Quadratic Programming and Kernel Methods

Jianlin Cheng, PhD
Computer Science Department
University of Missouri, Columbia
Fall, 2013
Reference

• C. Campbell. An Introduction to Kernel Methods

• Reading assignment: read the paper and submit a one page summary.
Quadratic Optimization

$y = x^2 - x - 2$

Optimization Approaches?
Kernel Methods

• A systematic and principled approach to training learning machines
• Achieve good generalization performance using statistical learning theory or Bayesian arguments
• Kernel methods for classification, regression and novelty detection
• Optimization of a convex cost function (quadratic programming)
• Applications
Classification Problem

• Diabetes data set:  
  
  http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary/diabetes

• First 10, 20, 30, ..., 50, or all the data points
Diabetes Data Example

Label

Input Feature Vector

\[ y_i, \langle d_1, ..., d_8 \rangle \]

1 \leq i \leq m
A Classification Learning Task

The learning task. Let us consider a binary classification task with datapoints $x_i \ (i = 1, \ldots, m)$ having corresponding labels $y_i = \pm 1$ and let the decision function be:

$$f(x) = \text{sign} \ (w \cdot x + b)$$

(1)

If the dataset is separable then the data will be correctly classified if $y_i(w \cdot x_i + b) > 0 \ \forall i$. Clearly this relation is invariant under a positive rescaling of the argument inside the $\text{sign}$-function, hence we can define a canonical hyperplane such that $w \cdot x + b = 1$ for the closest points on one side and $w \cdot x + b = -1$ for the closest on the other. For the separating
Visualization of Binary Classification

Figure 1. The *margin* is the perpendicular distance between the separating hyperplane and a hyperplane through the closest points (these are *support vectors*). The region between the hyperplanes on each side is called the *margin band*. $x_1$ and $x_2$ are examples of support vectors of opposite sign.
Figure 1. The *margin* is the perpendicular distance between the separating hyperplane and a hyperplane through the closest points (these are *support vectors*). The region between the hyperplanes on each side is called the *margin band*. $x_1$ and $x_2$ are examples of support vectors of opposite sign.
Figure 1. The *margin* is the perpendicular distance between the separating hyperplane and a hyperplane through the closest points (these are *support vectors*). The region between the hyperplanes on each side is called the *margin band*. $x_1$ and $x_2$ are examples of support vectors of opposite sign.
Figure 1. The *margin* is the perpendicular distance between the separating hyperplane and a hyperplane through the closest points (these are *support vectors*). The region between the hyperplanes on each side is called the *margin band*. $x_1$ and $x_2$ are examples of support vectors of opposite sign.
Maximize Margin

If the dataset is separable then the data will be correctly classified if \( y_i (w \cdot x_i + b) > 0 \ \forall i \). Clearly this relation is invariant under a positive rescaling of the argument inside the sign-function, hence we can define a canonical hyperplane such that \( w \cdot x + b = 1 \) for the closest points on one side and \( w \cdot x + b = -1 \) for the closest on the other. For the separating hyperplane \( w \cdot x + b = 0 \) the normal vector is clearly \( w/\|w\| \). Hence the margin is given by the projection of \( x_1 - x_2 \) onto this vector where \( x_1 \) and \( x_2 \) are closest points on opposite sides of the separating hyperplane (see Figure 1). Since \( w \cdot x_1 + b = 1 \) and \( w \cdot x_2 + b = -1 \) this means the margin is \( \gamma = 1/\|w\| \). To maximize the margin, the task is therefore:

\[
\min \left[ \frac{1}{2} \|w\|^2 \right]
\]  

subject to the constraints:

\[
y_i (w \cdot x_i + b) \geq 1 \quad \forall i
\]
Minimize Weights is Equivalent to Reducing the Function Complexity – Bayesian Perspective

• Smaller weights make function simpler
• Smaller weights make function smoother
• Pursuing smaller weights become a general paradigm in function optimization
Minimization of Primal Objective Function

\[ L = \frac{1}{2} (\mathbf{w} \cdot \mathbf{w}) - \sum_{i=1}^{m} \alpha_i (y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1) \]  

(4)

where \( \alpha_i \) are Lagrange multipliers (hence \( \alpha_i \geq 0 \)). Taking the derivatives with respect to \( b \) and \( \mathbf{w} \) gives:

\[ \sum_{i=1}^{m} \alpha_i y_i = 0 \]  

(5)

\[ \mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i \]  

(6)
Maximization of Dual Objective Function

\[
W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)
\]  

which must be maximized with respect to the \( \alpha_i \) subject to the constraint:

\[
\alpha_i \geq 0 \quad \sum_{i=1}^{m} \alpha_i y_i = 0
\]

This is a constrained quadratic optimization problem.

Questions: what if there are only two examples (one positive \( x^+ \) and one negative \( x^- \))?
Karush-Kuhn-Tucker (KKT) Condition

- \( w = \sum a_i y_i x_i \) \quad (1)
- \( \sum a_i y_i = 0 \) \quad (2)
- \( y_i(x_i . w + b) - 1 \geq 0, \quad i = 1, \ldots, l \) \quad (3)
- \( a_i \geq 0 \) \quad (4)
- \( a_i(y_i(x_i . w + b) - 1) = 0 \) \quad (5)

(5) is called complementary slackness due to the Lagrange theory and can be explained in intuition.
# Quadratic Programming Solver

<table>
<thead>
<tr>
<th>Name</th>
<th>Brief info</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIMMS</td>
<td>A popular modeling language for large-scale mathematical optimization.</td>
</tr>
<tr>
<td>AMPL</td>
<td>A popular modeling language for large-scale mathematical optimization.</td>
</tr>
<tr>
<td>EXCEL Solver Function</td>
<td>General-purpose programming language for mathematics. Solving a quadratic problem in Maple is accomplished via its QPSolve command.</td>
</tr>
<tr>
<td>GAMS</td>
<td>General-purpose programming language for mathematics. Solver with parallel algorithms for large-scale linear programs, quadratic programs and mixed-integer programs. Free for academic use.</td>
</tr>
<tr>
<td>IMSL</td>
<td>A set of mathematical and statistical functions that programmers can embed into their software applications.</td>
</tr>
<tr>
<td>Maple</td>
<td>A set of mathematical and statistical functions that programmers can embed into their software applications.</td>
</tr>
<tr>
<td>MATLAB</td>
<td>A general-purpose and matrix-oriented programming-language for numerical computing. Quadratic programming in MATLAB requires the Optimization Toolbox in addition to the base MATLAB product.</td>
</tr>
<tr>
<td>Mathematica</td>
<td>A general-purpose programming-language for mathematics, including symbolic and numerical capabilities.</td>
</tr>
<tr>
<td>MOSEK</td>
<td>A solver for large scale optimization with API for several languages (C++,java,.net, Matlab and python)</td>
</tr>
<tr>
<td>NAG Numerical Library</td>
<td>A collection of mathematical and statistical routines developed by the Numerical Algorithms Group for multiple programming languages (C, C++, Fortran, Visual Basic, Java and C#) and packages (MATLAB, Excel, R, LabVIEW). The Optimization chapter of the NAG Library includes routines for quadratic programming problems with both sparse and non-sparse linear constraint matrices, together with routines for the optimization of linear, nonlinear, sums of squares of linear or nonlinear functions with nonlinear, bounded or no constraints. The NAG Library has routines for both local and global optimization, and for continuous or integer problems.</td>
</tr>
<tr>
<td>OpenOpt</td>
<td>BSD licensed universal cross-platform numerical optimization framework, see its QP page and other problems involved. Uses NumPy arrays and SciPy sparse matrices.</td>
</tr>
<tr>
<td>OptimJ</td>
<td>Free Java-based Modeling Language for Optimization supporting multiple target solvers and available as an Eclipse plugin.[9][10]</td>
</tr>
<tr>
<td>TOMLAB</td>
<td>Supports global optimization, integer programming, all types of least squares, linear, quadratic and unconstrained programming for MATLAB. TOMLAB supports solvers like Gurobi, CPLEX, SNOP and KNITRO.</td>
</tr>
</tbody>
</table>
Kernel Substitution

Kernel substitution. This constrained quadratic programming (QP) problem will give an optimal separating hyperplane with a maximal margin if the data is separable. However, we have still not exploited the second observation from theorem 1: the error bound does not depend on the dimension of the space. This feature enables us to give an alternative kernel representation of the data which is equivalent to a mapping into a high dimensional space where the two classes of data are more readily separable. This space is called feature space and must be a pre-Hilbert or inner product space. For the dual objective function in (7) we notice that the datapoints, \( x_i \), only appear inside an inner product. Thus the mapping is achieved through a replacement of the inner product:

\[
x_i \cdot x_j \rightarrow \phi(x_i) \cdot \phi(x_j)
\]

(9)

The functional form of the mapping \( \phi(x_i) \) does not need to be known since it is implicitly defined by the choice of kernel:

\[
K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)
\]

(10)
Common Kernels

\[ K(x_i, x_j) = e^{-\|x_i - x_j\|^2 / 2\sigma^2} \] (11)

Other choices of kernel are possible, e.g.:

\[ K(x_i, x_j) = (x_i \cdot x_j + 1)^d \]  \[ K(x_i, x_j) = \tanh(\beta x_i \cdot x_j + b) \] (12)
Kernel-Based Quadratic Optimization

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

subject to the constraints of Equation (8). The associated Karush-Kuhn-Tucker (KKT) conditions are:

$$y_i (w \cdot x_i + b) - 1 \geq 0 \quad \forall i$$

$$\alpha_i \geq 0 \quad \forall i$$

$$\alpha_i (y_i (w \cdot x_i + b) - 1) = 0 \quad \forall i$$

which are always satisfied when a solution is found. Test examples are evaluated using a decision function given by the sign of:

$$f(z) = \sum_{i=1}^{m} y_i \alpha_i K(x_i, z) + b$$

Can weight vector w be explicitly represented?
Allow for Training Error

a positive slack variable $\xi_i$:

$$y_i \left( \mathbf{w} \cdot \mathbf{x}_i + b \right) \geq 1 - \xi_i$$  \hspace{1cm} (20)

and the task is now to minimize the sum of errors $\sum_{i=1}^{m} \xi_i$ in addition to $\|\mathbf{w}\|^2$:

$$\min \left[ \frac{1}{2} \mathbf{w} \cdot \mathbf{w} + C \sum_{i=1}^{m} \xi_i \right]$$  \hspace{1cm} (21)

This is readily formulated as a primal objective function:

$$L(\mathbf{w}, b, \alpha, \xi) = \frac{1}{2} \mathbf{w} \cdot \mathbf{w} + C \sum_{i=1}^{m} \xi_i$$

$$- \sum_{i=1}^{m} \alpha_i \left[ y_i \left( \mathbf{w} \cdot \mathbf{x}_i + b \right) - 1 + \xi_i \right] - \sum_{i=1}^{m} r_i \xi_i$$  \hspace{1cm} (22)
Quadratic Optimization Function
Tolerating Training Errors

\[
\frac{\partial L}{\partial w} = w - \sum_{i=1}^{m} \alpha_i y_i x_i = 0
\]  \hspace{1cm} (23)

\[
\frac{\partial L}{\partial b} = \sum_{i=1}^{m} \alpha_i y_i = 0
\]  \hspace{1cm} (24)

\[
\frac{\partial L}{\partial \xi_i} = C - \alpha_i - r_i = 0
\]  \hspace{1cm} (25)

\[
W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y_i y_j \alpha_i \alpha_j K(x_i, x_j)
\]
How can we identify the data points needing slackness?

KKT Conditions

1. $\partial_w \mathcal{L}_P = 0 \rightarrow w - \sum_i \alpha_i y_i x_i = 0$

2. $\partial_b \mathcal{L}_P = 0 \rightarrow \sum_i \alpha_i y_i = 0$

3. $\partial_{\xi} \mathcal{L}_P = 0 \rightarrow C - \alpha_i - \mu_i = 0$

4. constraint-1
5. constraint-2

6. multiplier condition-1
7. multiplier condition-2

8. complementary slackness-1

9. complementary slackness-1

Max Welling, 2005
Novelty Detection

• Novelty detection: identification of new or unknown data that a machine learning system has not been trained with and was not previously aware of.

• Application: detection of a disease or potential fault whose class may be under-represented in the training data.
An Novelty Detection Approach

• Find a hypersphere with a minimal radius $R$ and center $a$ which contains most of the data: novel test points lie outside the boundary of this hypersphere.

• The effect of outliers is reduced by using slack variables $\xi_i$ to allow for data points outside the sphere and the task is to minimize the volume of the sphere and number of data points outside.
Minimization of Objective

$$\min \left[ R^2 + \frac{1}{m\nu} \sum_i \xi_i \right]$$

subject to the constraints:

$$(x_i - a)^T(x_i - a) \leq R^2 + \xi_i$$
Primal and Dual Objective Function

The primal objective function is then:

\[
L(R, a, \alpha_i, \xi_i) = R^2 + \frac{1}{m\nu} \sum_{i=1}^{m} \xi_i \\
- \sum_{i=1}^{m} \alpha_i \left( R^2 + \xi_i - (x_i \cdot x_i - 2a \cdot x_i + a \cdot a) \right) - \sum_{i=1}^{m} \gamma_i \xi_i \tag{28}
\]

with \( \alpha_i \geq 0 \) and \( \gamma_i \geq 0 \). After kernel substitution the dual formulation amounts to maximization of:

\[
W(\alpha) = \sum_{i=1}^{m} \alpha_i K(x_i, x_i) - \sum_{i,j=1}^{m} \alpha_i \alpha_j K(x_i, x_j) \tag{29}
\]
with respect to $\alpha_i$ and subject to $\sum_{i=1}^{m} \alpha_i = 1$ and $0 \leq \alpha_i \leq 1/mv$. If $mv > 1$ then at bound examples will occur with $\alpha_i = 1/mv$ and these correspond to outliers in the training process. Having completed the training process a test point $z$ is declared novel if:

$$K(z, z) - 2 \sum_{i=1}^{m} \alpha_i K(z, x_i) + \sum_{i,j=1}^{m} \alpha_i \alpha_j K(x_i, x_j) - R^2 \geq 0 \quad \text{(30)}$$
For classification, regression or novelty detection we see that the learning task involves optimization of a quadratic cost function and thus techniques from quadratic programming are most applicable including quasi-Newton, conjugate gradient and primal-dual interior point methods. Certain QP packages are readily applicable such as MINOS and LOQO. These methods can be used to train an SVM rapidly but they have the disadvantage that the kernel matrix is stored in memory. For small datasets this is practical and QP routines are the best choice, but for larger datasets alternative techniques have to be used. These split into two categories:
algorithm [15]. For binary classification (with no soft margin or bias) this is a simple gradient ascent procedure on (13) in which $\alpha_i > 0$ initially and the $\alpha_i$ are subsequently sequentially updated using:

$$\alpha_i \leftarrow \beta_i \theta (\beta_i) \quad \text{where} \quad \beta_i = \alpha_i + \eta \left(1 - y_i \sum_{j=1}^{m} \alpha_j y_j K(x_i, x_j)\right) \quad (54)$$

and $\theta(\beta)$ is the Heaviside step function. The optimal learning rate $\eta$ can be readily evaluated: $\eta = 1/K(x_i, x_i)$ and a sufficient condition for convergence is $0 < \eta K(x_i, x_i) < 2$. With the given decision function of

$$W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y_i y_j \alpha_i \alpha_j K(x_i, x_j)$$
Decomposition and Sequential Minimal Optimization

Decomposition method provide a better approach: only use a fixed size subset of data, with $a_i$ for the remainder for fixed.

The limiting case of decomposition is the Sequential Minimal Optimization (SMO) algorithm of Platt [33] in which only two $\alpha_i$ are optimized at each iteration. The smallest set of parameters which can be optimized with each iteration is plainly two if the constraint $\sum_{i=1}^{m} \alpha_i y_i = 0$ is to hold. Remarkably, if only two parameters are optimized and the rest kept fixed then it is possible to derive an analytical solution which can be executed using few numerical operations. The algorithm therefore selects two La-
Consider a binary classification problem with a dataset \((x_1, y_1), \ldots, (x_n, y_n)\), where \(x_i\) is an input vector and \(y_i \in \{-1, +1\}\) is a binary label corresponding to it. A soft-margin support vector machine is trained by solving a quadratic programming problem, which is expressed in the dual form as follows:

\[
\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j K(x_i, x_j) \alpha_i \alpha_j,
\]

subject to:

\[
0 \leq \alpha_i \leq C, \quad \text{for } i = 1, 2, \ldots, n,
\]

\[
\sum_{i=1}^{n} y_i \alpha_i = 0
\]

where \(C\) is an SVM hyperparameter and \(K(x_i, x_j)\) is the kernel function, both supplied by the user; and the variables \(\alpha_i\) are Lagrange multipliers.
SMO is an iterative algorithm for solving the optimization problem described above. SMO breaks this problem into a series of smallest possible sub-problems, which are then solved analytically. Because of the linear equality constraint involving the Lagrange multipliers $\alpha_i$, the smallest possible problem involves two such multipliers. Then, for any two multipliers $\alpha_1$ and $\alpha_2$, the constraints are reduced to:

$$0 \leq \alpha_1, \alpha_2 \leq C,$$
$$y_1 \alpha_1 + y_2 \alpha_2 = k,$$

and this reduced problem can be solved analytically: one needs to find a minimum of a one-dimensional quadratic function. $k$ is the sum over the rest of terms in the equality constraint, which is fixed in each iteration.

The algorithm proceeds as follows:

1. Find a Lagrange multiplier $\alpha_1$ that violates the Karush–Kuhn–Tucker (KKT) conditions for the optimization problem.
2. Pick a second multiplier $\alpha_2$ and optimize the pair $(\alpha_1, \alpha_2)$.
3. Repeat steps 1 and 2 until convergence.

When all the Lagrange multipliers satisfy the KKT conditions (within a user-defined tolerance), the problem has been solved. Although this algorithm is guaranteed to converge, heuristics are used to choose the pair of multipliers so as to accelerate the rate of convergence.
Keep all other terms except $a_1, a_2$ fixed, get a new function like

$$a_1 + (k-a_1) + \text{other terms} - \frac{1}{2} (y_1 y_1 a_1 a_1 K(x_1, x_1) + y_2 y_2 a_2 a_2 K(x_2, x_2) 2y_1 y_2 a_1 a_1 (k-a_1) K(x_1, x_2) ) \times y_1 a_1 \times \text{Other}$$
Optimize Single Variable Quadratic Function

Check three values:

0, C, and x-coordinate of vertex

\[ f(x) = -\frac{1}{2}(x-1)^2 + 2 \]