Advanced Methods for Predictive Sequence Analysis

Gunnar Rätsch[1], Cheng Soon Ong[1,2], Petra Philips[1]

[1] Friedrich Miescher Laboratory, Tübingen
[2] MPI for Biological Cybernetics, Tübingen

Vorlesung WS 2007/2008
Eberhard-Karls-Universität Tübingen

30 October 2007

http://www.fml.mpg.de/raetsch/lectures/amsa07

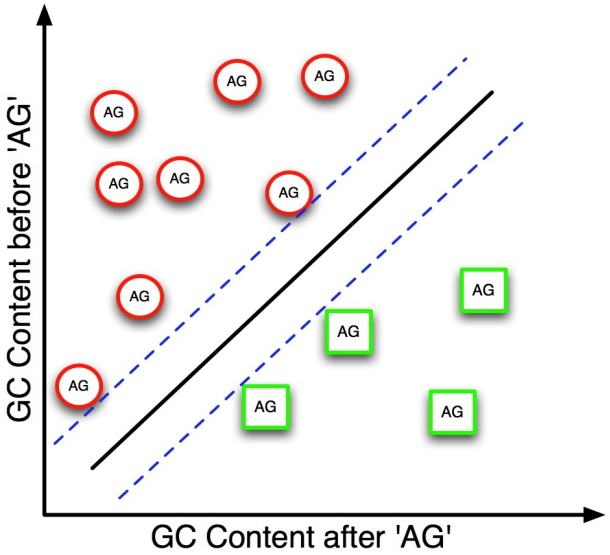
Recall: Splice Sites

- Given: Potential acceptor splice sites

```
AAACAAATAAGTAACTAATCTTTTAGGAAGAACGTTTCAACCATTTTGAG
AAGATTAAAAAAAAACAAATTTTTAGCATTACAGATATAATAATCTAATT
CACTCCCCAAATCAACGATATTTTAGTTCACTAACACATCCGTCTGTGCC
TTAATTTCACTTCCACATACTTCCAGATCATCAATCTCCAAAACCAACAC
```

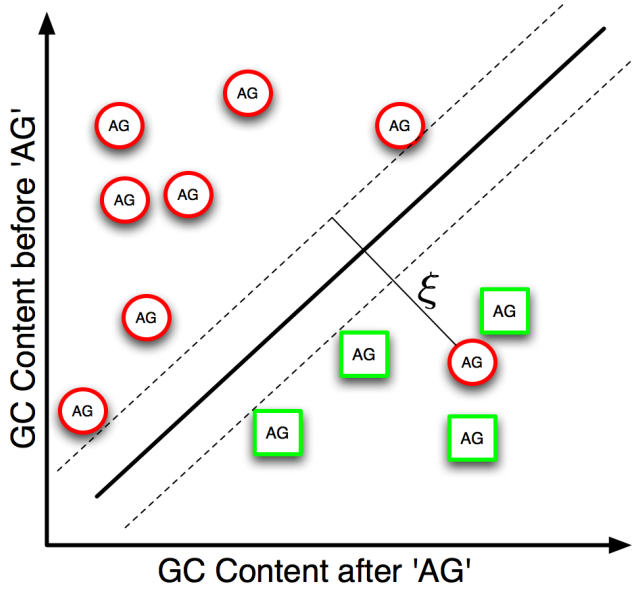
intron **exon**

- Goal: Rule that distinguishes true from false ones



Linear classifiers with large margin

Our aim today



Minimize

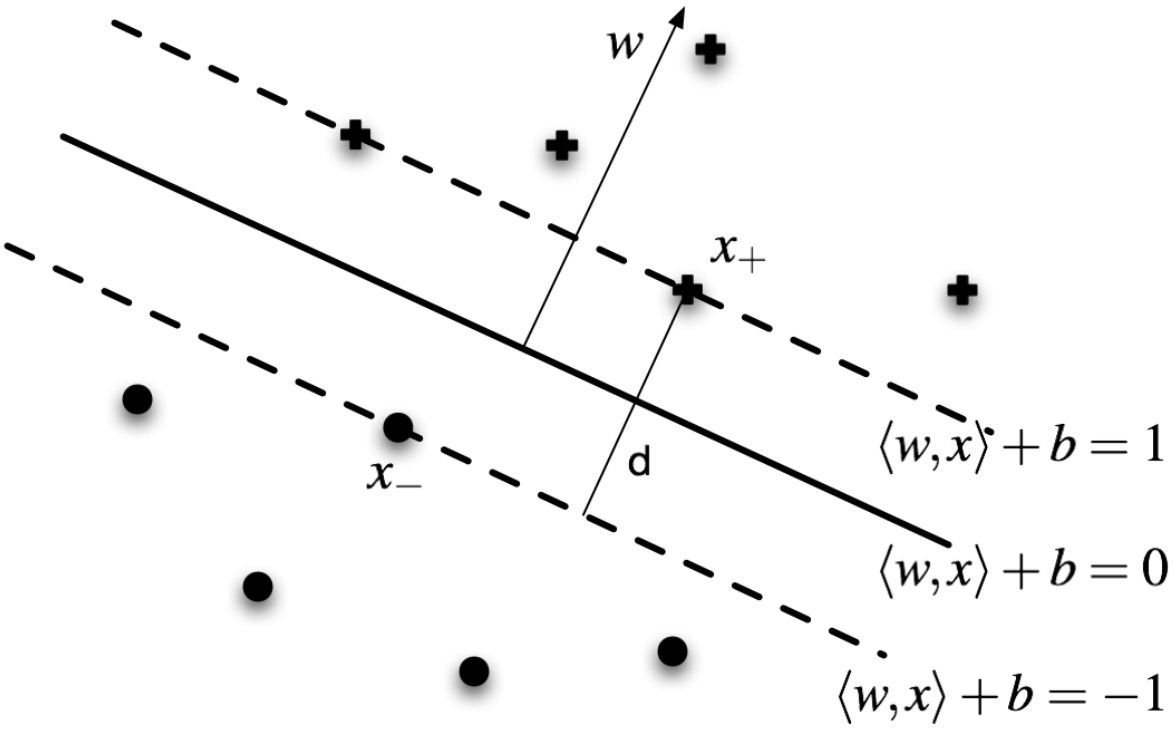
$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i$$

Subject to

$$y_i(\langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle + b) \geqslant 1 - \xi_i$$
$$\xi_i \geqslant 0$$
for all $i = 1, \ldots, N$.

- The examples on the margin are called support vectors [Vapnik, 1995]
- Called the soft margin SVM or the C-SVM [Cortes and Vapnik, 1995]

Margin Maximization



Margin maximization is equivalent to minimizing $\|\mathbf{w}\|$.

SVM: Geometric View

Find a function

$$f(\mathbf{x}) = \text{sign}(\langle \mathbf{w}, \mathbf{x} \rangle + b),$$

where $\mathbf{w}$ and b are found by

$$\begin{aligned} &\underset{\mathbf{w},b}{\text{minimize}} && \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i && (1) \\ &\text{subject to} && y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geqslant 1 - \xi_i \text{ and } \xi_i \geqslant 0 && (2) \\ & && \text{for all } i = 1, \ldots, N. \end{aligned}$$

Objective function (1) Maximize the margin.

Constraints (2) Correctly classify the training data.

The slack variables ξ allow points to be in the margin, but penalize them in the objective.

Soft Margin SVM

$$\begin{aligned} \text{minimize}_{\mathbf{w},b} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i \\ \text{subject to} \quad & y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geqslant 1 - \xi_i \text{ for all } i = 1, \ldots, N. \\ & \xi_i \geqslant 0 \text{ for all } i = 1, \ldots, N \end{aligned}$$

Objective function

By minimizing the squared norm of the weight vector, we maximize the margin.

Constraints

We can express the constraints in terms of a loss function.

SVM: Loss View

$$\underset{\mathbf{w},b}{\text{minimize}} \frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{N} \ell(f_{\mathbf{w},b}(\mathbf{x}), y_i),$$

where

$$\ell(f_{\mathbf{w},b}(\mathbf{x}), y_i) := C \max\{0, 1 - y_i(f_{\mathbf{w},b}(\mathbf{x}))\}$$
$$f_{\mathbf{w},b}(\mathbf{x}) := \mathbf{w}^\top \mathbf{x}_i + b.$$

The above loss function is known as the hinge loss.

Regularizer = $\frac{1}{2}\|\mathbf{w}\|^2$.

Empirical Risk = $\sum_{i=1}^{N} \ell(\mathbf{w}^\top \mathbf{x}_i + b, y_i)$.

How much does a mistake cost us?

Risk and Regularization

Basic Notion

In general, we can think of an SVM as optimizing a particular cost function,

$$\Omega(w) + \mathbf{R}_{emp}(w),$$

where $\mathbf{R}_{emp}(w)$ is the empirical risk measured on the training data, and $\Omega(w)$ is the regularizer.

Regularization

The regularizer is a function which measures the complexity of the function.

General principle

There is a trade-off between fitting the training set well (low empirical risk) and having a "simple" function (small regularization term).

Loss Functions

0-1 loss

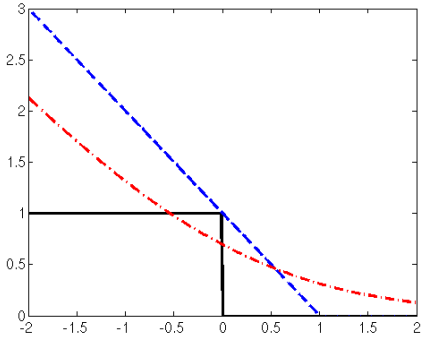
$$\ell(f(\mathbf{x}_i), y_i) := \begin{cases} 0 & y_i = f(\mathbf{x}_i) \\ 1 & y_i \neq f(\mathbf{x}_i) \end{cases}$$

hinge loss

$$\ell(f(\mathbf{x}_i), y_i) := \max\{0, 1 - y_i f(\mathbf{x}_i)\}$$

logistic loss

$$\ell(f(\mathbf{x}_i), y_i) := \log(1 + \exp(-y_i f(\mathbf{x}_i)))$$



Loss Functions

0-1 loss

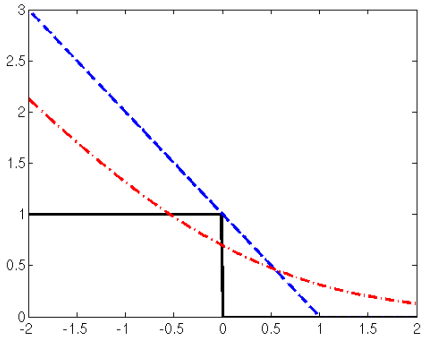
$$\ell(f(\mathbf{x}_i), y_i) := \begin{cases} 0 & y_i = f(\mathbf{x}_i) \\ 1 & y_i \neq f(\mathbf{x}_i) \end{cases}$$

hinge loss

$$\ell(f(\mathbf{x}_i), y_i) := \max\{0, 1 - y_i f(\mathbf{x}_i)\}$$

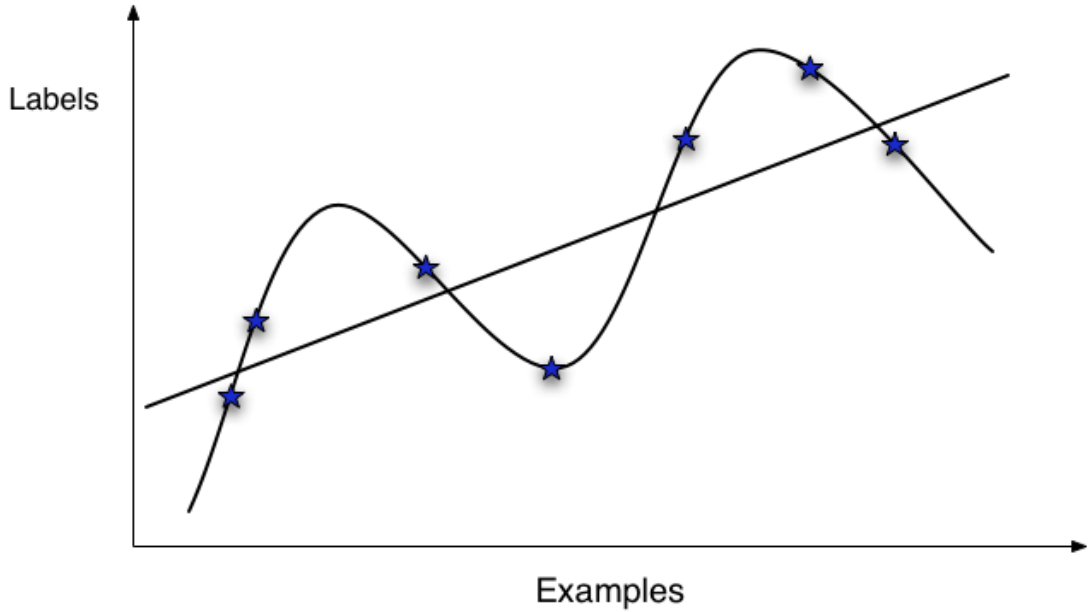
logistic loss

$$\ell(f(\mathbf{x}_i), y_i) := \log(1 + \exp(-y_i f(\mathbf{x}_i)))$$



Hinge loss and Logistic loss are convex.
Logistic loss is differentiable.

Regression



examples $\mathbf{x} \in \mathcal{X}$

labels $y \in \mathbb{R}$

Regression

ε-insensitive loss function

Extend "margin" to regression. Define a "tube" around the line where we can make mistakes.

$$\ell(f(\mathbf{x}_i), y_i) = \begin{cases} 0 & |f(\mathbf{x}_i) - y_i| < \varepsilon \\ |f(\mathbf{x}_i) - y_i| - \varepsilon & \text{otherwise} \end{cases}$$

Squared loss

$$\ell(f(\mathbf{x}_i), y_i) := (y_i - f(\mathbf{x}_i))^2$$

Huber's loss

$$\ell(f(\mathbf{x}_i), y_i) := \begin{cases} \frac{1}{2}(y_i - f(\mathbf{x}_i))^2 & |y_i - f(\mathbf{x}_i)| < \gamma \\ \gamma|y_i - f(\mathbf{x}_i)| - \frac{1}{2}\gamma^2 & (y_i - f(\mathbf{x}_i)) \geqslant \gamma \end{cases}$$

Multiclass

Real problems often have more than 2 classes

Generalize the SVM to multiclass, for $c > 2$.

Three approaches

one-vs-rest

For each class, label all other classes as "negative" (c binary problems).

one-vs-one

Compare all classes pairwise ($\frac{1}{2}c(c-1)$ binary problems).

multiclass loss

Define a new empirical risk term.

Multiclass Loss for SVM

Two class SVM

$$\underset{\mathbf{w},b}{\text{minimize}} \frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{N} \ell(f_{\mathbf{w},b}(\mathbf{x}), y_i),$$

Multiclass SVM

$$\underset{\mathbf{w},b}{\text{minimize}} \ \frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{N} \max_{u \neq y_i} \ \ell\left(f_{\mathbf{w},b}(\mathbf{x}_i, y_i) - f_{\mathbf{w},b}(\mathbf{x}_i, u), y_i\right)$$

Lost?



SVM is dependent on training data

Minimize

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

Subject to

$$y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geqslant 1 - \xi_i$$
$$\xi_i \geqslant 0$$
for all $i = 1, \ldots, N$.

Representer Theorem

$$\mathbf{w} = \sum_{i=1}^{N}\alpha_i\mathbf{x}_i$$

Minimize

$$\frac{1}{2}\sum_{i,j}^{N}\alpha_i\alpha_j\langle \mathbf{x}_i, \mathbf{x}_j\rangle + C\sum_{i=1}^{N}\xi_i$$

Subject to

$$y_i(\langle \sum_{j=1}^{N}\alpha_j\mathbf{x}_j, \mathbf{x}_i\rangle + b) \geqslant 1 - \xi_i$$
$$\xi_i \geqslant 0$$
for all $i = 1, \ldots, N$.

SVM solution only depends on scalar products between examples ($\rightsquigarrow$ kernel trick)

SVM is dependent on training data

Minimize

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

Subject to

$$y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geqslant 1 - \xi_i$$
$$\xi_i \geqslant 0$$
for all $i = 1, \ldots, N$.

Representer Theorem

$$\mathbf{w} = \sum_{i=1}^{N}\alpha_i\mathbf{x}_i$$

Minimize

$$\frac{1}{2}\sum_{i,j}^{N}\alpha_i\alpha_j\langle \mathbf{x}_i, \mathbf{x}_j\rangle + C\sum_{i=1}^{N}\xi_i$$

Subject to

$$y_i(\langle \sum_{j=1}^{N}\alpha_j\mathbf{x}_j, \mathbf{x}_i\rangle + b) \geqslant 1 - \xi_i$$
$$\xi_i \geqslant 0$$
for all $i = 1, \ldots, N$.

SVM solution only depends on scalar products between examples ($\rightsquigarrow$ kernel trick)

SVM is dependent on training data

Minimize

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

Subject to

$$y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geqslant 1 - \xi_i$$
$$\xi_i \geqslant 0$$
for all $i = 1, \ldots, N$.

Representer Theorem

$$\mathbf{w} = \sum_{i=1}^{N}\alpha_i\mathbf{x}_i$$

Minimize

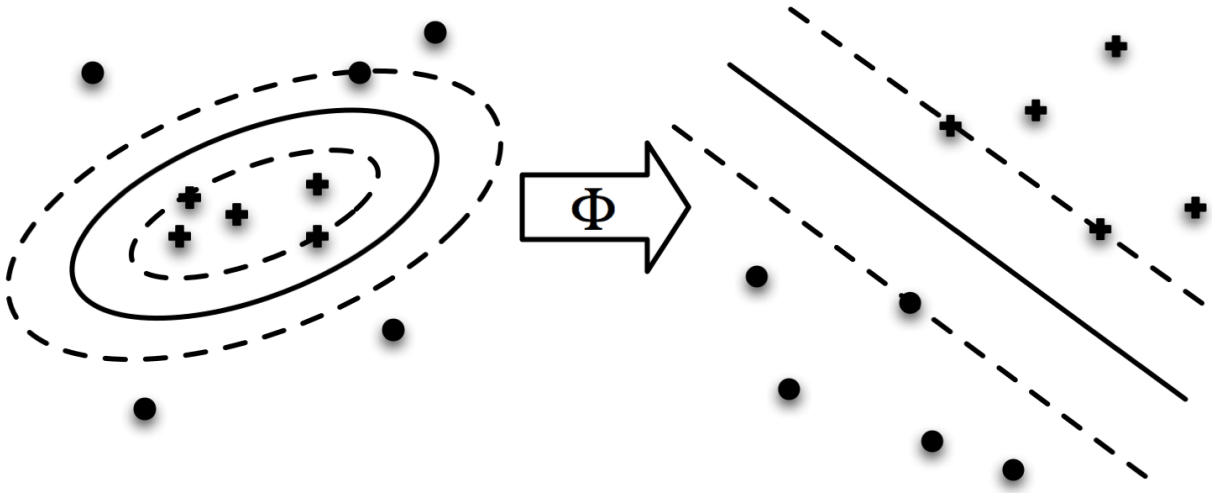
$$\frac{1}{2}\sum_{i,j}^{N}\alpha_i\alpha_j\langle \mathbf{x}_i, \mathbf{x}_j\rangle + C\sum_{i=1}^{N}\xi_i$$

Subject to

$$y_i(\langle \sum_{j=1}^{N}\alpha_j\mathbf{x}_j, \mathbf{x}_i\rangle + b) \geqslant 1 - \xi_i$$
$$\xi_i \geqslant 0$$
for all $i = 1, \ldots, N$.

SVM solution only depends on scalar products between examples ($\rightsquigarrow$ kernel trick)

Kernels



Example kernels

Linear kernel

$$k(\mathbf{x}, \mathbf{z}) := \langle \mathbf{x}, \mathbf{z} \rangle$$

Explicit Feature Map

$$k\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}\right) = \begin{bmatrix} x_1^2 \\ \sqrt{2}\, x_1 x_2 \\ x_2^2 \end{bmatrix}^\top \begin{bmatrix} z_1^2 \\ \sqrt{2}\, z_1 z_2 \\ z_2^2 \end{bmatrix}$$

Polynomial kernel

$$k(\mathbf{x}, \mathbf{z}) := \left(\frac{1}{\sigma}\langle \mathbf{x}, \mathbf{z} \rangle + c\right)^2$$

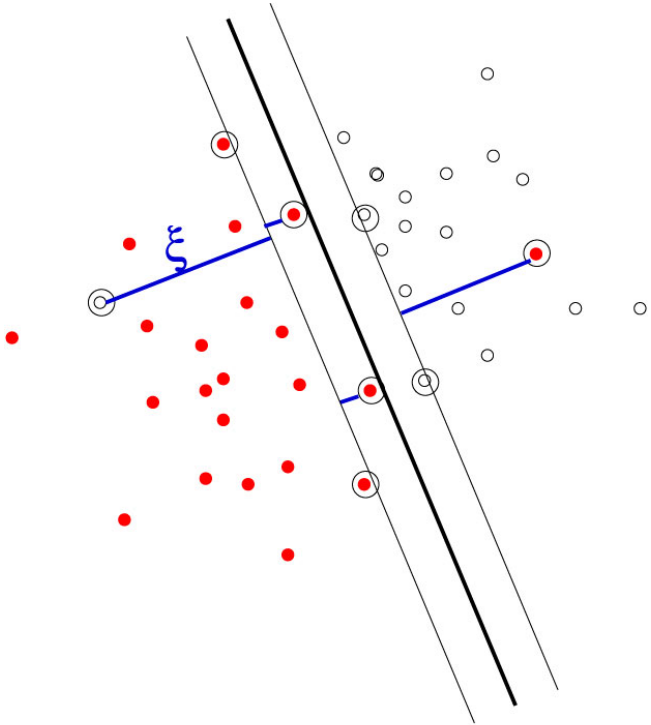
Gaussian kernel

$$k(\mathbf{x}, \mathbf{z}) := \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{x} - \mathbf{z}\|^2\right)$$

Toy Examples

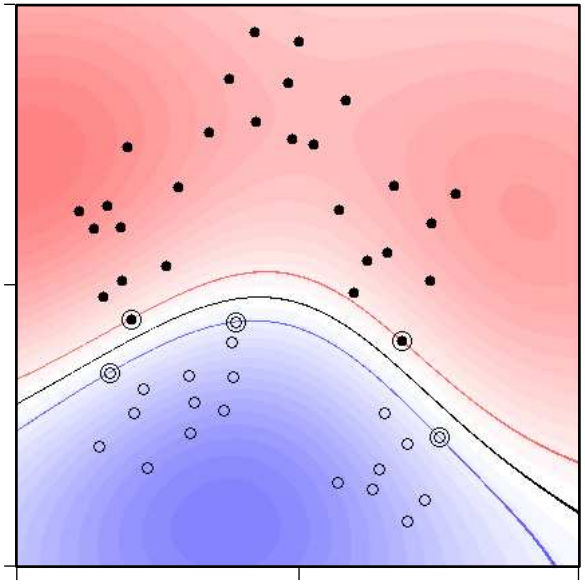
Linear kernel

$k(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle$



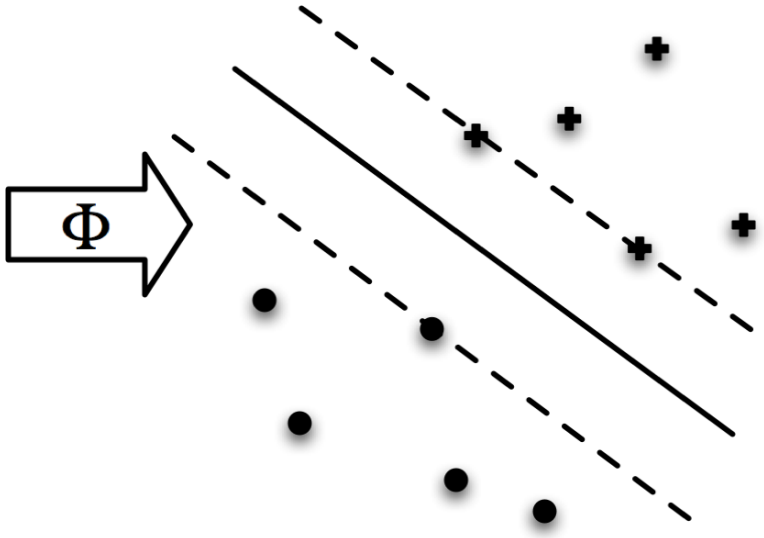
RBF kernel

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



String mapping

GCCCGGTTAAGGT	1
AACTTGGGTGTGCC	-1
ATATTATAGCCG	1
GCGCAAGGGGGCCTT	-1
TTGCCGAAAAGGCTAA	1



Kernel $\approx$ Similarity Measure

Distance:

$$\|\Phi(\mathbf{x}) - \Phi(\mathbf{y})\| = \|\Phi(\mathbf{x})\|^2 - 2\langle\Phi(\mathbf{x}), \Phi(\mathbf{y})\rangle + \|\Phi(\mathbf{y})\|$$

Scalar product:

- If $\|\Phi(\mathbf{x})\|^2 = \|\Phi(\mathbf{y})\|^2 = 1$, then

  scalar product = $2-$distance

- Angle between vectors, i.e.

$$\frac{\langle\Phi(\mathbf{x}), \Phi(\mathbf{y})\rangle}{\|\Phi(\mathbf{x})\| \, \|\Phi(\mathbf{y})\|} = \cos(\Phi(\mathbf{x}), \Phi(\mathbf{y}))$$

Can we use just any similarity measure?

Positive Semidefiniteness

- $$\sum_{i,j} c_i c_j k(\mathbf{x}_i, \mathbf{x}_j) \geqslant 0.$$
- Kernels are a similarity measure
- Dot product in feature space
- The matrix K defined by $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ is called the kernel matrix or the Gram matrix. Hence the above equation can be written as:
$$\mathbf{c}^\top K \mathbf{c} \geqslant 0. \text{ or } K \succeq 0.$$
- k is a positive definite kernel function if for any set of objects $\{\mathbf{x}_i\}_{i=1}^N$, the resulting kernel matrix K is positive definite.

Kernels

Feature Mapping

We define a function from the input space $\mathcal{X}$ to the feature space $\mathcal{H}$ where we perform the estimation.

Formally

$$\begin{aligned} \mathbf{\Phi} &: \mathcal{X} \longrightarrow \mathcal{H} \\ \mathbf{x} &\mapsto \mathbf{x} := \mathbf{\Phi}(\mathbf{x}) \end{aligned}$$

where $\mathcal{H}$ is a dot product space (*feature space*)

Kernel Trick

$$k(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle$$

Putting it together

Minimize

$$\frac{1}{2}\sum_{i,j}^{N}\alpha_i\alpha_j\langle\mathbf{x}_i,\mathbf{x}_j\rangle$$

Subject to

$$y_i(\langle\sum_{j=1}^{N}\alpha_j\mathbf{x}_j,\mathbf{x}_i\rangle+b)\geqslant 1$$

for all $i=1,\ldots,N$.

Minimize

$$\frac{1}{2}\sum_{i,j}^{N}\alpha_i\alpha_j k(\mathbf{x}_i,\mathbf{x}_j)$$

Subject to

$$y_i(\sum_{j=1}^{N}\alpha_j k(\mathbf{x}_j,\mathbf{x}_i)+b)\geqslant 1$$

for all $i=1,\ldots,N$.

Recall the linearity of scalar products:

$$\langle\sum_{j=1}^{N}\alpha_j\mathbf{x}_j,\mathbf{x}_i\rangle=\sum_{j=1}^{N}\alpha_j\langle\mathbf{x}_j,\mathbf{x}_i\rangle=\sum_{j=1}^{N}\alpha_j k(\mathbf{x}_j,\mathbf{x}_i)$$

The kernel trick

Basic Notion

Replace dot products by kernels.

Similarity based algorithms

There are many vector space based algorithms where data is only accessed via comparisons between pairs of points.

SVM

SVMs are a case where this is true, but only because of the representer theorem. The representer theorem allows us to express the weight vector w as a linear combination of training points.

Which space are you in?



How to construct a kernel

At least two ways to get to a kernel:

- Construct Φ and think about efficient ways to compute $\langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle$
- Construct similarity measure, show positiveness and think about what it means

What can you do if kernel is not positive definite?

- Optimization problem is not convex!
- Add constant to diagonal (cheap)
- Exponentiate kernel matrix (all eigenvalues become positive)
- SVM-pairwise use similarity as features

Closure properties

Addition and Multiplication

- If k_1, k_2 are kernels, then
- $k_1 + k_2$ is a kernel.
- $k_1 * k_2$ is a kernel.
- $\lambda * k_1$ is a kernel, where $\lambda > 0$.

Pointwise limit

- If $k_1, k_2, \ldots$ are kernels, and $k(x, x') := \lim_{n\to\infty} k_n(x, x')$ exists for all x,x',
- then k is a kernel.

Closure properties

Zero extension

- $S \subset X$, k is kernel on $S \times S$.
- If k is zero-extended to X, i.e. k(x,y)= 0 if either $x \notin S$ or $y \notin S$,
- then k is a kernel.

Closure properties

Tensor product and direct sum

- If k_1, k_2 are kernels defined on $\mathcal{X}_1 \times \mathcal{X}_1$ and $\mathcal{X}_2 \times \mathcal{X}_2$,
- then their tensor product,

$$(k_1 \otimes k_2)(x_1, x_2, x_1', x_2') = k_1(x_1, x_1') * k_2(x_2, x_2')$$

  is a kernel on $(\mathcal{X}_1 \times \mathcal{X}_2) \times (\mathcal{X}_1 \times \mathcal{X}_2)$
- and their direct sum

$$(k_1 \oplus k_2)(x_1, x_2, x_1', x_2') = k_1(x_1, x_1') + k_2(x_2, x_2')$$

  is a kernel on $(\mathcal{X} \times \mathcal{X}) \times (\mathcal{X} \times \mathcal{X})$.
- Here $x_1, x_1' \in \mathcal{X}_1$ and $x_2, x_2' \in \mathcal{X}_2$.

Forced positive definiteness

Problem

- Given: a similarity measure that works well in practice
- Want to use it with kernel methods
- But it is not positive definite

Empirical map

- Each object is represented by a vector
- Vector contains similarity scores to other objects

Negative eigenvalue removal

- Eigendecomposition of the similarity matrix
- Removal of negative eigenvalues

Forced positive definiteness

Cutting off negative eigenvalues

- Compute kernelmatrix K
- Determine eigenvalues λ_i and eigenvectors v_i of K
- Cut off negative eigenvalues by calculating

$$K_I = \sum_{i:\lambda_i>0} \lambda_i v_i v_i'$$

- Then K_I is a positive definite approximation to K

Summary I

Basic Notion

Replace dot products by kernels.

Similarity based algorithms

There are many vector space based algorithms where data is only accessed via comparisons between pairs of points.

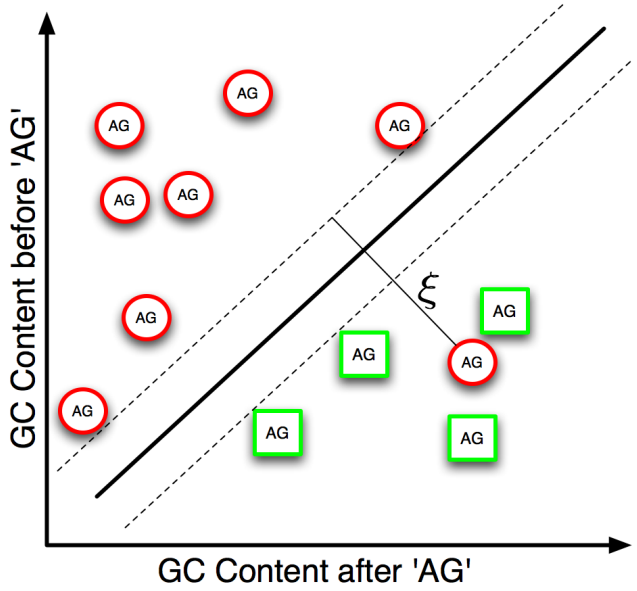
SVM

SVMs are a case where this is true, but only because of the representer theorem. The representer theorem allows us to express the weight vector w as a linear combination of training points.

Designing Kernels

For a particular application, it is often useful to combine previously known kernels.

Summary II



Minimize

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

Subject to

$$y_i(\langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle + b) \geqslant 1 - \xi_i$$
$$\xi_i \geqslant 0$$
for all $i = 1, \ldots, N$.

- Loss view $\Leftrightarrow$ Geometric view
- Next week: How to solve the problem numerically.