of choice of } \nu \)

Increase in the number of SV-s with \( \nu=0.9 \)
\( \nu\)-SVM: Example of effect of choice of \( \nu \)

Good classification with \( \nu = 0.2 \)
\( \nu \)-SVM: Example of effect of choice of \( \nu \)

Increase in the error with \( \nu = 0.9 \)
Other extensions to SVM

• Relevance Vector Machine (RVM), see supplement (Tipping, IJML 2001) – sparse classification technique


• Support Vector Clustering (SVC), see supplement (Ben-hur et al, IJML 2001)
Relevance Vector Machine (RVM) (sparse SVM)
Relevance Vector Machine (RVM)

RVM was offered to address three shortcomings of standard SVM:

1) Even though SVM usually results in a relatively small number of support vectors compared to total number of data-points, nothing ensures that a sparse solution is obtained. The number of SV tends to grow “linearly” with the number of training datapoints.

2) Unlike other bayesian techniques (e.g. GMM classification using naïve Bayes), SVM’s prediction are not accompanied by a metric measuring the confidence of the model’s prediction.

3) SVM requires also to find hyper-parameters (C, ν) and to have special form for the basis function (the kernel must satisfy the Mercer conditions).

RVM relaxes assumption 3 and takes a Bayesian approach to estimate the model’s parameters. The Bayesian framework captures the uncertainty of the prediction. It also results in a sparse version of classical SVM.
Relevance Vector Machine

Start from the solution of SVM (dropping the sign function – provides regression solution first, see slides on non-linear regression)

\[ y(x) = f(x) = \sum_{i=1}^{M} \alpha_i k(x, x^i) + b \]

Rewrite the solution of SVM as a linear combination over M basis functions

A sparse solution has a majority of entries with alpha zero.

In the (binary) classification case, \( y \in [0; 1] \).
In the regression case, \( y \in \mathbb{R} \).
Relevance Vector Machine

The problem consists in finding the parameters $\alpha$. The problem is made easy in that it is *linear in the parameters*. 

Rewrite the solution of SVR in a compact form such that the problem is linear in the parameters:

$$
y(x) = f(x) = \sum_{i=1}^{M} \alpha_i k(x, x^i) + b = \alpha_0
$$

$$
y(x) = \alpha^T \Psi(x), \quad \Psi(x) = \begin{bmatrix} \psi_0(x) & \psi_1(x) & \ldots & \psi_M(x) \end{bmatrix}^T, \quad \psi_0(x) = 1.
$$
Relevance Vector Machine

Rewrite the solution of SVR in a compact form such that the problem is linear in the parameters:

\[ y(x) = f(x) = \sum_{i=1}^{M} \alpha_i k(x, x^i) + b = \alpha_0 \]

\[ y(x) = \alpha^T \Psi(x), \quad \Psi(x) = \left[ \psi_0(x) \; \psi_1(x) \ldots \ldots \psi_M(x) \right]^T, \; \psi_0(x) = 1. \]

Take a Bayesian approach and assume that all samples \( y^i \) are i.i.d and that they are measurements of the real value \( \alpha^T \Psi(x^i) \) subjected to white noise \( \varepsilon \), i.e.:

\[ y^i(x) = \alpha^T \Psi(x^i) + \varepsilon, \; \varepsilon \sim N(0, \sigma_{\varepsilon}) \]
Relevance Vector Machine

Since the measurements are i.i.d, the likelihood of the model is given by:

\[
L(\alpha) = \prod_{i=1}^{M} p\left(y^i \mid x^i; \alpha, \sigma_\varepsilon\right) \sim \prod_{i=1}^{M} e^{-\frac{1}{\sigma_\varepsilon^2} \|y^i - \alpha \Psi(x^i)\|^2}
\]

**Question:** What is the result if we estimate the \(\alpha\) parameters through maximum likelihood?

Doing maximum likelihood would lead to overfitting, as we have as many parameters as data points (or more if considering the variance of the noise too).

With rbf kernel for basis functions, all alphas are +/-1 putting one rbf function on each data point.

\(\rightarrow\) idea: approximate the distribution of the alpha with a probability density function which reduces the number of parameters to estimate.
Relevance Vector Machine

Start from the solution of SVM

\[ y(x) = f(x) = \sum_{i=1}^{M} \alpha_i k(x, x^i) + b = \alpha_0 \]

\[ y(x) = \alpha^T \Psi(x), \quad \Psi(x) = \begin{bmatrix} \psi_0(x) & \psi_1(x) & \ldots & \psi_M(x) \end{bmatrix}^T, \quad \psi_0(x) = 1. \]

Introduces a prior on the distribution of the parameters, i.e. \( p(\alpha) \), to prevent them from taking arbitrary values.

Sparsity is obtained when the distribution is sharply peaked at zero, e.g. \( E\{p(\alpha)\} \sim 0 \) and \( \text{var}\{p(\alpha)\} \ll 1 \)
Relevance Vector Machine

Prior distribution is zero-mean with variance $\sigma$.

$\sigma$ is a set of $M$ parameters that controls for the breadth of values taken by the $\alpha$:

$$p(\alpha_i) = \sim N(\alpha_i; 0, \sigma_i)$$

Question 1: Why zero-mean?

Equivalent to the KKT condition: $\sum_i y_i \alpha_i = 0$

Solving the problem now requires estimating the optimal set of parameters, i.e. all the $\{\alpha_i, \sigma_i, \epsilon \}_{i=0..M}$ and $\sigma_\epsilon$

One cannot compute the optimal alpha in closed form. One must use an iterative procedure similar to expectation maximization. The procedure differs depending on whether we consider the classification or regression case. (see Tipping 2001, supplementary material, for details).
Relevance Vector Machine
Relevance Vector Machine

RVM with kernel width = 0.01
477 datapoints, 19 support vectors
Relevance Vector Machine

SVM with $C=1000$, kernel width = 0.01
477 datapoints, 51 support vectors
Notice the notion of uncertainty of the model encapsulated in the distribution (shadings of grey and red)
Transductive SVM
(semi-supervised clustering)
Transductive Support Vector Machine

Transductive SVM: learn from partially labeled datapoints
Inductive SVM: learn from fully labeled datapoints (regular SVM)

Which hyperplane separates best?
Transductive Support Vector Machine

Transductive SVM: learn from partially labeled datapoints
Inductive SVM: learn from fully labeled datapoints (regular SVM)

This is the solution found by classical (inductive) SVM.

It maximizes the margin between the two labelled points in each class.

This ignores the distribution of unlabeled points.

How can one take into account the unlabeled points?
Transductive Support Vector Machine

Transductive SVM: learn from partially labeled datapoints
Inductive SVM: learn from fully labeled datapoints (regular SVM)

We want:
1) zero error on labeled points
2) unlabeled points well separated

Mix between:
1) classification problem
2) clustering problem

→ use labels in 1 to guide 2.
Transductive Support Vector Machine

Consider the dataset composed of:

\[ \{x^i, y^i\} : \text{set of labeled datapoints} \]

\[ \{\tilde{x}^i, \tilde{y}^i\} : \text{set of un-labeled datapoints, } \tilde{y}^i : \text{unknown!} \]

Formulates the constrained optimization problem:

\[ \min \|w\|^2 \quad \text{with the following constraints:} \]

\[ \forall i = 1^M \left\{ \begin{array}{l}
  y^i \langle w, x^i \rangle \geq 1 \\
  \tilde{y}^i \langle w, \tilde{x}^i \rangle \geq 1 \\
  \tilde{y}^i \in \{-1; +1\}
\end{array} \right. \]

Requests perfect classification on labeled points

Searches for the labels \( \tilde{y}^i \) such that unlabeled points live on either side of the margin.
Transductive Support Vector Machine

Which hyperplanes satisfy these constraints?

\[ \bigwedge_{i=1}^{M} \begin{cases} \tilde{y}^i \langle w, \tilde{x}^i \rangle \geq 1 \\ \tilde{y}^i \in \{-1; +1\} \end{cases} \]

Searches for the labels \( \tilde{y}^i \) such that unlabeled points live on either side of the margin.
Transductive Support Vector Machine

Requests perfect classification on labeled points

\[
\forall_{i=1}^{M} \begin{cases} 
  y^i \langle w, x^i \rangle \geq 1 \\
  \tilde{y}^i \langle w, \tilde{x}^i \rangle \geq 1 \\
\end{cases}
\]

\[\tilde{y}^i \in \{-1; +1\}\]

Searches for the labels \(\tilde{y}^i\) such that unlabeled points live on either side of the margin.

Adding labeled points reduces the possibilities
Transductive Support Vector Machine

Formulates the constrained optimization problem:
\[
\min \quad \text{subject to} \quad y^i \langle w, x^i \rangle \geq 1 \quad \forall i = 1, \ldots, M
\]

Requests perfect classification on labeled points

Adding labeled points reduces the possibilities

Searches for the labels \( \tilde{y}^i \) such that unlabeled points live on either side of the margin.
Transductive Support Vector Machine

Adding labeled points reduces the possibilities

\[ \forall_{i=1}^{M} \begin{cases} 
    y^i \langle w, x^i \rangle \geq 1 \\
    \tilde{y}^i \langle w, \tilde{x}^i \rangle \geq 1 \\
    \tilde{y}^i \in \{-1; +1\} 
\end{cases} \]

Requests perfect classification on labeled points

Searches for the labels \( \tilde{y}^i \) such that unlabeled points live on either side of the margin.
Transductive Support Vector Machine

\[ \forall_{i=1}^{M} \begin{cases} y^i \langle w, x^i \rangle \geq 1 \\ \tilde{y}^i \langle w, \tilde{x}^i \rangle \geq 1 \\ \tilde{y}^i \in \{-1; +1\} \end{cases} \]

Requests perfect classification on labeled points

Searches for the labels \( \tilde{y}^i \) such that unlabeled points live on either side of the margin.

Adding labeled points reduces the possibilities.
Transductive Support Vector Machine

\[ \forall_{i=1}^{M} \begin{cases} 
  y_i \langle w, x^i \rangle \geq 1 \\
  \tilde{y}_i \langle w, \tilde{x}^i \rangle \geq 1 \\
  \tilde{y}_i \in \{-1; +1\} 
\end{cases} \]

Requests perfect classification on labeled points

Searches for the labels \( \tilde{y}_i \) such that unlabeled points live on either side of the margin.

Adding labeled points reduces the possibilities
Transductive Support Vector Machine: Exercise

Consider the set of 4 datapoints with two unlabelled datapoints and assume that you use linear SVM.

a) Draw a boundary line that minimizes: \( \|w\| \)

and satisfies all the constraints:

\[
\begin{align*}
\forall_{i=1}^{M} & \quad y^i \langle w, x^i \rangle \geq 1 \\
\forall_{i=1}^{M} & \quad \tilde{y}^i \langle w, \tilde{x}^i \rangle \geq 1 \\
\tilde{y}^i & \in \{-1; +1\}
\end{align*}
\]

b) Find a configuration of the points where the two unlabeled datapoint are misclassified.
Transductive Support Vector Machine

Can be extended to give some slack to the constraints.

The constrained optimization problem becomes:

\[
\min \|w\| + C \sum_{i=1}^{M} \xi_i + \tilde{C} \sum_{i=1}^{M} \tilde{\xi}_i \quad \text{with the following constraints:}
\]

\[
\begin{align*}
\forall i \in \{1, \ldots, M\} \quad & y^i \langle w, x^i \rangle \geq 1 + \xi_i \\
& \tilde{y}^i \langle w, \tilde{x}^i \rangle \geq 1 + \tilde{\xi}_i \\
\end{align*}
\]

Slack variables and associated constraints:

\[
\xi_i \geq 0 \quad \tilde{\xi}_i \geq 0
\]
Support Vector Clustering (SVC)
Support Vector Clustering

**Idea:**
- Data points are mapped from data space to a high dimensional feature space using a Gaussian kernel (RBF kernel).
- In feature space, we look for the smallest sphere that encloses the image of the data.

Original dataset $X$ $\phi$ Images $\phi(X)$
Support Vector Clustering

**Idea:**
- Data points are mapped from data space to a high dimensional feature space using a Gaussian kernel (RBF kernel).
- In feature space, we look for the smallest sphere that encloses the image of the data.
- This sphere is mapped back to data space, where it forms a set of contours which enclose the data points. These contours are interpreted as cluster boundaries.
Support Vector Clustering

Requesting that all points $x^j, j = 1...M$, be in a sphere in feature space gives the following constraint:

$$\left\| \phi(x^j) - \mu \right\|^2 \leq R^2, \quad \forall x^j, j = 1...M$$

$\mu$: center of the sphere, $R$: radius of the sphere

To soften the constraint, one can add some slack variable $\xi^i$:

$$\left\| \phi(x^j) - \mu \right\|^2 \leq R^2 + \xi^i, \quad \text{with } \xi^i \geq 0$$

To find the tightest sphere enveloping the points means that one wants the smallest $R$.

→ This can be formulated as a constrained optimization problem.
Support Vector Clustering

\[ \min_{R, \xi} \left( R^2 + C \sum_{j=1}^{M} \xi_j \right) \quad (C \geq 0: \text{hyperparameter}) \]

under the constraints

\[ \|\phi(x^j) - \mu\|^2 \leq R^2 + \xi_j, \quad \forall x^j, j = 1...M \]

\[ \xi_j \geq 0, \quad j = 1...M \]

The Lagrangian is:

\[ L(R, \xi, \beta, \eta) = R^2 + C \sum_{j=1}^{M} \xi_j - \beta_j \left( R^2 + \xi_j - \|\phi(x^j) - \mu\|^2 \right) - \sum_{j=1}^{M} \eta_j \xi_j \]

Lagrange multipliers for all the constraints
Support Vector Clustering

This problem can be solved as for the standard SVM problem. It has one global optimum. The quadratic form for the constraint yields a series of inner product of the form \( \langle \phi(x^i), \phi(x^j) \rangle \), which we can replace by the kernel form: \( k(x^i, x^j) = \langle \phi(x^i), \phi(x^j) \rangle \).

(see Ben-Hur et al, IJML 2001 - supplementary material for the derivation)

The Lagrangian is:

\[
L(R, \xi, \beta, \eta) = R^2 + C \sum_{j=1}^{M} \xi^i - \beta_j \left( R^2 + \xi^i - \|\phi(x^j) - \mu\|^2 \right) - \sum_{j=1}^{M} \eta_j \xi^i
\]

Lagrange multipliers for all the constraints
Support Vector Clustering

For a query point $x$, we can write the distance of its image in feature space from the center of the sphere as:

$$R^2(x) = k(x, x) - 2 \sum_{j=1}^{M} \beta_j k(x, x^j) + \sum_{i,j}^{M} \beta_i \beta_j k(x^i, x^j)$$

In Kernel K-means, the cluster boundaries are determined by considering all the datapoints. Each datapoint has "equal weight" within one cluster. Each cluster's influence is weighted by the number of datapoints in the cluster.

$$d(x, C^k) = k(x, x) - \frac{2 \sum_{x^j \in C^k} k(x, x^j)}{m_k} + \frac{\sum_{x^j, x^{j'} \in C^k} k(x^j, x^{j'})}{(m_k)^2}$$

In SVC, only a selected subset of datapoints are used to compute the cluster boundaries. The influence of each datapoint is weighted by its $\beta_i$.

The optimization process of SVC determines the influence of the datapoints (the $\beta_i$ are variables in the optimization).

→ SVC is a sparse version of Kernel K-means
Support Vector Clustering

For a query point $x$, we can write the distance of its image in feature space from the center of the sphere as:

$$R^2(x) = k(x, x) - 2 \sum_{j=1}^{M} \beta_j k(x, x^j) + \sum_{i,j} \beta_i \beta_j k(x^i, x^j)$$

Cluster assignment is then determined:

1) either by looking at the isolines solution of $R(x) = cst$.

2) or one can build a binary similarity matrix $S$ for each pair of datapoints

$$S_{ij} = \begin{cases} 1 & \text{if } R^2(x^i, x^j) < cst \\ 0 & \text{otherwise} \end{cases}$$

$\rightarrow$ we are back to spectral clustering. The clusters are found through a spectral decomposition of the Laplacian.

The coefficients $\beta_i$ are non zero for the support vectors.
Support Vector Clustering

As the kernel width is decreased, the number of disconnected contours in data space increases, leading to an increasing number of clusters.
Summary

Support Vector Machines provides a framework for a variety of computation including classification, semi-supervised clustering, unsupervised clustering and regression (see in a few weeks).

SVM uses convex optimization (finds the optimum of a convex problem; it is hence ensured to find the global optimum).

Variants on SVM have been offered to:
- Restrict the number of support vectors ($\nu$-SVM, RVM)
- Deal with partially labelled or unlabeled data (Transductive-SVM)

This class was meant to provide you with an enough understanding of the conceptual and mathematical similarities between these different techniques, so as to develop variants on these yourself later on.