Linear Sparsity in Machine Learning

Tues 14 March 2006

Tyler Neylon
Outline

• **Introduction**
  - Intuition for sparse linear classifiers

• **Previous Work**
  - Sparsity in support vector machines
  - Guarantees for $L^1$ minimization

• **New Work** *(note: not actually in these slides)*
  - Most matrices are unsparsifiable
  - A VC dim bound for sparse linear classifiers
  - Detecting matrix sparsifiability
  - A better heuristic for max sparsity?
Outline

• Introduction
  – Intuition for sparse linear classifiers
    • Sparsity as subset selection
    • $L^1$ minimization approximates $L^0$ min'n
Advantages of sparsity

- **Philosophically:**
  - Occam's Razor

- **More precisely:**
  - Compression
  - Faster computations (after learning)
  - Subset selection
  - Better learning
  - Fights gum disease*

*not really
Subset selection \(\rightarrow\) Learning one coordinate from a subset of the others
• **Example:** Stamp collectors
  - Each piece of data is a vector \( x \in \mathbb{R}^n \)
  - Each coordinate \( x_i \) represents how many stamps collector \( i \) has at that moment
  - Suppose collectors 5, 13, and 28 mainly trade stamps among themselves; then

\[
x_5 + x_{13} + x_{28} = C
\]
Learning on a subset of coord's

- Think of each training point as a row in a matrix $X$
- Each column corresponds to a single stamp collector
- Let $n = (0,0,0,0,1,0,...,0,1,0,...,0,1,0,-C)$, with 1's in position 5, 13, 28
- Then $Xn=0$ is our matrix equation
- Vector $n$ is sparse

$$x_5 + x_{13} + x_{28} = C$$
Learning on a subset of coord's

- If $X_n = 0$, and $n$ is not all zeros, then we can single out a particular column of $X$:
  \[
  \hat{X}_n = C - x_{13} - x_{28} = x_5
  \]

- We can think of it as learning coordinate 5 from a small subset of the others
  \[
  x_5 + x_{13} + x_{28} = C
  \]
Learning on a subset of coord's

- **Example**: Diamonds & Jewelers
  - Number of diamonds in our data may change
  - Out of many diamond dealers and jewelers, maybe competitive dealers A & B work primarily with jewelers C & D
  - Then, if $x_i$ is the revenue of person $i$,
    \[ x_A + x_B = \text{const}(x_C + x_D) \]

Again, we could rewrite this as $Xn=0$, or $Xn = x_A$ etc
Learning on a subset of coord's

- **Example**: Neurons firing
  - Samples taken from unknown locations in a live human brain
  - Neurons react to each other, but not simultaneously
  - Key idea: *take FFT of data first* to ignore phase shifts if there is an interdependent subset of interactive neurons
Learning on a subset of data

- **Example**: Support Vectors
  - Previous were regression, this is classification
  - Now the sparsity is in which training points we use to learn
  - Learn only from the “border” cases
  - Why only a few support vectors?
  - Is it good to have only a few?
The $L^0$ wannabe norm

• Reminder: $L^p$ norm

\[ \| x \|_p := \left( \sum_i |x_i|^p \right)^{1/p} \]

• It's a real norm when $p \geq 1$
• Not well-defined for $p=0$
• Take limit of sum as $p \to 0$
The $L^0$ “norm”

- So we define

$$\|x\|_0 := \#\{i \mid x_i \neq 0\}$$

- So that minimizing this maximizes sparsity

- How close to being a norm?
A hard optimization problem

\[
P_0 \begin{cases} 
\min & ||x||_0 \\
\text{s.t.} & Ax = b 
\end{cases}
\]

- Shown NP-hard by Larry Stockmeyer
- Essentially* shown NP-hard to approximate by Piotr Berman and Marek Karpinski in “Approximating Minimum Unsatisfiability in Linear Equations”

*but the authors don't point out the equivalence with this problem
An “easy” optimization problem

\[ P_1 \begin{cases} \min & \|x\|_1 \\ \text{s.t.} & Ax = b \end{cases} \]

- Can be solved with linear programming
- Heuristic: solution to \( P_1 \) often \textit{identical} to \( P_0 \) solution
- Whyzat?
Intuition: Inflating level sets

- Solving P1 is like blowing up an L1-balloon* until it hits the hyperplane $Ax=b$

- Sparsest points are most extreme on the balloon

*Contains pointy edges -- unsafe for children under 3.
The tetherball counterexample

• We're in R3
• The line \( \{ x | Ax = b \} \) goes through points
  – \( s = (0,0,3) \)
  – \( t = (1,1,0) \)
• Then \( ||s||_1 > ||t||_1 \), so \( P1 \neq P0 \) here
• Replace 3 by anything \( > 2^{(1/p)} \) to get a case in which \( Pp \neq P0 \) for any \( p > 0 \) (!)
Outline

- Previous Work
  - Sparsity in support vector machines
  - Guarantees for $L^1$ minimization
Reminder: The Classic Support Vector Machine

• This is the traditional classification version

\[
\begin{aligned}
\text{min} & \quad ||w||_2 \\
\text{s.t.} & \quad y_i[(x_i \cdot w) + b] \geq 1 \ \forall i
\end{aligned}
\]

• Will \( w \) be sparse here?
• Sometimes – depends on the data! (how?)
Classification Sparsity

- Some results due to Ingo Steinwart in “Sparseness of SVM's – some asymptotically sharp bounds”
- We use the alternative version:

\[
\min \lambda \|f^*\|_H^2 + \frac{1}{n} \sum_{i=1}^{n} |y_i(f^*(x_i) + b)|^p_{-\epsilon}
\]

where \( |z|_{-\epsilon} := \max(0, \epsilon - z) \)

If \( p=1 \), we call it L1-SVM; if \( p=2 \), we call it L2-SVM.
Classification Sparsity

- Let $R$ denote the Bayes error
- Let $S$ be the $\text{Prob}(\text{point } x \text{ may be noisy})$
- Then $S \geq 2R$
- Let $N_{svp} = \text{Number of support vectors from solving } Lp\text{-SVM}$

\[
\frac{N_{SV1}}{n} \rightarrow R
\]

and

\[
\frac{N_{SV2}}{n} \rightarrow S \quad \text{in probability}
\]
Reminder: The Classic Support Vector Machine

- This is the *regression* version

\[
\min \frac{1}{2}||w||^2 + C \sum_i |y_i - f^*(x_i)|_\varepsilon
\]

where

\[
f^*(x) = \sum_i w_i K(x, x_i) + b
\]

and

\[|z|_\varepsilon := \max(0, |z| - \varepsilon)\]
Another Version

• An alternative presented by Federico Girosi in “An equivalence between sparse approximation and SVM's"  
  \[
  \begin{aligned}
  \min & \quad \frac{1}{2}\|f(x) - f^*(x)\|_{\mathcal{H}}^2 + \epsilon\|w\|_1 \\
  \text{s.t.} & \quad <f^*, 1>_{\mathcal{H}} = 0
  \end{aligned}
  \]

  where \[f^*(x) = \sum_i w_i K(x, x_i)\]

• and \(f(x) = \) target function
An Equivalence

• This version is equivalent (same solution!) to the traditional regression SVM when:
  - Data is noiseless
  - $<f,1>_H = 0$
  - $<1,K(x,y)>_H = 1$ for any $y$

Awesome!
Why low $N_{sv}$ is good

- We've seen that many forms of SVM's give sparse solutions
- But does it improve our generalization error?
- Yes! Vladimir Vapnik* says:

$$E[error] \leq \frac{E[N_{sv}]}{n}$$

*In, for example, “Advances in Kernel Methods,” chapter 3, editors Bernhard Scholkopf, Christopher Burges, Alexander Smola
Another version of sparse regression

- Useful for finding/approximating an entire basis of a sparse null space:

\[
\begin{align*}
\min \quad & \|w\|_1 \\
\text{s.t.} \quad & \sum_i (y_i - w \cdot x_i)^2 < \epsilon 
\end{align*}
\]
Other methods of approximating sparsity

- MOF
- MP
- OMP
- BOB

...also known as...
Other methods of approximating sparsity

- Least squares
- Matching Pursuit
- Orthogonal matching pursuit
- Best Orthogonal Basis

Most people nowadays agree that L1-minimization is generally best, although other methods may be a little faster / better in other ways

A good exposition can be found in “Atomic Decomposition by Basis Pursuit” by Scott Chen, David Donoho, and Michael Saunders
Some L1-min guarantees

• In solving

\[
\begin{align*}
\begin{cases}
\min & \|x\|_0 \\
\text{s.t.} & Ax = b
\end{cases}
\end{align*}
\quad \text{vs.} \quad \begin{align*}
\begin{cases}
\min & \|x\|_1 \\
\text{s.t.} & Ax = b
\end{cases}
\end{align*}
\]

• For a fixed aspect ratio of a sequence of matrices \(A_n\), there exists a constant percentage \(p\) so that if there exists a solution \(x\) with density at most \(p\), then

\[
\text{Prob}(P0 = P1) \to 1 \quad \text{as} \quad n \to \infty
\]

Where the size of \(A_n\) increases without bound with \(n\)
Some L1-min guarantees

• That's a proof of David Donoho, in “For most large underdetermined systems of linear equations, the minimal L1-norm solution is also the sparsest solution”

• That paper is hard to read, by the way
Outline

- **New Work**
  - Most matrices are unsparsifiable
  - A VC dim bound for sparse linear classifiers
  - Detecting matrix sparsifiability
  - A better heuristic for max sparsity?

**Coming Soon!**

This is the last slide for now.

Thank you!