Support Vector Machines

Eric Xing

Lecture 6, September 26, 2016

Reading: Chap. 6&7, C.B book, and listed papers
What is a good Decision Boundary?

- Consider a binary classification task with $y = \pm 1$ labels (not 0/1 as before).
- When the training examples are linearly separable, we can set the parameters of a linear classifier so that all the training examples are classified correctly.
- Many decision boundaries!
  - Generative classifiers
  - Logistic regressions …
- Are all decision boundaries equally good?
What is a good Decision Boundary?
Not All Decision Boundaries Are Equal!

- Why we may have such boundaries?
  - Irregular distribution
  - Imbalanced training sizes
  - Outliners
Classification and Margin

- Parameterizing decision boundary
  - Let $w$ denote a vector orthogonal to the decision boundary, and $b$ denote a scalar "offset" term, then we can write the decision boundary as:

$$w^T x + b = 0$$

where $w^T$ is the transpose of $w$. The distance from a point $x$ to the decision boundary is given by $d = \frac{|w^T x + b|}{||w||}$. The margin $d^+$ and $d^-$ represent the distances from the closest points of Class 1 and Class 2, respectively, to the decision boundary.
Classification and Margin

- Parameterizing decision boundary
  - Let $w$ denote a vector orthogonal to the decision boundary, and $b$ denote a scalar "offset" term, then we can write the decision boundary as:

  $\frac{w^T x_i + b}{||w||} > \frac{c}{||w||}$ for all $x_i$ in class 2
  $\frac{w^T x_i + b}{||w||} < -\frac{c}{||w||}$ for all $x_i$ in class 1

  Or more compactly:

  $\frac{(w^T x_i + b)}{||w||} > \frac{c}{||w||}$

- Margin

  The margin between any two points $m = d^- + d^+$
Maximum Margin Classification

- The minimum permissible margin is:
  \[ m = \frac{w^T}{\|w\|} (x_i^* - x_j^*) = \frac{2c}{\|w\|} \]

- Here is our Maximum Margin Classification problem:

\[
\begin{align*}
\max_w & \quad \frac{2c}{\|w\|} \\
\text{s.t} & \quad y_i(w^T x_i + b) / \|w\| \geq c / \|w\|, \quad \forall i
\end{align*}
\]
Maximum Margin Classification, con'd.

- The optimization problem:

\[
\max_{w,b} \quad \frac{c}{\|w\|}
\]

\[
s.t \quad y_i(w^T x_i + b) \geq c, \ \forall i
\]

- But note that the magnitude of \(c\) merely scales \(w\) and \(b\), and does not change the classification boundary at all! (why?)

- So we instead work on this cleaner problem:

\[
\max_{w,b} \quad \frac{1}{\|w\|}
\]

\[
s.t \quad y_i(w^T x_i + b) \geq 1, \ \forall i
\]

- The solution to this leads to the famous Support Vector Machines -- believed by many to be the best "off-the-shelf" supervised learning algorithm.
Support vector machine

- A convex quadratic programming problem with linear constraints:

\[
\begin{align*}
\max_{w,b} & \quad \frac{1}{\|w\|} \\
\text{s.t. } & \quad y_i(w^T x_i + b) \geq 1, \quad \forall i
\end{align*}
\]

- The attained margin is now given by \(\frac{1}{\|w\|}\)
- Only a few of the classification constraints are relevant \(\Rightarrow\) support vectors

- Constrained optimization
  - We can directly solve this using commercial quadratic programming (QP) code
  - But we want to take a more careful investigation of Lagrange duality, and the solution of the above in its dual form.
  \(\Rightarrow\) deeper insight: support vectors, kernels …
  \(\Rightarrow\) more efficient algorithm
Digression to Lagrangian Duality

- The Primal Problem

Primal:

\[
\min_w \ f(w)
\]

s.t. \( g_i(w) \leq 0, \ i = 1, \ldots, k \)

\( h_i(w) = 0, \ i = 1, \ldots, l \)

The generalized Lagrangian:

\[
\mathcal{L}(w, \alpha, \beta) = f(w) + \sum_{i=1}^{k} \alpha_i g_i(w) + \sum_{i=1}^{l} \beta_i h_i(w)
\]

the \( \alpha \)'s (\( \alpha_i \geq 0 \)) and \( \beta \)'s are called the Lagrangian multipliers

Lemma:

\[
\max_{\alpha, \beta, \alpha_i \geq 0} \mathcal{L}(w, \alpha, \beta) = \begin{cases} 
  f(w) & \text{if } w \text{ satisfies primal constraints} \\
  \infty & \text{o/w}
\end{cases}
\]

A re-written Primal:

\[
\min_w \max_{\alpha, \beta, \alpha_i \geq 0} \mathcal{L}(w, \alpha, \beta)
\]
Lagrangian Duality, cont.

- Recall the Primal Problem:
  \[
  \min_w \max_{\alpha, \beta, \alpha_i \geq 0} \mathcal{L}(w, \alpha, \beta)
  \]

- The Dual Problem:
  \[
  \max_{\alpha, \beta, \alpha_i \geq 0} \min_w \mathcal{L}(w, \alpha, \beta)
  \]

- Theorem (weak duality):
  \[
  d^* = \max_{\alpha, \beta, \alpha_i \geq 0} \min_w \mathcal{L}(w, \alpha, \beta) \leq \min_w \max_{\alpha, \beta, \alpha_i \geq 0} \mathcal{L}(w, \alpha, \beta) = p^*
  \]

- Theorem (strong duality):
  If there exist a saddle point of \( \mathcal{L}(w, \alpha, \beta) \), we have
  \[
  d^* = p^*
  \]
A sketch of strong and weak duality

- Now, ignoring $h(x)$ for simplicity, let's look at what's happening graphically in the duality theorems.

$$d^* = \max_{\alpha_i \geq 0} \min_w f(w) + \alpha^T g(w) \leq \min_w \max_{\alpha_i \geq 0} f(w) + \alpha^T g(w) = p^*$$
A sketch of strong and weak duality

- Now, ignoring $h(x)$ for simplicity, let's look at what's happening graphically in the duality theorems.

\[
d^* = \max_{\alpha_i \geq 0} \min_w f(w) + \alpha^T g(w) \leq \min_w \max_{\alpha_i \geq 0} f(w) + \alpha^T g(w) = p^*
\]
A sketch of strong and weak duality

- Now, ignoring $h(x)$ for simplicity, let's look at what's happening graphically in the duality theorems.

$$d^* = \max_{\alpha_i \geq 0} \min_w f(w) + \alpha^T g(w) \leq \min_w \max_{\alpha_i \geq 0} f(w) + \alpha^T g(w) = p^*$$
The KKT conditions

- If there exists some saddle point of $\mathcal{L}$, then the saddle point satisfies the following "Karush-Kuhn-Tucker" (KKT) conditions:

$$\frac{\partial}{\partial w_i} \mathcal{L}(w, \alpha, \beta) = 0, \quad i = 1, \ldots, k$$

$$\frac{\partial}{\partial \beta_i} \mathcal{L}(w, \alpha, \beta) = 0, \quad i = 1, \ldots, l$$

$$\alpha_i g_i(w) = 0, \quad i = 1, \ldots, m \quad \text{Complementary slackness}$$

$$g_i(w) \leq 0, \quad i = 1, \ldots, m \quad \text{Primal feasibility}$$

$$\alpha_i \geq 0, \quad i = 1, \ldots, m \quad \text{Dual feasibility}$$

- **Theorem:** If $w^*$, $\alpha^*$ and $\beta^*$ satisfy the KKT condition, then it is also a solution to the primal and the dual problems.
Solving optimal margin classifier

- Recall our opt problem:
  \[ \max_{w,b} \frac{1}{\|w\|} \]
  \[ \text{s.t.} \quad y_i(w^T x_i + b) \geq 1, \quad \forall i \]

- This is equivalent to
  \[ \min_{w,b} \frac{1}{2} w^T w \]
  \[ \text{s.t.} \quad 1 - y_i(w^T x_i + b) \leq 0, \quad \forall i \]

- Write the Lagrangian:
  \[ \mathcal{L}(w, b, \alpha) = \frac{1}{2} w^T w - \sum_{i=1}^{m} \alpha_i [y_i(w^T x_i + b) - 1] \]

  - Recall that (*) can be reformulated as \( \min_{w,b} \max_{\alpha \geq 0} \mathcal{L}(w, b, \alpha) \)
  
  Now we solve its dual problem: \( \max_{\alpha \geq 0} \min_{w,b} \mathcal{L}(w, b, \alpha) \)
The Dual Problem

\[ \mathcal{L}(w, b, \alpha) = \frac{1}{2} w^T w - \sum_{i=1}^{m} \alpha_i [y_i (w^T x_i + b) - 1] \]

\[ \max_{\alpha_i \geq 0} \min_{w, b} \mathcal{L}(w, b, \alpha) \]

- We minimize \( \mathcal{L} \) with respect to \( w \) and \( b \) first:

\[ \nabla_w \mathcal{L}(w, b, \alpha) = w - \sum_{i=1}^{m} \alpha_i y_i x_i = 0, \quad (\ast) \]

\[ \nabla_b \mathcal{L}(w, b, \alpha) = \sum_{i=1}^{m} \alpha_i y_i = 0, \quad (\ast\ast) \]

Note that \( (\ast) \) implies:

\[ w = \sum_{i=1}^{m} \alpha_i y_i x_i, \quad (\ast\ast\ast) \]

- Plug \( (\ast\ast\ast) \) back to \( \mathcal{L} \), and using \( (\ast\ast) \), we have:

\[ \mathcal{L}(w, b, \alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (x_i^T x_j) \]
The Dual problem, cont.

- Now we have the following dual opt problem:

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

s.t. \(\alpha_i \geq 0, \quad i = 1, \ldots, k\)

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

- This is, (again,) a quadratic programming problem.
  - A global maximum of \(\alpha_i\) can always be found.
  - But what's the big deal??
  - Note two things:
    1. \(w\) can be recovered by \(w = \sum_{i=1}^{m} \alpha_i y_i x_i\) See next …
    2. The "kernel" \(x_i^T x_j\) More later …
Support vectors

- Note the KKT condition --- only a few $\alpha_i$'s can be nonzero!!

$$\alpha_i g_i(w) = 0, \quad i = 1, \ldots, m$$

Call the training data points whose $\alpha_i$'s are nonzero the support vectors (SV)
Support vector machines

- Once we have the Lagrange multipliers \( \{ \alpha_i \} \), we can reconstruct the parameter vector \( w \) as a weighted combination of the training examples:

\[
w = \sum_{i \in SV} \alpha_i y_i x_i
\]

- For testing with a new data \( z \)
  - Compute
  \[
w^T z + b = \sum_{i \in SV} \alpha_i y_i (x_i^T z) + b
  \]
  and classify \( z \) as class 1 if the sum is positive, and class 2 otherwise
  - Note: \( w \) need not be formed explicitly

© Eric Xing @ CMU, 2006-2016
Interpretation of support vector machines

- The optimal $w$ is a linear combination of a small number of data points. This “sparse” representation can be viewed as data compression as in the construction of kNN classifier.

- To compute the weights $\{\alpha_i\}$, and to use support vector machines we need to specify only the inner products (or kernel) between the examples $x_i^T x_j$.

- We make decisions by comparing each new example $z$ with only the support vectors:

$$y^* = \text{sign} \left( \sum_{i \in SV} \alpha_i y_i (x_i^T z) + b \right)$$
Non-linearly Separable Problems

- We allow “error” $\xi_i$ in classification; it is based on the output of the discriminant function $w^T x + b$
- $\xi_i$ approximates the number of misclassified samples
Soft Margin Hyperplane

- Now we have a slightly different opt problem:

\[
\begin{align*}
\min_{w,b} & \quad \frac{1}{2} w^T w + C \sum_{i=1}^{m} \xi_i \\
\text{s.t} & \quad y_i (w^T x_i + b) \geq 1 - \xi_i, \quad \forall i \\
& \quad \xi_i \geq 0, \quad \forall i
\end{align*}
\]

- ξ_i are “slack variables” in optimization
- Note that ξ_i=0 if there is no error for x_i
- ξ_i is an upper bound of the number of errors
- C : tradeoff parameter between error and margin
The Optimization Problem

- The dual of this new constrained optimization problem is

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

s.t. \quad 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, m

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

- This is very similar to the optimization problem in the linear separable case, except that there is an upper bound \(C\) on \(\alpha_i\) now

- Once again, a QP solver can be used to find \(\alpha_i\)
The SMO algorithm

- Consider solving the unconstrained opt problem:

\[
\max_{\alpha} W(\alpha_1, \alpha_2, \ldots, \alpha_m)
\]

- We’ve already see three opt algorithms!
  - ?
  - ?
  - ?

- Coordinate ascend:
Coordinate ascend
Sequential minimal optimization

- Constrained optimization:

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

s.t. \( 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, m \)

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

- Question: can we do coordinate along one direction at a time (i.e., hold all \( \alpha_{[\neq i]} \) fixed, and update \( \alpha_i \)?)
The SMO algorithm

Repeat till convergence

1. Select some pair $\alpha_i$ and $\alpha_j$ to update next (using a heuristic that tries to pick the two that will allow us to make the biggest progress towards the global maximum).

2. Re-optimize $J(\alpha)$ with respect to $\alpha_i$ and $\alpha_j$, while holding all the other $\alpha_k$'s ($k \neq i, j$) fixed.

Will this procedure converge?
Convergence of SMO

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

\[ \text{s.t.} \quad 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, k \]

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

- Let's hold \( \alpha_3, \ldots, \alpha_m \) fixed and reopt J w.r.t. \( \alpha_1 \) and \( \alpha_2 \)
Convergence of SMO

- The constraints:
  \[ \alpha_1 y_1 + \alpha_2 y_2 = \xi \]
  \[ 0 \leq \alpha_1 \leq C \]
  \[ 0 \leq \alpha_2 \leq C \]

- The objective:
  \[ J(\alpha_1, \alpha_2, \ldots, \alpha_m) = J((\xi - \alpha_2 y_2) y_1, \alpha_2, \ldots, \alpha_m) \]

- Constrained opt:
Cross-validation error of SVM

- The leave-one-out cross-validation error does not depend on the dimensionality of the feature space but only on the # of support vectors!

\[
\text{Leave-one-out CV error} = \frac{\text{# support vectors}}{\text{# of training examples}}
\]
Summary

- Max-margin decision boundary

- Constrained convex optimization
  - Duality
  - The KTT conditions and the support vectors
  - Non-separable case and slack variables
  - The SMO algorithm