Advanced Methods for Sequence Analysis

Cheng Soon Ong[1,2]

[1] Friedrich Miescher Laboratory, Tübingen
[2] Max Planck Institute for Biological Cybernetics, Tübingen

Vorlesung WS 2006/2007
Eberhard Karls Universität Tübingen

25 October 2006

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

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

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

Objective function (1) Maximize the margin.

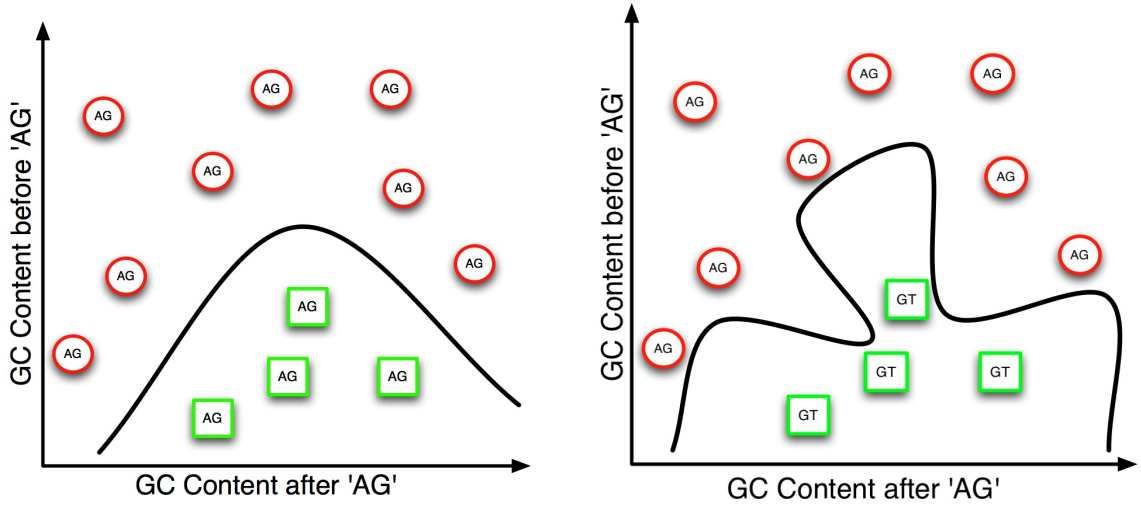
Constraints (2) Correctly classify the training data.

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

Lecture 2: Outline

- Support Vector Machines
- Computing the Optimal

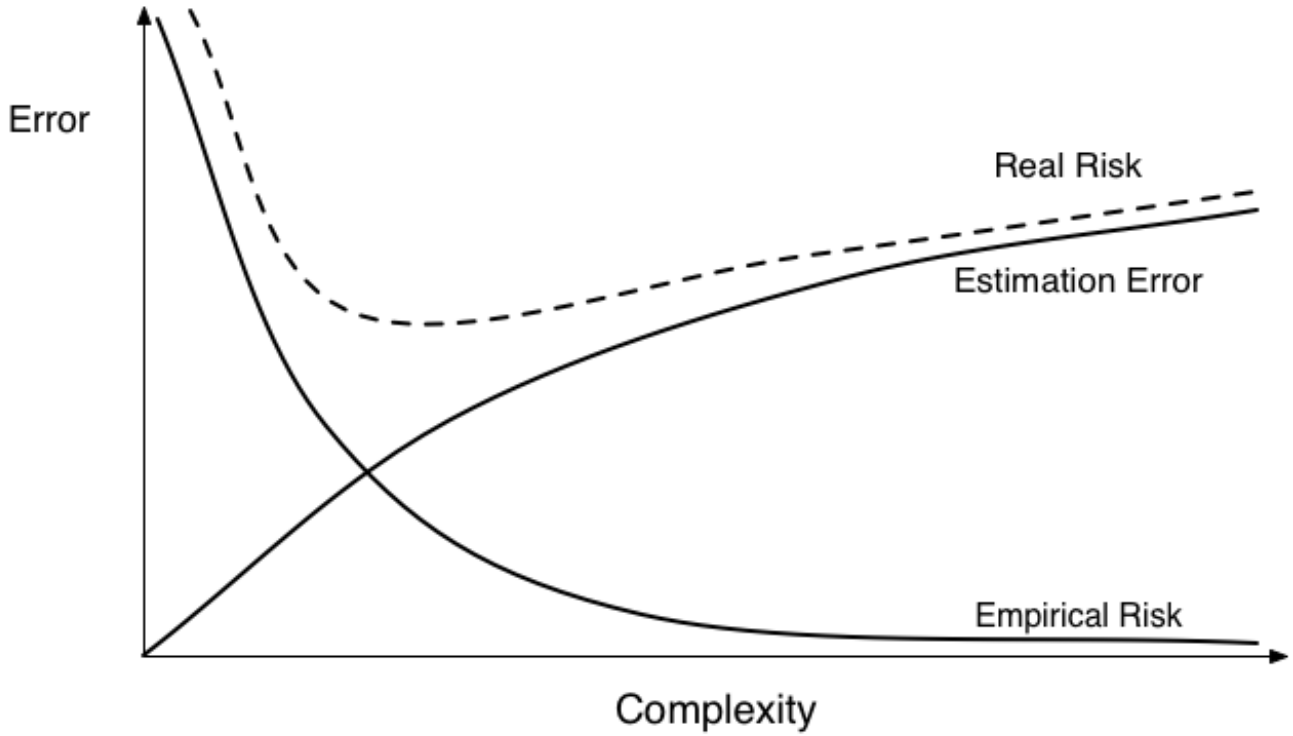
Simple vs Complex Functions



For a given set of training data, there are many possible functions which can explain it. However, some functions are “simple” and others are “complex”. We want to estimate a functional dependence from a set of examples.

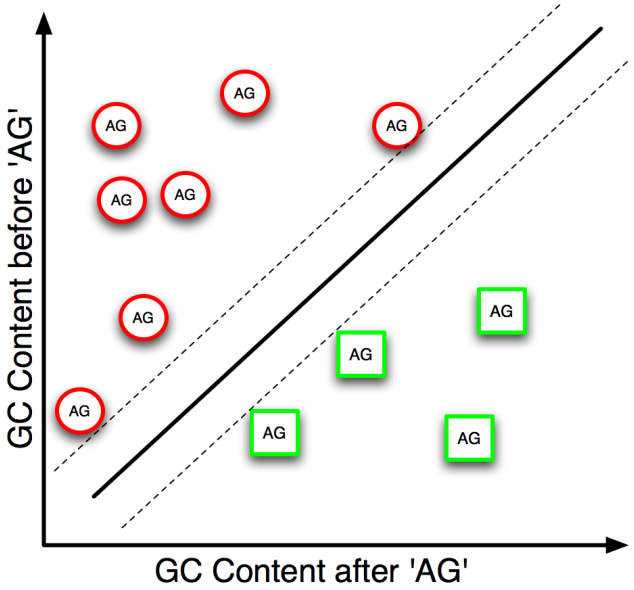
Which function is preferable?

Structural Risk Minimization



The complexity or capacity is a property of the function class, and not any individual function f.

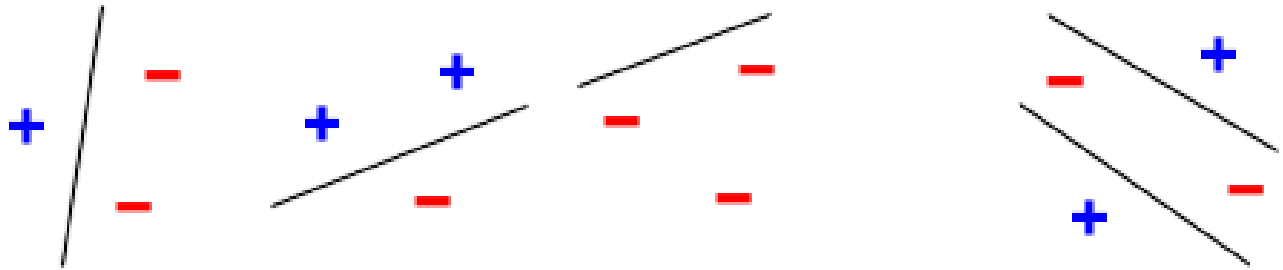
Complexity of Hyperplanes



What is the complexity of a hyperplane classifier?

VC Dimension

A model class shatters a set of data points if it can correctly classify any possible labeling.



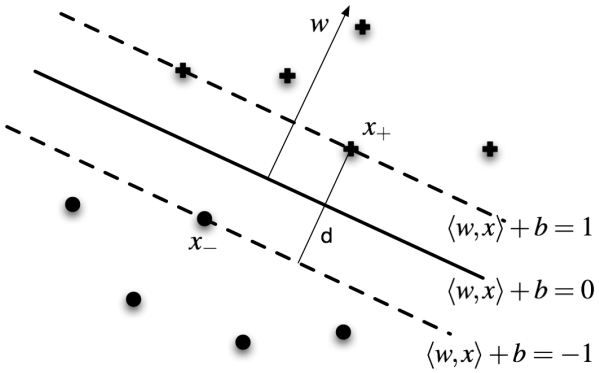
Lines shatter any 3 points in $\mathbb{R}^2$, but not 4 points.

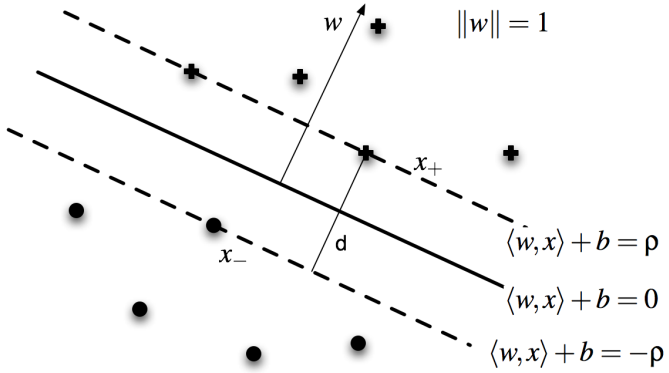
VC dimension [Vapnik, 1995]

The VC dimension of a model class is the maximum h such that some data point set of size h can be shattered by the model. (e.g. VC dimension of $\mathbb{R}^2$ is 3.)

Complex model classes have large VC dimension.

Margin Maximization



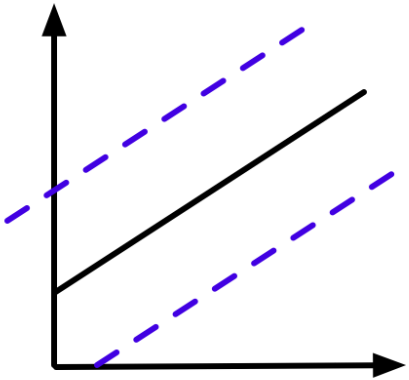


There are two equivalent formulations of the problem.

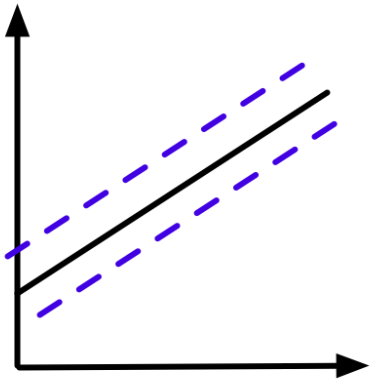
Larger Margin $\Rightarrow$ Less Complex

Large Margin $\Rightarrow$ Small VC dimension

Hyperplane classifiers with large margin have small VC dimension [Vapnik, 1995].



VC dim. small



VC dim. large

Maximum Margin $\Rightarrow$ Minimum Complexity

Minimize complexity by maximizing margin (irrespective of the dimension of the space).

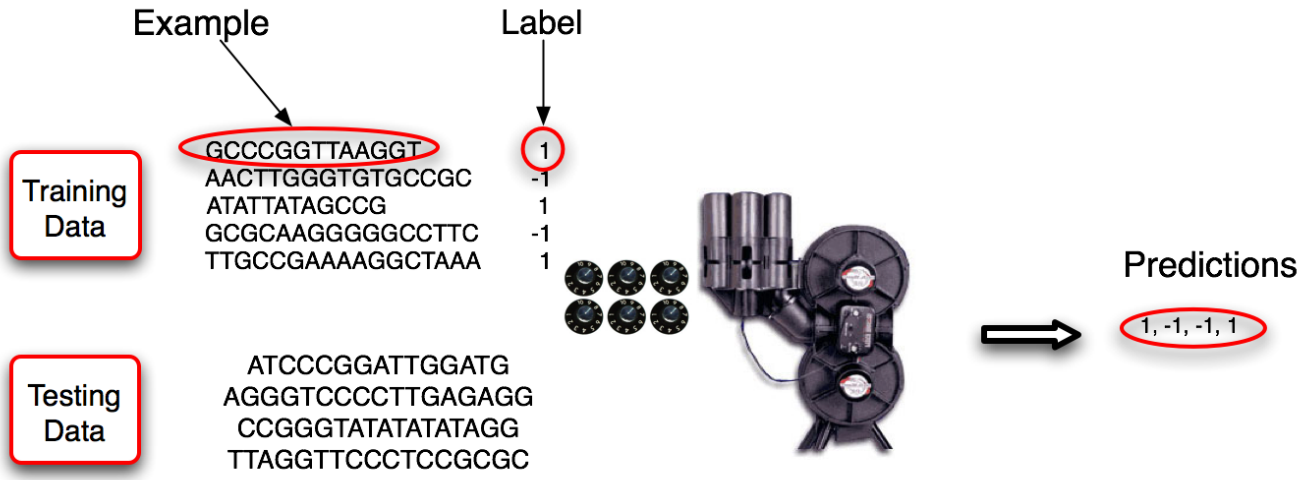
Why maximize the margin?

- Intuitively, it feels the safest.
- For a small error in the separating hyperplane, we do not suffer too many mistakes.
- Empirically, it works well.
- VC theory indicates that it is the right thing to do.
- There is one global maximum, i.e. the problem is convex.

Why maximize the margin?

- Intuitively, it feels the safest.
- For a small error in the separating hyperplane, we do not suffer too many mistakes.
- Empirically, it works well.
- VC theory indicates that it is the right thing to do.
- There is one global maximum, i.e. the problem is convex.

Review: Generalization Error



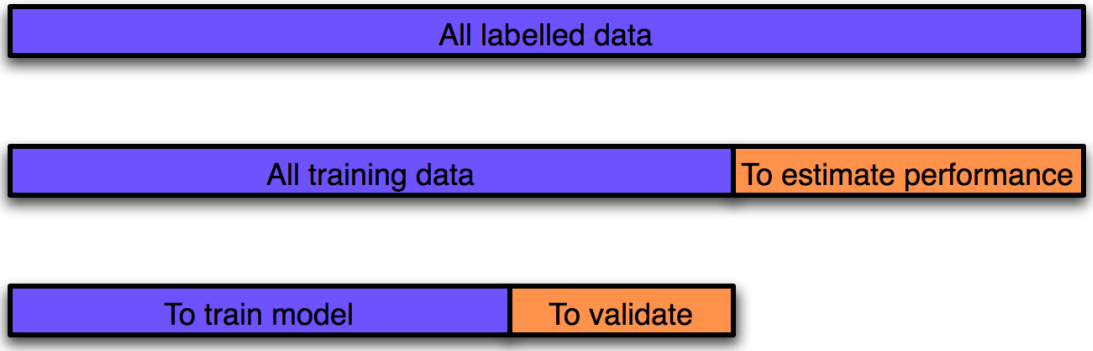
Estimator $f : \mathcal{X} \to \mathcal{Y}$, the function we want to learn.

Loss $\ell(f(\mathbf{x}_i), y_i)$, the error for a particular example.

Risk $\mathbf{R}(f)$, The expected loss for all data, including unseen.

Empirical Risk $\mathbf{R}_{emp}(f)$. The average loss we observe.

Measuring performance



Do not train on the test set!

- Use subset of data for training
- From subset, further split to select model.

Model Selection = Find best parameters

- SVM parameter C.
- Kernel parameters.

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?

Loss Functions

0-1 loss

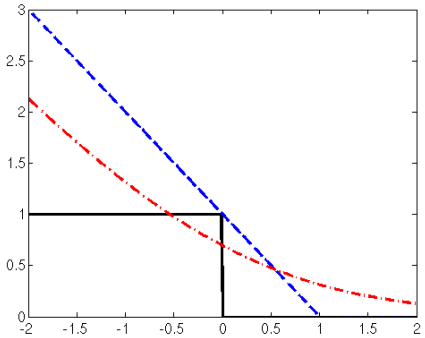
$$\ell(f(\mathbf{x}_i), y_i) := \begin{cases} 0 & y_i = f(\mathbf{x}_i) \\ 1 & y_i \neq f(\mathbf{x}_i) \end{cases}$$

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

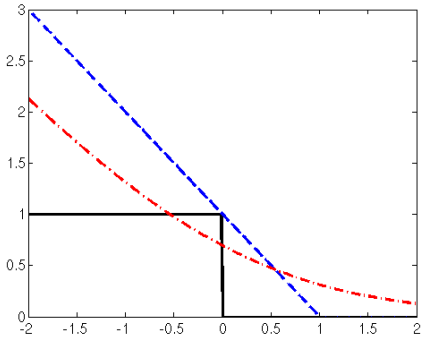
$$\ell(f(\mathbf{x}_i), y_i) := \begin{cases} 0 & y_i = f(\mathbf{x}_i) \\ 1 & y_i \neq f(\mathbf{x}_i) \end{cases}$$

hinge loss

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

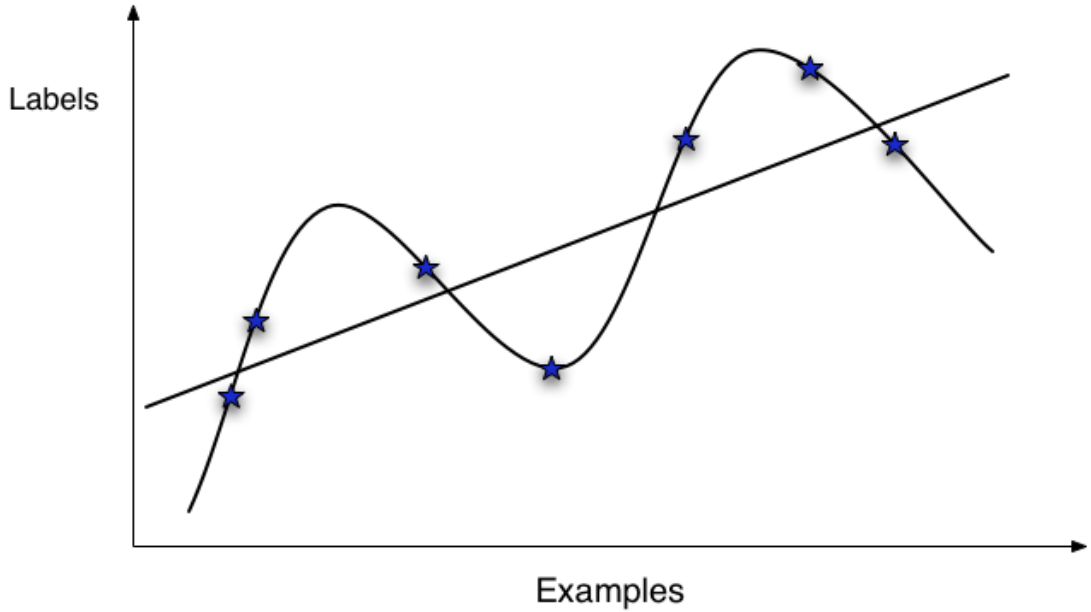
logistic loss

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



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

Regression



examples $x \in \mathcal{X}$

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

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

See e.g. Smola and Schölkopf [2001] for other loss functions and more details.

Multiclass

Real problems often have more than 2 classes

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

Three approaches [Schölkopf and Smola, 2002]

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

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

So far...

- Support Vector Machines
  - VC theory shows that maximizing the margin controls capacity.
  - Loss function view for binary classification, regression and multiclass
- Computing the Optimal
  - Two equivalent formulations of the SVM, the primal and the dual.
  - Many possible tools for unconstrained and constrained optimization.

SVM: How to find solution?

Find a function

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

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

$$\underset{\mathbf{w},b}{\text{minimize}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} \xi_i \tag{3}$$

$$\text{subject to} \quad y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geqslant 1 - \xi_i \text{ and } \xi_i \geqslant 0 \tag{4}$$

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

Approach 1

Optimize the regularized loss directly.

Approach 2

Solve the dual problem.

Approach 2 is more commonly used.

Minimize Loss Directly

Gradient Descent

Minimize the loss by searching along the line of steepest gradient. This requires that the loss function is differentiable.

Newton Method

Use second order information to improve the search. Usually requires fewer iterations to converge to the solution. This requires that the loss function is twice differentiable.

Conjugate Gradient

Like gradient descent, but do not search along directions which have been searched before. Usually converges significantly faster than simple gradient descent.

Constrained optimization

Convex optimization

There is a large body of work and software which efficiently solves constrained optimization problems.

Interior point methods

The current method of choice is known as primal dual interior point methods. The method finds the minimum objective function while keeping in the "interior" of the constraints.

More details

The method solves a series of linear systems of equations, called the "Karush-Kuhn-Tucker" conditions. These conditions also tell us how close we are to optimality.

Recipe to find Dual

1. Express optimization problem in standard form.
2. Form the Lagrangian
3. Differentiate the Lagrangian with respect to the primal variables
4. Find stationary points by setting the gradients above to zero
5. Substitute new equations into Lagrangian
6. Don't forget the constraints on the Lagrange multipliers

Lagrangian Duality

For $f_i : \mathbb{R}^N \to \mathbb{R}$ and $g_j : \mathbb{R}^N \to \mathbb{R}$:

$$\begin{array}{ll} \min_{x \in \mathbb{R}^N} & f_0(x) \\ \text{subject to} & f_i(x) \leqslant 0 \text{ for } i = 1, \ldots, m \,, \\ & g_j(x) = 0 \text{ for } j = 1, \ldots, p \end{array}$$

The Lagrangian is defined as

$$\mathcal{L}(x, \lambda, \gamma) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{j=1}^{p} \gamma_j g_j(x),$$

where λ and γ are real valued vectors called the Lagrange multipliers with $\lambda_i \geqslant 0$ and γ_i free.

The vectors λ and γ are also known as dual variables.

Hard Margin SVM

$$\begin{array}{ll} \text{minimize}_{w,b} & \frac{1}{2}\|w\|^2 \\ \text{subject to} & y_i(\langle w, x_i\rangle + b) \geqslant 1 \text{ for all } i = 1, \ldots, N. \end{array}$$

We get the Lagrangian

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \alpha_i (y_i(\langle w, x_i\rangle + b) - 1),$$

where $\alpha_1, \ldots, \alpha_m$ are Lagrange multipliers. At the optimal, the derivatives of $\mathcal{L}$ with respect to the primal variables must vanish:

$$\frac{\partial \mathcal{L}(w, b, \alpha)}{\partial w} = w - \sum_{i=1}^{N} y_i \alpha_i x_i = 0.$$

$$\frac{\partial \mathcal{L}(w, b, \alpha)}{\partial b} = \sum_{i=1}^{N} y_i \alpha_i = 0.$$

Dual Problem

maximize

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle$$

subject to the constraints:

(1) $\sum_{i=1}^{N} \alpha_i y_i = 0$

(2) $\alpha_i \geqslant 0$ for $i = 1, 2, \ldots, N$

Observations

- Solution is a function of the training data (This is a simple version of the Representer Theorem)

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i$$

- The points where $\alpha_i > 0$ are called support vectors).
- Dual optimization problem expressed as dot products between training data

Optimization

We have two versions of the same problem

Primal

$$\text{minimize}_{w,b} \frac{1}{2}\|w\|^2 + \sum_{i=1}^{N} \ell(x_i, w^\top x_i + b, y_i).$$

For a convex differentiable loss function, we can solve this directly using gradient methods or Newton's method.

Dual

$$\begin{aligned} \text{maximize}_\alpha \quad & \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \\ \text{subject to} \quad & \sum_{i=1}^{N} \alpha_i y_i = 0 \\ & \alpha_i \geqslant 0 \text{ for } i = 1, 2, \ldots, N. \end{aligned}$$

We can use primal dual interior point methods to solve the convex optimization problem.

Convex Optimization

- SVMs are a special case of Quadratic Programs (QPs)
  - Maximizing the margin is convex.
  - Loss function is convex.
- QPs can be efficiently solved via constrained optimization.
  For $f_i : \mathbb{R}^N \to \mathbb{R}$ and $g_j : \mathbb{R}^N \to \mathbb{R}$:

$$\begin{array}{ll} \min_{x \in \mathbb{R}^N} & f_0(x) \\ \text{subject to} & f_i(x) \leqslant 0 \text{ for } i = 1, \ldots, m \ , \\ & g_j(x) = 0 \text{ for } j = 1, \ldots, p \end{array}$$

- There exists many open source and commercial packages for solving convex optimization problems.

QPs for SVMs

- General Purpose QP solver (e.g. CPLEX [CPL, 1994])
  - Does not exploit problem structure.
- Chunking Methods [Osuna et al., 1997]
  - Select subsets, solve QPs, join the sets, ...
- SVM-Light [Joachims, 1999]
  - Select n variables, solve QP, ...
- SMO Algorithm [Platt, 1999]
  - Select two variables, solve QP analytically, ...
- ...

Shogun toolbox Sonnenburg et al. [2006]

- SVM-Light type QP optimization
- Many string kernels implementations

Dual Problem

$$\text{maximise}_\alpha \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle$$

$$\text{subject to } \sum_{i=1}^{N} \alpha_i y_i = 0$$

$$\alpha_i \geqslant 0 \text{ for } i = 1, 2, \ldots, N$$

Dual Problem + Kernel

$$\text{maximise}_\alpha \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle$$

$$\text{subject to } \sum_{i=1}^{N} \alpha_i y_i = 0$$

$$\alpha_i \geqslant 0 \text{ for } i = 1, 2, \ldots, N$$

+ kernel trick $\rightsquigarrow$

$$\text{maximise}_\alpha \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j k(x_i, x_j)$$

$$\text{subject to } \sum_{i=1}^{N} \alpha_i y_i = 0$$

$$\alpha_i \geqslant 0 \text{ for } i = 1, 2, \ldots, N$$

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

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

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

Kernel Trick

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

Lecture 2: Summary

- Support Vector Machines
  - VC theory shows that maximizing the margin controls capacity.
  - Loss function view for binary classification, regression and multiclass
- Computing the Optimal
  - Two equivalent formulations of the SVM, the primal and the dual.
  - Many possible tools for unconstrained and constrained optimization.

References

Using the CPLEX Callable Library. CPLEX Optimization Incorporated, Incline Village, Nevada, 1994.

T. Joachims. Making large–scale SVM learning practical. In B. Schölkopf, C.J.C. Burges, and A.J. Smola, editors, *Advances in Kernel Methods — Support Vector Learning*, pages 169–184, Cambridge, MA, 1999. MIT Press.

E. Osuna, R. Freund, and F. Girosi. An improved training algorithm for support vector machines. In J. Principe, L. Gile, N. Morgan, and E. Wilson, editors, *Neural Networks for Signal Processing VII — Proceedings of the 1997 IEEE Workshop*, pages 276–285, New York, 1997. IEEE.

J. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C.J.C. Burges, and A.J. Smola, editors, *Advances in Kernel Methods — Support Vector Learning*, pages 185–208, Cambridge, MA, 1999. MIT Press.

B. Schölkopf and A.J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

A.J. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 2001.

Sören Sonnenburg, Gunnar Rätsch, Christin Schäfer, and Bernhard Schölkopf. Large Scale Multiple Kernel Learning. *Journal of Machine Learning Research*, 7:1531–1565, July 2006.

V.N. Vapnik. *The nature of statistical learning theory*. Springer Verlag, New York, 1995.