CSCI 181 / E-181 Spring 2014

2nd midterm review

David Wihl
davidwihl@gmail.com

April 27, 2014

1 Support Vector Machines

1.1 Background

Characteristics of SVMs:

- *stock* – SVMs are "off the shelf" and ready to use. No special modification is necessary.
- *linearly separable* – assumes that linear separation is possible. Used natively as a binary classifier.
- *convex optimization.* SVM originated as a backlash against neural nets due to nets' non-convexity. In Neural Nets, results were often non-reproducible as different researchers found different results due to different initializations.
- *global optimum* – SVMs will find the global optimum.

SVMs are based on three "big ideas":

- *margin* Maximizes distance between the closest points
- *duality* Take a hard problem and transform it into an easier problem to solve.
- *kernel trick* Map input vectors to higher dimensional, more expressive features.

1.2 Definitions

Data: $\{x_n, t_n\}_{n=1}^N, t_n \in \{-1, +1\}$. t_n is the target or the expected result of the classification.

J Basis functions: $\phi_j(x) \to \Re$, therefore

Vector function: $\Phi X \to \Re^J$ produces a column vector.

Objective function: $f(\mathbf{x}, \mathbf{w}, b) = \mathbf{w}^\intercal \phi(\mathbf{x}) + b$ where b is the bias.

The sign of $f(\cdot)$ will determine classification $(-1, +1)$

So the actual classifier will be:

$$y(\mathbf{x}, \mathbf{w}, b) = \begin{cases} +1, & \text{if } \mathbf{w}^\intercal \phi(\mathbf{x}) + b > 0 \\ -1, & \text{otherwise} \end{cases}$$

Unlike Logistic Regression (which uses $\{0, 1\}$), it is preferable to use $\{-1, +1\}$ as the classification result. If $t_n * y$ is positive, then the produced classification is correct (positive × positive is positive, negative × negative is also positive).

Decision Boundary is the hyperplane where $\mathbf{w}^\intercal \phi(\mathbf{x}) + b = 0$. We want to find the Decision Boundary that creates the most separation between the two different classes by maximizing the distance between the two closest points. The distance between the Decision Boundary and the closest point is called the *margin.* The points closest to the Decision Boundary are called the *support vectors.*

1.3 Max Marginalization

The margin is determined by the orthogonal distance from the closest point to the Decision Boundary:

$$\frac{|\mathbf{w}^\intercal \mathbf{x} + b|}{||\mathbf{w}||} \tag{1}$$

Maximizing the margin can be written as:

$$\underset{w,b}{\operatorname{argmax}} \left\{ \min_n (t_n \cdot (\mathbf{w}^\intercal \mathbf{x} + b)) \cdot \frac{1}{||w||} \right\} \tag{2}$$

Maximizing the margin helps ensure that points which are close to margin will not be pushed over the boundary by noise.

$\mathbf{w}$ is orthogonal to vectors in the Decision Boundary. Here's how: pick two points on the

Decision Boundary $\phi(x_1)$ and $\phi(x_2)$. So

$$\begin{aligned}
&\mathbf{w}^\intercal \left(\phi(x_2) - \phi(x_1)\right) = 0 \text{ for orthogonal dot product}\\
&\mathbf{w}^\intercal \phi(x_2) - \mathbf{w}^\intercal \phi(x_1) = 0\\
\text{Note: } &\mathbf{w}^\intercal \phi(x_n) = (-b) \text{, so}\\
&=(-b) - (-b)\\
&=0
\end{aligned}$$

Since $\mathbf{w}$ is orthogonal, we want to maximize it. It is not unit length, but could be, by scaling with a factor of r.

See Stanford CS229 SVM Notes re: Functional vs. Geometric Margins

The support vector is defined as $r\frac{\mathbf{w}}{||\mathbf{w}||_2}$, where r is multiplied by the unit vector orthogonal to the Decision Boundary hyperplane.

Define the point where the vector meets the plane as $\phi_\perp(\mathbf{x})$, so

$$\phi(\mathbf{x}) = \phi_\perp(\mathbf{x}) + r\frac{\mathbf{w}}{||\mathbf{w}||_2} \tag{3}$$

Solving for r, multiple both sides by $\mathbf{w}^\intercal$.

(Recall: $\mathbf{w}^\intercal\mathbf{w} = ||\mathbf{w}||^2$)

$$\begin{aligned}
\mathbf{w}^\intercal\phi(\mathbf{x}) =& \mathbf{w}^\intercal\phi_\perp(\mathbf{x}) + r\frac{\mathbf{w}^\intercal\mathbf{w}}{||\mathbf{w}||} && (4)\\
=&(-b) + r||\mathbf{w}|| && (5)
\end{aligned}$$

Therefore, the margin for a point $\mathbf{x}$.

$$\begin{aligned}
r =& \frac{\phi(\mathbf{x})^\intercal\mathbf{w} + b}{||w||} && (6)\\
=& \frac{f(\mathbf{x}, \mathbf{w}, b)}{||w||} && (7)
\end{aligned}$$

This makes it easy to calculate how far away a point is from the Decision Boundary. r is strictly not a length because it could be negative. However, we only care about the actual distance to the boundary.

Margin for a datum n:

$$margin = t_n\frac{\phi(\mathbf{x})^\intercal\mathbf{w} + b}{||\mathbf{w}||} \tag{8}$$

This is getting close to a loss function as we can now figure out the worst of these. The Margin for all the training data will be the point closest to the Decision Boundary:

$$min_n \left\{ t_n \frac{\phi(\mathbf{x})^\intercal \mathbf{w} + b}{||\mathbf{w}||} \right\} \tag{9}$$

As mentioned at the beginning of this section, the Objective Function is

$$\mathbf{w}^*, b^* = \underset{w,b}{\operatorname{argmax}} \left\{ \min_n (t_n \cdot (\phi(\mathbf{x})^\intercal \mathbf{w} + b)) \cdot \frac{1}{||w||} \right\} \tag{10}$$

but now we can simplify some things. $\mathbf{w}$ are b are scale free (if we multiply by some β, the max and min will still be the same.)

Let's define a set of linear constraints such that the margin is always ≥ 1 to make this easier to solve.

$$\mathbf{w}^*, b^* = \underset{\mathbf{w},b}{\operatorname{argmax}} \frac{1}{||w||} \tag{11}$$

subject to

$$t_n \cdot (\phi(x_n)^\intercal \mathbf{w} + b) \geq 1 \, \forall \, n \tag{12}$$

This can be made even easier. Finding the max of $\frac{1}{||w||}$ is like finding the min of $||w||^2$, so

$$\mathbf{w}^*, b^* = \underset{\mathbf{w},b}{\operatorname{argmin}} \, ||w||^2 \tag{13}$$

$$\text{s.t. } t_n(\phi(x_n)^\intercal \mathbf{w} + b) \geq 1 \tag{14}$$

so this reduces to just a quadratic program (QP) with linear constraints that could be solved by any number of commercial packages and produces a global minimum.

1.4 Slack Variables

If the data is not strictly linearly separable, it can mitigated by slack variables.

$\xi_n \leftarrow$ one for each datum.

$\xi_n = 0$ then the datum is correctly classified and outside the margin.

$0 < \xi_n <= 1$ the datum is correctly classified and within the margin

$\xi_n > 1$ the datum is misclassified

We will now add ξ as a constraint to minimize.

$$t_n \cdot (\phi(x_n)^\intercal \mathbf{w} + b) \geq 1 - \xi_n \tag{15}$$

New objective function:

$$\mathbf{w}^*, b^*, \xi^* = \underset{\mathbf{w}, b, \xi}{\operatorname{argmin}} \left\{ ||w||^2 + c \sum_{n=1}^{N} \xi_n \right\} \tag{16}$$

$$\text{s.t. } t_n(\phi(x_n)^\intercal \mathbf{w} + b) \geq 1 - \xi_n \tag{17}$$

$$\xi \geq 0 \tag{18}$$

$$\forall\, n \tag{19}$$

where $c > 0$ is the regularization parameter. A small C means that you don't care much about errors. The sum of ξ is the upper bound on how many can be wrong. If $c = 0$, it becomes the original function. Typically,

$$c = \frac{1}{\nu N}, \text{ where } 0 < \nu \leq 1 \tag{20}$$

where ν is the tolerance for percentage willing to get wrong.

1.5 Duality

1.6 Kernel Tricks

Mercer function, infinite dimensions (justification for duality)

Slack variables to break the linear separability.

1.7 Sources

1. Lecture 14, March 24, 2014
2. Lecture 15
3. Bishop 6.0-6.2
4. Bishop 7.0-7.1
5. Course notes - maxmargin
6. Section 7 review
7. Section 8 review

8. Stanford CS229 SVM notes
9. Machine Learning in Action, Chapter 6

2 Markov Decision Processes

Lecture 16

Course notes - MDP

Section 9

2.1 Partially Observable MDP

Course notes - POMDP

Section 10

2.2 Hidden Markov Models

Bishop 13.0-13.2

2.3 Mixture Models

Bishop 9.0-9.2

3 Reinforcement Learning

Course notes - RL

Section 9

3.1 Value and Policy Iteration

Lecture 17

Course notes - policyiter

4 Expectation Maximization

Bishop 9.3

Section 11