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

6 November 2007

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

Recall: How to form kernels

Addition and Multiplication

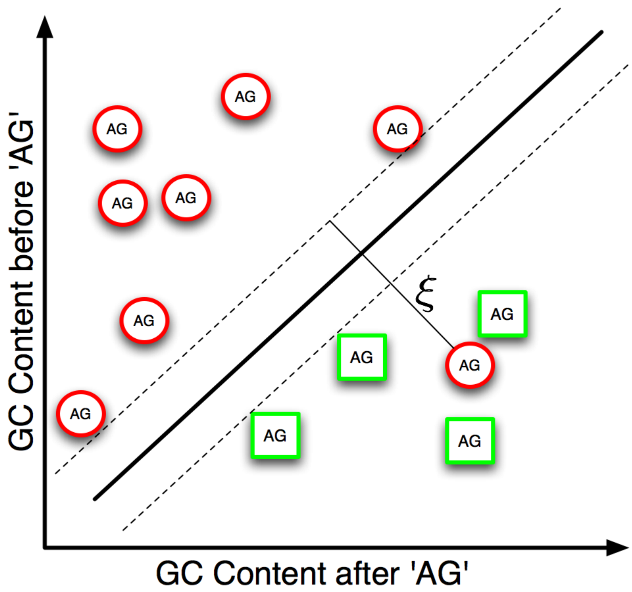
- If k_1, k_2 are kernels, then
- $k_1 + k_2$ is a kernel.
- $k_1 * k_2$ is a kernel.
- $\lambda * k_1$ is a kernel, where $\lambda > 0$.

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.

Zero extension

Recall the SVM



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
- Today: How to solve the problem numerically.

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

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

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.

Gradient Descent

Gradient

For a multivariate function $f : \mathbb{R}^n \to \mathbb{R}$, define the gradient of f to be

$$\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \ldots, \frac{\partial f(x)}{\partial x_n}, \right).$$

Algorithm

For an intial value x_0 and precision ε,

$k = 0$

while $(\|\nabla f(x_k)\| \geqslant \varepsilon)$

  Compute $g = \nabla f(x_k)$

  Perform line search on $f(x_k - \gamma g)$ for optimal γ

  $x_{k+1} = x_k - \gamma g$

  $k = k + 1$

Newton Method (1)

Hessian

For a multivariate function $f : \mathbb{R}^n \to \mathbb{R}$,

$$\nabla^2 f(x)_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}, \qquad i, j = 1, \ldots, n$$

Motivation

The second order Taylor approximation $\hat{f}$ of f at x is

$$\hat{f}(x + v) = f(x) + \nabla f(x)^\top v + \frac{1}{2} v^\top \nabla^2 f(x) v,$$

which is a convex quadratic function of v, with minimizer

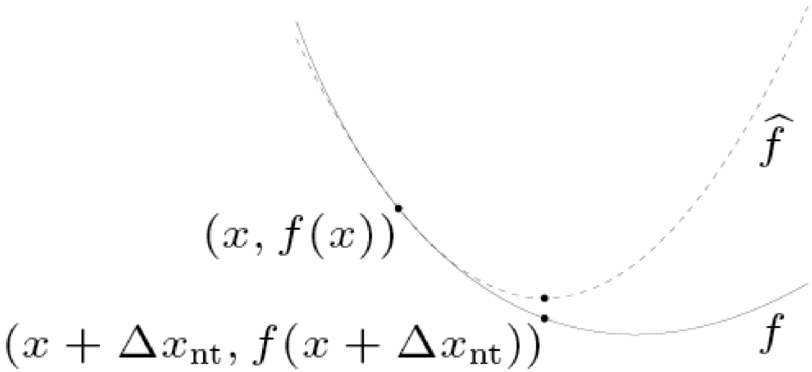
$$v^* = -\nabla^2 f(x)^{-1} \nabla f(x).$$

Newton step

The vector v^* above is called the Newton step for f at x.

Newton Method (2)

Motivation



Algorithm

For an intial value x_0 and precision ε,

$k = 0$

while $(\|\nabla f(x_k)\| \geqslant \varepsilon)$

$$x_{k+1} = x_k - \nabla^2 f(x)^{-1} \nabla f(x)$$

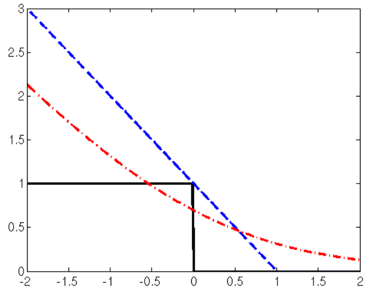
$$k = k + 1$$

Non-smooth functions

Motivation

Non differentiable functions f have to be treated carefully. For example the hinge loss

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



Piecewise differentiable

For piecewise differentiable functions, we can treat each piece individually, and then we can apply the above methods.

Convex Optimization

MAX-PLANCK-GESELLSCHAFT

- 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.

Constrained Optimization

$$\begin{array}{rl} \min_x & f_0(x) \\ \text{subject to} & f_i(x) \leqslant 0 \text{ for all } i \\ & g_j(x) = 0 \text{ for all } j \end{array}$$

- $x \in \mathbb{R}^n$ is the optimization variable
- $f_0 : \mathbb{R}^n \to \mathbb{R}$ is the objective or cost function
- $f_i : \mathbb{R}^n \to \mathbb{R}$ are inequality constraint functions
- $g_j : \mathbb{R}^n \to \mathbb{R}$ are equality constraint functions

Convex Optimization

Constrained Optimization (generally hard)

$$\begin{aligned} \min_x \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \leqslant 0 \text{ for all } i \\ & g_j(x) = 0 \text{ for all } j \end{aligned}$$

Convex Optimization (generally easy)

$$\begin{aligned} \min_x \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \leqslant 0 \text{ for all } i \\ & a_j^\top x = b_j \text{ for all } j \end{aligned}$$

$f_0, f_1, \ldots, f_m$ are convex, and the equality constraints are affine [Boyd and Vandenberghe, 2004].

Convex Function

$f : \mathbb{R}^n \rightarrow \mathbb{R}$ is convex if the domain of f is a convex set and

$$f(\theta x_1 + (1-\theta)x_2) \leqslant \theta f(x_1) + (1-\theta) f(x_2)$$

for all x_1, x_2 in the domain of f, $0 \leqslant \theta \leqslant 1$.

- affine: $ax + b$ on $\mathbb{R}$, for any $a, b \in \mathbb{R}$.
- affine: $a^\top x + b$ on $\mathbb{R}^n$, for any $a \in \mathbb{R}^n, b \in \mathbb{R}$.
- exponential: $\exp^{ax}$ for any $a \in \mathbb{R}$.
- powers: x^a on $\mathbb{R}_{++}$, for $a \geqslant 1$ or $a \leqslant 0$.
- powers of absolute value: $|x|^a$ on $\mathbb{R}$, for $a \geqslant 1$.
- negative entropy: $x \log x$ on $\mathbb{R}_{++}$.
- norms: $\|x\|_p = (\sum_{i=1}^n |x_i|^p)^{\frac{1}{p}}$ for $p \geqslant 1$.

How to check (1)

Restrict to line

$f : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if the function $g : \mathbb{R} \to \mathbb{R}$,

$$g(t) = f(x + tv), \quad \mathbf{dom}\ g = \{t | x + tv \in \mathbf{dom}\ f\}$$

is convex (in t) for any $x \in \mathbf{dom}\ f, v \in \mathbb{R}^n$.

First-order condition

The first order approximation of f is a global underestimator.

$$\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \ldots, \frac{\partial f(x)}{\partial x_n}, \right)$$

f with convex domain is convex if and only if

$$f(x_2) \geqslant f(x_1) + \nabla f(x_1)^\top (x_2 - x_1)$$

for all $x_1, x_2 \in \mathbf{dom}\ f$.

How to check (2)

Second-order condition

The hessian is positive semi-definite.

$$\nabla^2 f(x)_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}, \quad i, j = 1, \ldots, n$$

f is convex if and only if

$$\nabla^2 f(x) \succeq 0 \text{ for all } x \in \mathbf{dom}\ f.$$

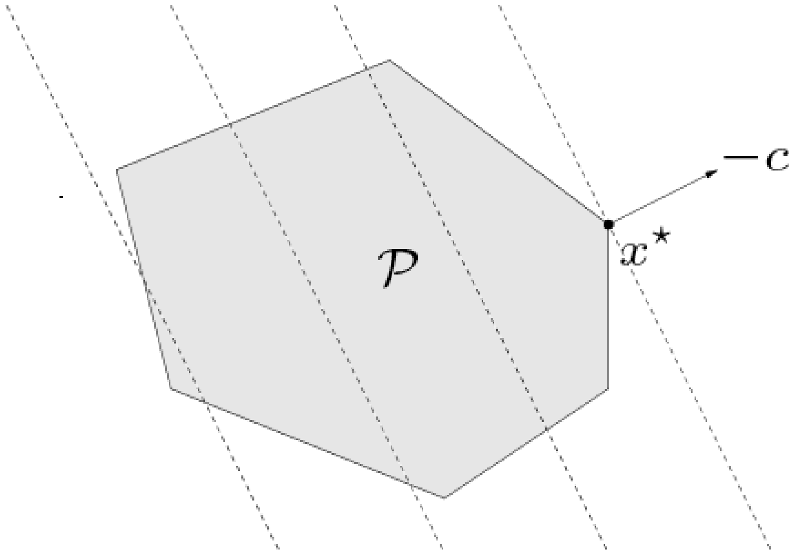
How to check (3)

Show that f is obtained from simple convex functions by operations that preserve convexity

- nonnegative weighted sum
- composition with affine function
- pointwise maximum and supremum
- composition
- minimization
- perspective

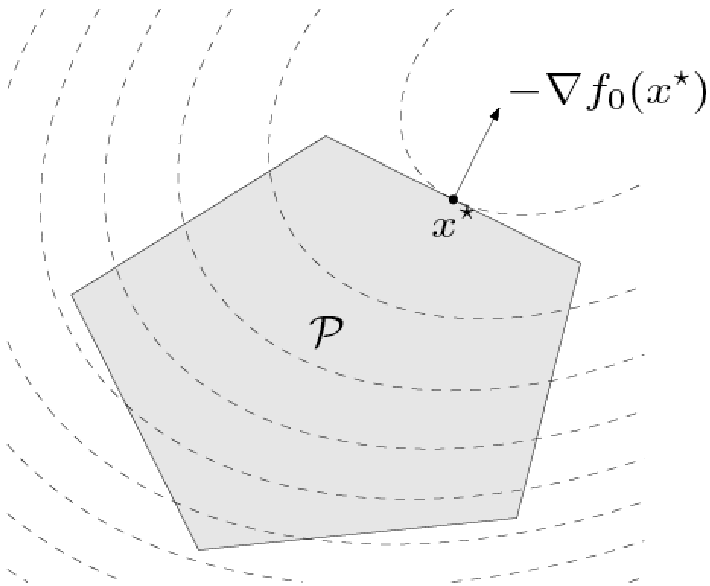
Linear Program

$$\begin{aligned} \min_x \;& c^\top x + d \\ \text{subject to } \;& Gx \leqslant h \\ & Ax = b \end{aligned}$$



Quadratic Program

$$\begin{array}{rl} \min_x & \frac{1}{2}x^\top Px + q^\top x + r \\ \text{subject to} & Gx \leqslant h \\ & Ax = b \end{array}$$



Some other programs

Quadratically constrained quadratic program

$$
\begin{array}{rl}
\min_x & \frac{1}{2}x^\top P_0 x + q_0^\top x + r_0 \\
\text{subject to} & \frac{1}{2}x^\top P_0 x + q_0^\top x + r_0 \leqslant 0 \text{ for all } i \\
& Ax = b
\end{array}
$$

Second order cone program

$$
\begin{array}{rl}
\min_x & f^\top x \\
\text{subject to} & \|A_i x + b_i\|_2 \leqslant c_i^\top x + d_i \text{ for all } i \\
& Fx = g
\end{array}
$$

Semidefinite program

$$
\begin{array}{rl}
\min_x & c^\top x \\
\text{subject to} & x_1 F_1 + x_2 F_2 + \ldots + x_n F_n + G \preceq 0 \\
& Ax = b
\end{array}
$$

Too many programs?

Problem reformulations

- equivalent formulations of a problem can lead to very different duals
- reformulating the primal problem can be useful when the dual is difficult to derive, or uninteresting.

Common reformulations

- introduce new variables and equality constraints
- make explicit constraints implicit and vice versa
- transform objective or constraint functions

Equivalent convex problems (1)

Two problems are (informally) equivalent if the solution of one is readily obtained from the solution of the other, and vice-versa.

Eliminating equality constraints

$$\begin{aligned} \min_x \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \leqslant 0 \text{ for all } i \\ & Ax = b \end{aligned}$$

is equivalent to

$$\begin{aligned} \min_z \quad & f_0(Fz + x_0) \\ \text{subject to} \quad & f_i(Fz + x_0) \leqslant 0 \text{ for all } i \end{aligned}$$

where F and x_0 are such that $Ax = b \Leftrightarrow x = Fz + x_0$ for some z.

Equivalent convex problems (2)

Introducing Equality Constraints

$$\begin{array}{rl} \min_x & f_0(A_0 x + b) \\ \text{subject to} & f_i(A_i x + b) \text{ for all } i \end{array}$$

is equivalent to

$$\begin{array}{rl} \min_{x,y_i} & f_0(y_0) \\ \text{subject to} & f_i(y_i) \leqslant 0, \text{ for all } i \\ & y_i = A_i x + b_i \end{array}$$

Equivalent convex problems (3)

Introducing slack variables

$$\begin{array}{rl} \min_x & f_0(x) \\ \text{subject to} & a_i^\top x \leqslant b_i \text{ for all } i \end{array}$$

is equivalent to

$$\begin{array}{rl} \min_{x,s} & f_0(x) \\ \text{subject to} & a_i^\top x + s_i = b_i \text{ for all } i \\ & s_i \geqslant 0 \end{array}$$

Equivalent convex problems (4)

Epigraph form

$$
\begin{array}{rl}
\min_x & f_0(x) \\
\text{subject to} & f_i(x) \leqslant 0 \text{ for all } i \\
& Ax = b
\end{array}
$$

is equivalent to

$$
\begin{array}{rl}
\min_{x,t} & t \\
\text{subject to} & f_0(x) - t \leqslant 0 \\
& f_i(x) \leqslant 0 \text{ for all } i \\
& Ax = b
\end{array}
$$

Equivalent convex problems (5)

Minimizing over some variables

$$\begin{array}{rl} \min_{x_1,x_2} & f_0(x_1,x_2) \\ \text{subject to} & f_i(x_1) \leqslant 0 \text{ for all } i \end{array}$$

is equivalent to

$$\begin{array}{rl} \min_{x_1} & \tilde{f}_0(x_1) \\ \text{subject to} & f_i(x_1) \leqslant 0 \text{ for all } i \end{array}$$

where $\tilde{f}_0(x_1) = inf_{x_2} f_0(x_1, x_2)$.

Recall: Convex Optimization

$$
\begin{aligned}
\min_x \quad & f_0(x) \\
\text{subject to} \quad & f_i(x) \leqslant 0 \text{ for all } i \\
& g_j(x) = 0 \text{ for all } j
\end{aligned}
$$

- $x \in \mathbb{R}^n$ is the optimization variable
- $f_0 : \mathbb{R}^n \to \mathbb{R}$ is the objective or cost function
- $f_i : \mathbb{R}^n \to \mathbb{R}$ are inequality constraint functions
- $g_j : \mathbb{R}^n \to \mathbb{R}$ are equality constraint functions

Lagrange Duality

Lagrangian

$\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$ with

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

- weighted sum of objective and constraint functions
- λ_i is the Lagrange multiplier associated with $f_i(x) \leqslant 0$.
- ν_j is the Lagrange multiplier associated with $g_j(x) = 0$.

Dual function

Lagrange dual function

$h : \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$,

$$h(\lambda, \nu) = \inf_x \mathcal{L}(x, \lambda, \nu).$$

h is concave.

Lagrange dual problem

$$\begin{aligned} \max_{\lambda,\nu} \ & h(\lambda, \nu) \\ \text{subject to} \ & \lambda \geqslant 0 \end{aligned}$$

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

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

Remember kernels?

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

subject to the constraints:
(1) $\sum_{i=1}^{m} \alpha_i y_i = 0$
(2) $\alpha_i \geqslant 0$ for $i = 1, 2, \ldots, N$

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

subject to the constraints:
(1) $\sum_{i=1}^{m} \alpha_i y_i = 0$
(2) $\alpha_i \geqslant 0$ for $i = 1, 2, \ldots, N$

SVM 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

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

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

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, ...
- ...
- http://www.shogun-toolbox.org [Sonnenburg et al., 2006]
  - SVM-Light type QP optimization
  - Many string kernels implementations

Summary

- SVMs are convex optimization problems (specifically quadratic programs).
- Convex optimization problems ...
  - have a unique global minimum.
  - have equivalent primal and dual forms.
- For differentiable problems, can use gradient methods.
- SVMs are commonly solved in the dual form as quadratic programs.

We are using Python with `shogun` for the computer exercises

References

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

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.

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