Learning Optimally Sparse Support Vector Machines

Andrew Cotter\textsuperscript{1}    Shai Shalev-Shwartz\textsuperscript{2}    Nathan Srebro\textsuperscript{1}

\textsuperscript{1}Toyota Technological Institute at Chicago
\textsuperscript{2}Hebrew University of Jerusalem

June 17, 2013
Sparse SVMs

Kernel SVMs

- Learn predictor of the form $w = \sum_{i=1}^{n} \alpha_i y_i \Phi(x_i)$
- Classify $x$ using sign of $\langle w, \Phi(x) \rangle = \sum_{i=1}^{n} \alpha_i y_i K(x_i, x)$
  - Computational cost is $\|\alpha\|_0$ kernel evaluations
  - Memory footprint is $\|\alpha\|_0$ training vectors

Goal: learn sparse classifiers (i.e. $\|\alpha\|_0$ small)

- For optimum of the SVM objective, every margin violation is a support vector
- For non-separable problems, a constant fraction of the training examples will be support vectors
Contributions

Our contributions:

- We show that it's always possible to have a support size of $O\left(\|w^*\|^2\right)$ without losing accuracy
  - This is tight (up to a constant)
- We provide an algorithm which achieves this efficiently
  - Finds a predictor with $\|\tilde{\alpha}\|_0 = O\left(\|w^*\|^2\right)$
  - No increase in $O\left(\text{runtime}\right)$
  - No increase in $O\left(\text{sample complexity}\right)$
  - Optimal sparsity
  - Sparsity is independent of desired accuracy $\varepsilon$ and training size $n$
Our Approach

We propose a multi-step procedure:

1. Train an SVM using a traditional solver, yielding a classifier \( w \)
2. Create a relaxed problem which:
   - Ignores examples misclassified by \( w \)
   - Is lenient on margin violations
3. Optimize this relaxed problem to yield a sparse approximation \( \tilde{w} \) to \( w \)
Our Approach

We propose a multi-step procedure:

1. Train an SVM using a traditional solver, yielding a classifier $w$

2. Create a relaxed problem which:
   - Ignores examples misclassified by $w$
   - Is lenient on margin violations

3. Optimize this relaxed problem to yield a sparse approximation $\tilde{w}$ to $w$

If we use the Stochastic Batch Perceptron for step (1), and find a $\tilde{w}$ s.t.:

$$\mathcal{L}_{0/1}(\tilde{w}) \leq \mathcal{L}_{\text{hinge}}(w^*) + \varepsilon$$

Then:

$$n = \tilde{O}\left(\left(\frac{L^*+\varepsilon}{\varepsilon}\right) \frac{\|w^*\|^2}{\varepsilon}\right) \quad \#\text{KEV} = \tilde{O}\left(\left(\frac{L^*+\varepsilon}{\varepsilon}\right)^3 \frac{\|w^*\|^4}{\varepsilon}\right) \quad \#\text{SV} = O\left(\|w^*\|^2\right)$$

(best known) (best known) (optimal)
Learning Optimally Sparse SVMs
Cotter, Shalev-Shwartz and Srebro

We present an algorithm for finding sparse SVM solutions

- First algorithm with sparsity and generalization guarantee
- Is extremely fast and simple
- Works well in practice

---

**IJCNN**

![IJCNN graph](image1)

**TIMIT**

![TIMIT graph](image2)

---

Support Size