Training Linear SVMs’

By - Thorsten Joachims
Prasad Seemakurthi
Agenda

• What is SVM
• Kernel
• Hard Margins
• Soft Margins
• Linear Algorithm
• Few Examples
• Conclusion
SVM – Curtain Raiser

• Linear Classification Algorithm

• SVM have a clever way to prevent over-fitting

• SVMs have a very clever way to use huge number of features nearly as much as computation as seems to be necessary
Linear Classifiers (Intuition)

\[ f(x, w, b) = \text{sign}(w \cdot x + b) \]

- \( \alpha \) denotes +1
- \( \circ \) denotes -1

How would you classify this data?
Linear Classifiers

Any of these would be fine ...

But which is best ... ?
Linear Classifier

- denotes +1
  - denotes -1

\[ f(x,w,b) = \text{sign}(w \cdot x + b) \]

How would you classify this data?

Misclassified to +1 class
Maximizing margin is good according to intuition and PAC theory.

2. Implies only support vectors are important.

3. Empirically works well.

Classifier with the maximum margin.

This kind of simplest kind of SVM is called Linear SVM.
Maximizing the margin
Why maximize the margin?

Points near decision surface ----> Uncertain classification decision (50% either way)

A classifier with a large margin make no low classification decision

Gives classification safety margin w.r.t slight errors in measurement
Why maximize the margin?

• SVM Classifier: Large Margin around Decision boundary

• Compare to decision hyperplane: Place a fat separator between classes

• Fewer choices of where it can be put
  • Decreased memory capacity
  • Increased ability to correctly generalize the test data
Linear SVM mathematically

What we know:

- $\mathbf{w} \cdot \mathbf{x}^+ + b = +1$
- $\mathbf{w} \cdot \mathbf{x}^- + b = -1$
- $\mathbf{w} \cdot (\mathbf{x}^+ - \mathbf{x}^-) = 2$

$$M = \frac{(x^+ - x^-) \cdot \mathbf{w}}{||\mathbf{w}||} = \frac{2}{||\mathbf{w}||}$$
Linear SVM mathematically

Goal: 1) Correctly classify all training data

\[ wx_i + b \geq 1 \text{ if } y_i = +1 \]
\[ wx_i + b \leq 1 \text{ if } y_i = -1 \]

\[ y_i (wx_i + b) \geq 1 \text{ for all } i \]

2) Maximize the Margin

\[ M = \frac{2}{|w|} \]

same as minimize

\[ \frac{1}{2} w^T w \]

- This is a Quadratic Optimization Problem in linear constraints. Solve for \( w \) & \( b \).

- Minimize \( \Phi(w) = \frac{1}{2} w^T w \)

subject to \( y_i (wx_i + b) \geq 1 \) \( \forall i \)
Linear (hard – Margin ) SVM – formulation

- Find $w,b$ that solves
  \[ \min \frac{1}{2} \|w\|^2 \]
  \[ s.t. \ y_i(w \cdot x_i + b) \geq 1, \ \forall x_i \]
- Problem is convex so, there is a unique global minimum value (when feasible)
- Non-solvable if the data is not linearly separable
- Quadratic Programming
  - Very efficient computationally with modern constraint optimization engines (handles thousands of constraints and training instances).
Solving the Optimization Problem

• Find w and b such that
  • $\phi(w) = \frac{1}{2} w^t w$ is minimized
  For all $\{(x_i, y_i)\}$: $y_i (w^T x_i + b) \geq 1$

The solution involves constructing a dual problem where a Lagrange multiplier $\alpha_i$ is associated with every constraint in the primary problem:

Find $\alpha_1, \ldots, \alpha_N$ such that

$Q(\alpha) = \sum \alpha_i - \frac{1}{2} \sum \sum \alpha_i \alpha_j y_i y_j x_i^T x_j$ is maximized and

$\sum \alpha_i y_i = 0$
(1) $\sum \alpha_i y_i = 0$
(2) $\alpha_i \geq 0$ for all $\alpha_i$
Dataset with noise

- denotes +1
- denotes -1

- Hard Margin: So far we require all data points be classified correctly
  - No training error
- What if the training set is noisy?
  - Solution 1: use very powerful kernels

Problem?

OVERFITTING!
Soft Margin Classification

- Slack variables can be added to allow misclassification of difficult or noisy data

What should be our quadratic optimization criterion be?

Minimize

$$\frac{1}{2} w^T w + C \sum_{K=1}^{R} \varepsilon$$
Hard vs. Soft Margin SVM

• Hard-margin doesn’t require to guess the cost parameter (requires no parameters at all)

• Soft-margin also **always** has a solution

• Soft –margin is more robust to outliers
  Smoother surfaces (in non –liner cases)
Algorithm 1 for training Classification SVMs via OP2.

1: Input: $S = ((x_1, y_1), \ldots, (x_n, y_n)), C, \epsilon$
2: $\mathcal{W} \leftarrow \emptyset$
3: repeat
4: \hspace{1em} $(w, \xi) \leftarrow \arg \min_{w, \xi \geq 0} \frac{1}{2} w^T w + C \xi$
   \hspace{1em} s.t. \forall c \in \mathcal{W}: $\frac{1}{n} w^T \sum_{i=1}^n c_i y_i x_i \geq \frac{1}{n} \sum_{i=1}^n c_i - \xi$
5: \hspace{1em} for $i=1,\ldots,n$ do
6: \hspace{2em} $c_i \leftarrow \begin{cases} 1 & y_i (w^T x_i) < 1 \\
0 & \text{otherwise} \end{cases}$
7: \hspace{1em} end for
8: \hspace{1em} $\mathcal{W} \leftarrow \mathcal{W} \cup \{c\}$
9: until $\frac{1}{n} \sum_{i=1}^n c_i - \frac{1}{n} \sum_{i=1}^n c_i y_i (w^T x_i) \leq \xi + \epsilon$
10: return $(w, \xi)$
SVM Applications

• SVM has been used successfully in many real world applications
  • Text (and hypertext) categorization
  • Image classification
  • Bioinformatics (protein classification, cancer classification)
• Hand-written char. classification