L1 vs. L2 Regularization and feature selection.

Paper by Andrew Ng (2004)

Presentation by Afshin Rostami

Main Topics

- Covering Numbers
  - Definition
  - Convergence Bounds
- L1 regularized logistic regression
  - L1 Regression Convergence Upper Bound
- Rotational Invariance
  - Rotational Invariance Convergence Lower Bound

Main Theorems

- L1 Regularized Logistic Regression requires a sample size that grows logarithmically in the number of irrelevant features.
- Rotationally invariant algorithms (i.e. L2 Regularized Logistic Regression) requires a sample size that grows linearly in the number of irrelevant features.

Covering Numbers

Definition:

$$\text{Let} \quad \vec{x} = (x_{1,} x_{2,} \ldots, x_m) \in X^m$$

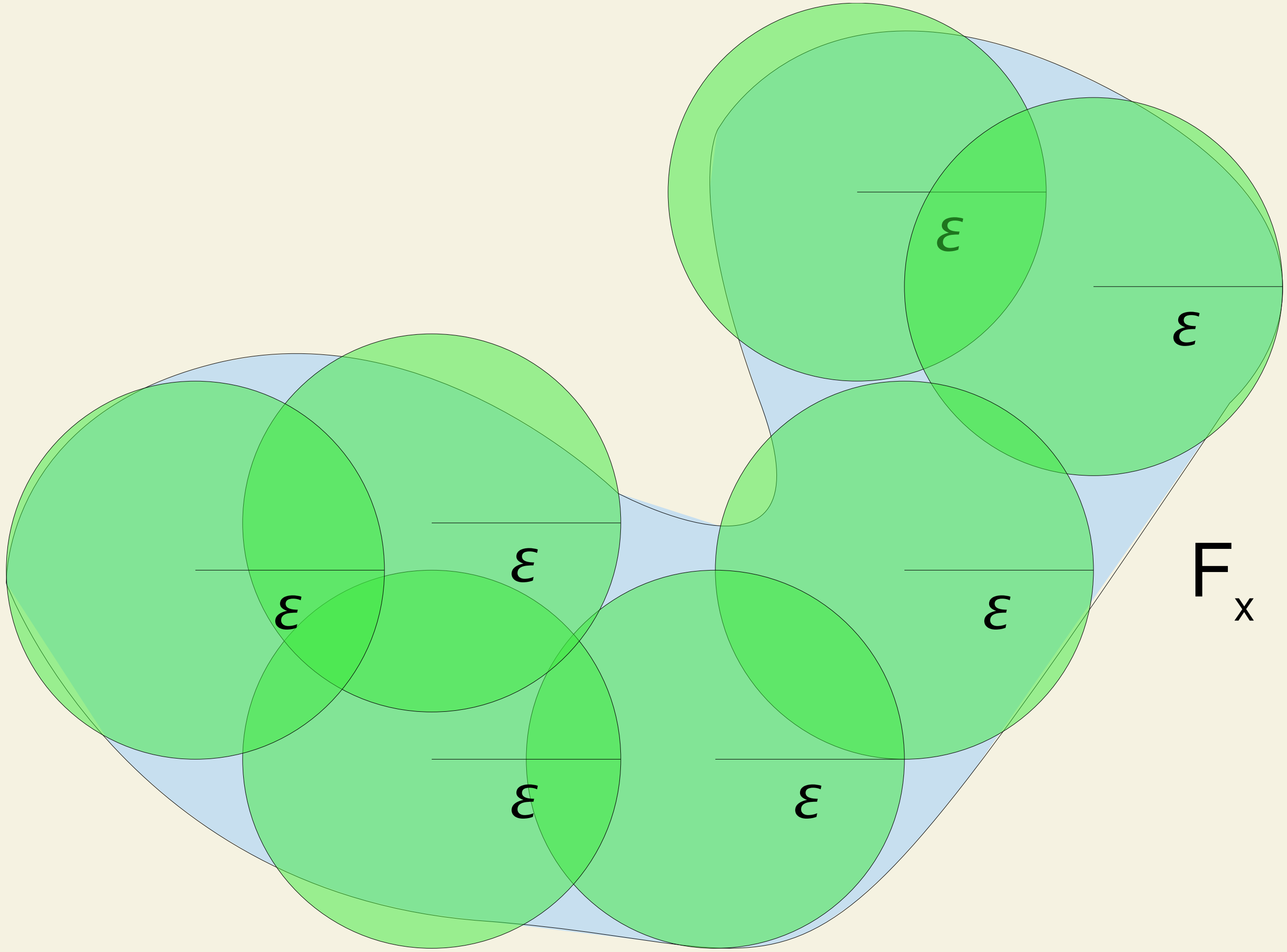
Then for a function class *F* with domain *U* and range

[-*M* , *M*], let

$$F_{\vec{x}} = \{ f(x_1), f(x_2), \ldots, f(x_m) \ : \ f \in F \}$$

Then the covering number $N_p(F, \epsilon, \vec{x})$ is defined as the smallest set of points, such that for any $u \in F_{\vec{x}}$ there is a point in the set no further than ε away as measured by the *p*-norm.

Covering Numbers



Covering Numbers

Then, define:

$$N_p(F, \epsilon, m) = \max_{\vec{x}} N_p(F, \epsilon, \vec{x})$$

Notice the similarity to the shattering coefficient. Both functions measure the number of different labeled sets of size m realizable by a certain function class. In particular, notice if some function class *H* takes values in {0,1} and positive ε < 0 then:

$$N_\infty(H, \epsilon, m) = \prod_H (m)$$

Convergence with Covering Numbers

Convergence Result by Pollard (1984):

For iid $\vec{x} = (x_{1,} x_{2,} \ldots, x_m)$

$$P\left[\exists f \in F : \left| \frac{1}{m} \sum_{i=1}^{m} f(x_i) - E_{x \sim D}[f(x)] \right| > \epsilon \right] \leq 8 E[N_1(F, \epsilon/8, \vec{x})] \exp\left(\frac{-m\epsilon^2}{512 M^2} \right)$$

Again, notice similarity to sample bound using the shattering coefficient. Proof is similar too, use symmetrization and analyze permutations.

Logistic Regression

When using logistic regression, we model the conditional probability that our label is 1 as:

$$p(y=1|x,\theta)=\frac{1}{1+\exp(-\theta^T x)}$$

Where θ are the parameters we wish to learn. Thus we wish to maximize the following

$$\arg\max_\theta \sum_{i=1}^{m} \log p(y_i|x_i,\theta)-\alpha R(\theta)$$

Regularization Term

In the last slide, R(θ) is the regularization term, which forces the parameters to be small (for α > 0). This paper wishes to compare the performance of L1 and L2 regularization.

L1: $$R(\theta)=\|\theta\|_1=\sum_{i=1}^{n}|\theta_i|$$

L2: $$R(\theta)=\|\theta\|_2^2=\sum_{i=1}^{n}\theta_i^2$$

An Equivalent Optimization

Another way to state the optimization problem (which our algorithm will actually use) is as the following:

$$\max_\theta \sum_{i=1}^{m} \log p(y_i | x_i, \theta)$$

$$\text{subject to:} \quad R(\theta) \leq B$$

Where for every α there exists a corresponding *B*. In fact notice that our previous optimization is the Lagrangian of this one.

Algorithm Implementation

- Scale data so that for all x, |x| < 1
- Split the data (S) into training (S_1) and hold-out (S_2) of size $(1 - \gamma)m$ and γm respectively.
- For B = 0, 1, 2, 4, . . . , C
  - Solve optimization problem and store solution θ_B
- From step 2, choose the θ_B with the lowest hold-out error on S_2.

  (Notice we are tuning the parameter α when searching through values for B, and in a sense doing Structural Risk Minimization).

Measuring Error

The upper bound proof will be based on the logloss error:

$$\epsilon^{l}(\theta)=E_{(x,y)\sim D}[-\log p(y|x,\theta)]]$$

$$\hat{\epsilon}^{l}_{S}(\theta)=\frac{1}{|S|}\sum_{(x,y)\in S}-\log p(y|x,\theta)$$

If we denote the usual 0,1 error as ε^{m} then it can be shown

$$\epsilon^{l}(\theta)\geq(\log 2)\epsilon^{m}(\theta)$$

So an upper bound on the logloss is also an upper bound on the 0,1 misclassification error. Similarly a lower bound on misclassification error is also a lower bound on the logloss.

L1 Logistic Regression Upper Bound

Theorem: Suppose we are in a situation where only r (which we assume to be much smaller than n) of our features are necessary to learn a good parameter vector θ^r. Then for θ produced by the previously mentioned algorithm and $K > |\theta_i|$, we have:

$$\epsilon^l(\theta) \leq \epsilon^l(\theta^r) + \epsilon$$

With sample complexity:

$$m = \Omega((\log n) \mathrm{poly}(r, K, \log(1/\delta), 1/\epsilon, C))$$

L1 Logistic Regression Upper Bound

Proof: The proof makes use of many supporting lemmas regarding covering numbers.

Lemma1:

If: $G = \{ g : g(x) = \theta^T x,\ x \in \mathbb{R}^n, \|\theta\|_q \leq a \} \quad \wedge \quad \forall x \ \ \|x\|_p \leq b$

$$\text{such that} \quad \frac{1}{p} + \frac{1}{q} = 1 \quad \wedge \quad 2 \leq p \leq \infty$$

Then: $\log_2 N_2(G, \epsilon, m) \leq \text{ceil}\left[\frac{a^2 b^2}{\epsilon^2}\right] \log_2(2\text{n}+1)$

A few more Lemmas

Proof cont: Two more useful Lemmas:

Lemma 2: For any parameters: $N_1 \leq N_2$

Lemma 3: For G as defined before and F defined as:

$$F = \{ f_g(x, y) = l\,(g(x), y) \,:\, g \in G \,, y \in \{0,1\} \}$$

Where the function l(. , y) for fixed y is Liphschitz with constant L, we have:

$$N_1(F, \epsilon, m) \leq N_1(G, \epsilon / L, m)$$

Upper Bound Proof

Proof cont: Let B' be the first values in {0,1,2,4,...} larger than rK, so $rK < B' < 2rK$. Then using Lemma 1 and 2, we have:

$$\log_2 N_1(G,\epsilon,m) \le \log_2 N_2(G,\epsilon,m)$$
$$\le \mathrm{ceil}\left[\frac{\|\theta\|_1\|x\|_\infty}{\epsilon^2}\right]\log_2(2\mathrm{n}+1)$$
$$= \mathrm{ceil}\left[\frac{B'^2}{\epsilon^2}\right]\log_2(2\mathrm{n}+1)$$

Upper Bound Proof

Proof cont: Notice, if we define l(g(x), y) as the logloss suffered by logistic regression, then we have l(. , y) for fixed y is Lipschitz with L = 1.

Thus, using Lemma 3 we have:

$$\log_2 N_1(F, \epsilon, m) \leq \text{ceil}\left[\frac{B'^2}{\epsilon_2}\right]\log_2(2\text{n}+1)$$

Now to bound M:

$$|f(x,y)| = |l(g(x),y)| = |l(\theta^T x, y)|$$
$$\leq |\theta^T x| + 1 \leq \|\theta\|_1 \|x\|_\infty + 1 \leq B' + 1$$

The first inequality is due to the fact that l(. , y) is defined as the logloss and the second inequality is due to Holder's Inequality (|ab| = $||a||_p$ $||b||_q$ for conj. p, q.

Putting it all Together

Now putting together all the steps of the proof, we have shown:

$$P\left[\exists f \in F : \left| \hat{\epsilon}^l(\theta) - \epsilon^l(\theta) \right| > \epsilon \right]$$

$$\leq 8 \cdot 2^{64 B'^2/\epsilon+1}\ (2\text{n}+1) \cdot \exp\left(\frac{-m\epsilon^2}{512(B'+1)^2}\right)$$

Thus, for confidence (1 - δ) we have *m* satisfy:

$$m = \Omega((logn) \cdot \text{poly}(r, K, \frac{1}{\epsilon}, \log\frac{1}{\delta}))$$

Then using standard convergence result (Vapnik, 1982) we have

$$\epsilon^l(\theta) \leq \max_{\theta' : \|\theta'\|_1 \leq \text{B}'} \epsilon^l(\theta') + 2\epsilon$$

$$\leq \epsilon^l(\theta^r) + 2\epsilon$$

Rotationally Invariant Algorithms

- M is a rotation matrix if $M^T M = MM^T = I$ and $|M| = 1$
- An algorithm, L, is Rotationally Invariant if for training set S and test x, $L_S(x) = L_{MS}(Mx)$
- Examples: Log Regression w/ L2 Reg., SVMs, Perceptron, Unregularized Log Regression, Neural Network w/ back-propagation, any algorithm that uses PCA preprocessing.

Rotationally Invariant Algorithm Lower Bound

For any $0 < \varepsilon < 1/8$ and $0 < \delta < 1/100$, if we have a rotational algorithm L, and a labeling that is determined by a single feature (i.e. $y = 1$ if $x_1 > t$, $y = 0$ otherwise), then in order for L to attain a 0/1 misclassification lower than ε with probability at least $(1 - \delta)$ it is necessary that the training size be at least:

$$m = \Omega(n/\epsilon)$$

Lower Bound Proof

Proof: Let C be the concept class of linear separators:

$$C = \{ h_\theta ; h_\theta(x) = 1\{ \theta^T x \geq \beta \}, \theta \neq 0 \}$$

Recall that such a concept class has VC dimension n + 1. Also, recall that the standard VC lower bound states there exists a distribution D_X that induces a sample complexity of Ω(d / ε), which, in the case of linear separators, can be simplified to Ω(n / ε).

Let $\tilde{h}(x) = 1\{ \tilde{\theta}^T x \geq \tilde{\beta} \}$ be the target concept and without loss of generality let $\|\tilde{\theta}\|_2 = 1$.

There exists an orthogonal matrix M whose first row is $\tilde{\theta}^T$.

Lower Bound Proof

Proof cont: Notice,

$$M\tilde{\theta} = [1,0,\ldots,0]^T = e_1$$

Also, if we need do, we can flip the sign of any row (other than the first) to ensure |M| = 1, so that M is a rotation matrix.

Now, examine the problem with datapoints $x = (x_1, x_2, \ldots, x_n) \sim D_X$ and let x' = Mx, and labels determined by the first feature $y' = 1\{x'_1 \geq \tilde{\beta}\}$.

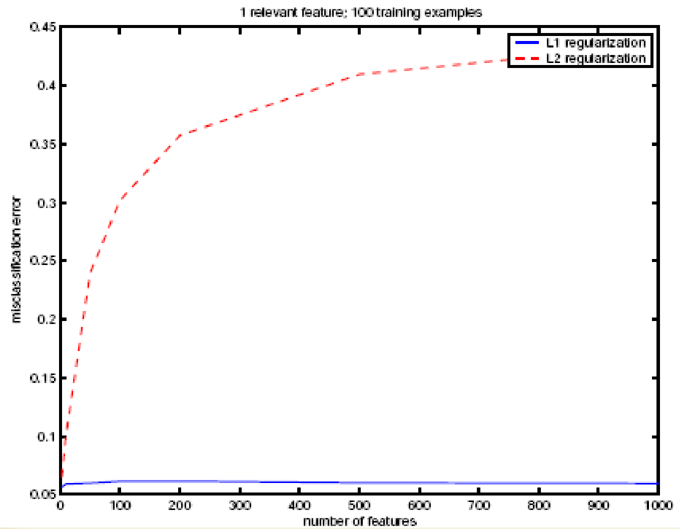
Lower Bound Proof

Proof cont:

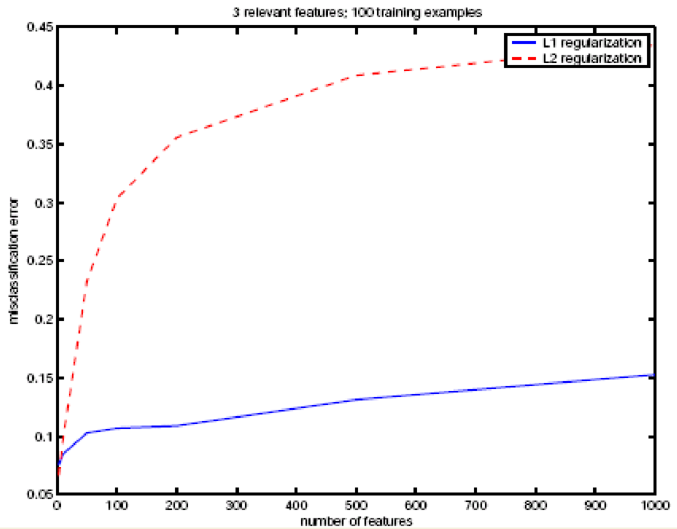
$$y'=1\{x'_1\geq\tilde{\beta}\}=1\{e_1^T x'\geq\tilde{\beta}\}$$
$$=1\{(M\tilde{\theta})^T(Mx)\geq\tilde{\beta}\}$$
$$=1\{\tilde{\theta}^T x\geq\tilde{\beta}\}=y$$

So the problem on (x',y') is actually a rotated version of (x,y), and, since we assume our learning algorithm is rotationally invariant, we expect it to perform the same on both problems. In other words, we are trying to learn a linear separation in n dimensional space giving us a sample complexity lower bound of . $m=\Omega(n/\epsilon)$

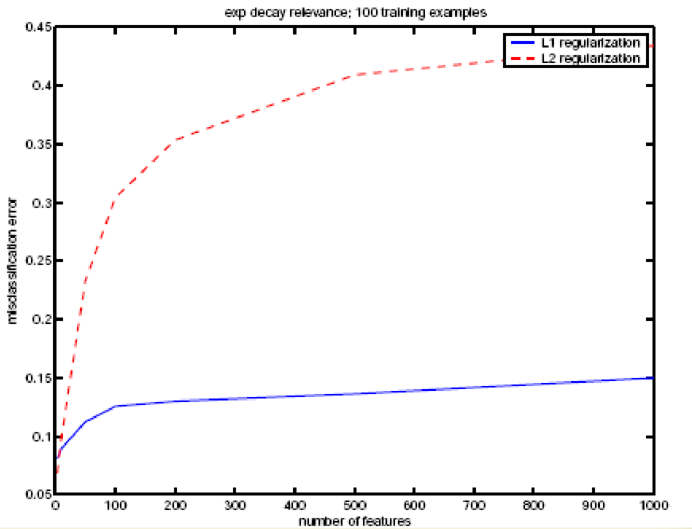
Experiments



1 relevant feature, 100 training examples



3 relevant feature, 100 training examples



exp decay in relevance, 100 training examples