Shrink-stretch of labels for regularizing logistic regression

Manfred K. Warmuth

University of California - Santa Cruz

February 7, 2008
Developed with lots of help Dima Kuzmin
Recent experiments by D. Sculley
Setup

\[ \hat{y} = f(\hat{a}) \]
probability estimate

\[ \hat{a} = w \cdot x \]
linear activation

w weight vector

x feature vector
Purpose of transfer function

- Transfers linear activation $\rightarrow$ probability

- Logistic regression:
  uses probability estimates $\hat{y} = h(\hat{a}) = \frac{\exp(\hat{a})}{1+\exp(\hat{a})}$

- Will use other non-decreasing functions

  $$h : \mathbb{R} \rightarrow [0, 1]$$
Data?

Examples \((x_t, y_t)\)

- Feature vectors \(x_t\) typically binary
- Label \(y_t\) “true” probability
  Typically binary
Logistic loss

Estimate: \( \hat{y} = h(\mathbf{w} \cdot \mathbf{x}) = \frac{\exp(\mathbf{w} \cdot \mathbf{x})}{1 + \exp(\mathbf{w} \cdot \mathbf{x})} \)

Loss: \( \text{loss}(y, \hat{y}) = y \ln \frac{y}{\hat{y}} + (1 - y) \ln \frac{1 - y}{1 - \hat{y}} \)

\( y \in \{0, 1\} \)

\( \forall \)

\[
\begin{align*}
\text{if } y = 0 & : -\ln(1 - \hat{y}) = \ln(1 + \exp(\mathbf{w} \cdot \mathbf{x})) \\
\text{if } y = 1 & : -\ln \hat{y} = \ln(1 + \exp(\mathbf{w} \cdot \mathbf{x})) - \mathbf{w} \cdot \mathbf{x}
\end{align*}
\]

= negative log likelihood
Crucial property

\[ \frac{\partial}{\partial \mathbf{w}} \text{loss}(y, h(\mathbf{w} \cdot \mathbf{x})) = \left( h(\mathbf{w} \cdot \mathbf{x}) - y \right) \mathbf{x} \]

delta rule

Derivatives for sum of examples = 0

\[ \sum_t \hat{y}_t x_{t,i} = \sum_t y_t x_{t,i} \quad \text{for all features } i \]

est. prob of 1 when i on

true prob of 1 when i on
Outline

3 Overfitting

4 Regularization
### Danger of enforcing constraints

**One feature - inseparable**

<table>
<thead>
<tr>
<th>$y_t$</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_{t,1}$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
</tbody>
</table>

**Add one sparse feature - separable**

<table>
<thead>
<tr>
<th>$y_t$</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_{t,1}$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>$x_{t,2}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Adding sparse feature

Top row - Logic regression on original data set
Bottom row - ditto after feature was added
Danger of enforcing the derivative equations

\[ \forall \text{ features } i : \]

\[
\sum_{t=1}^{T} \hat{y}_t x_{t,i} = \frac{\sum_{t=1}^{T} y_t x_{t,i}}{T} \]

\[ x_{t,i} \text{ binary} \quad \frac{\sum_{t: x_{t,i} = 1} \hat{y}_t}{T} = \frac{\sum_{t: x_{t,i} = 1} y_t}{T} \]

- As lhs goes to 0 or 1, some weights of \( w \) must get unbounded
- In practice there is always must be some sort of early stopping when just the total logistic loss is minimized
Adding sparse features

\[
\begin{array}{cccc}
0 & 1 & 0 & 0 & 1 \\
1 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 \\
1 & 1 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 1 \\
\Rightarrow \\
0 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 \\
\end{array}
\]

Trivial to find sol if each example has private feature
Classical case of overfitting
Our data set: 1.6K features, 150K examples
(Typical datasets: $10^9$ features, $10^{12}$ examples)
data with sparse features added (2)

Sparse noisy features improve training performance but degrade test performance
Optimization versus machine learning

Not enough to minimize a convex loss function
In machine learning minimize

\[ \text{convex regularization} + \eta \text{ convex loss} \]

Goal: improve test performance
Standard Fix: throw more data at it

```
0 1 0 0 1 1 0 1 1
```

more examples always help

more features

danger of overfitting

Sparse feature easily overfit
Outline

3 Overfitting

4 Regularization
Regularization needed

Weights need to be “controlled”
- Early stopping of training algorithm
  - No large weights in logistic regression because $\text{sigmoid}(10) = 1$
- Clip the weights
- Regularize with $\sum_i w_i^2$
- Regularize with $\sum_i |w_i|$ or relative entropies
- Feature selection
- New trick: clip range of labels $y_t$
A simple definition of regularization

"Any tunable method that increases the average loss on the training set but decreases the average loss on the test set"
If you run a gradient descent algorithm for minimizing the total logistic loss, then stopping the algorithm early amounts to an implicit regularization (because initially the weights are small).

Clean methodology

- Start with a convex minimization problem
- Report the precision to which you trained
- This deemphasizes the algorithm that was used for minimization
Examples

\[
\inf_w \sum_{t=1}^T \frac{\text{loss}(y_t, \sigma(w \cdot x_t))}{T}
\]

or

\[
\inf_w \sum_{t=1}^T \left( \frac{1}{2\eta} \|w\|_2^2 + \text{loss}(y_t, \sigma(w \cdot x_t)) \right) \frac{1}{T}
\]

- Train until one-norm of gradient \( \leq 10^{-4} \)
- Check that decreasing the above to \( \leq 10^{-6} \) has no qualitative effect
Ideally train until

For the logistic loss:

\[ \left\| \frac{1}{T} \sum_t (\sigma(w \cdot x_t) - y_t)x_t \right\|_1 \leq 10^{-4} \]

or

\[ \left\| \frac{1}{\eta} w + \frac{1}{T} \sum_t (\sigma(w \cdot x_t) - y_t)x_t \right\|_1 \leq 10^{-4} \]

- Is there a better criterium?
- Want to be close to the minimum without computing it
Canonical hard example for $\sum_i w_i^2$

Random $n \times n$ matrix

$$
\begin{bmatrix}
0 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 & 1 & 1 \\
1 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 1 & 1 & 0
\end{bmatrix}
$$

Target is one of the rows

Any alg. that predicts with linear combination of instances has error half after seeing half of the examples

- Lower bound hold for linear (and logistic ?) regression based on gradient descent or $\sum_i w_i^2$ regularization
- Solution is unit vector that picks out the right row
- Additional features/kernels don’t help
- Easy to learn with 1-norm (or entropic regularization)
### 1 versus 2 norm - $n = 256$

|       | $\sum_i w_i^2$ | $\sum_i |w_i|$ |
|-------|----------------|----------------|
| **lin. regr.** | ![Graph](image1.png) | ![Graph](image2.png) |
| **log. regr.** | ![Graph](image3.png) | ![Graph](image4.png) |
1 versus 2 norm

1 norm solution sparser

2 norm solution lots of small weights on random features

1 norm solution slightly smaller loss on test set
Regularization by clipping labels

Top row: logistic regression w. single feature
Middle row: logistic regression w. added feature
Bottom row: logistic regression w. added feature and clipped labels

<table>
<thead>
<tr>
<th>$y_t$</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_{t,1}$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>$x_{t,2}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

$y = 1 \rightarrow y = \text{high value}$
$y = 0 \rightarrow y = \text{low value}$

Clipping ameliorates the negative effect of sparse features
Better way: clip average features in derivative equations
Clipping the data with added sparse features

Added \approx \text{third more features with 5 ones each}

Training based on clipped labels
- rescaled predictions achieve smaller logistic loss on test set
Clipping is ”new” method for preventing overfitting with sparse features
More details: Shrinking

- Data: \((x_t, y_t)\), where \(y_t \in [0, 1]\)
- Linear transformation of interval \([0,1]\) to subinterval \([a, b]\)

\[
y_t' := a + (b - a) y_t
\]

i.e. \(0 \rightarrow a\) and \(1 \rightarrow b\)
- Minimize logistic loss on training data \((x_t, y_t')\)
Logistic loss when label $1 - > b$

> \text{\texttt{sigmoid}}: =x \rightarrow \frac{e^x}{1 + e^x}

> \text{\texttt{loss}}:  (y, ah) \rightarrow y \ln (y / \text{\texttt{sigmoid}}(ah)) + (1 - y) \ln (1 - y / (1 - \text{\texttt{sigmoid}}(ah))) ;

> \text{\texttt{plot}} ([\text{\texttt{sigmoid}}(ah), .99999, \text{\texttt{loss}}(.99999, ah), .95, \text{\texttt{loss}}(.95, ah), .6, \text{\texttt{loss}}(.6, ah), .4, ...
More details: Stretching the prediction

- **A)** Predict on instance \( \mathbf{x} \) with \( \hat{y} = \sigma(\mathbf{w} \cdot \mathbf{x}) \)
  where \( \mathbf{w} \) is weight vector produced in training

- **B)** Predict with

\[
\hat{y} = \begin{cases} 
  1 & \text{if } \sigma(\mathbf{w} \cdot \mathbf{x}) > b \\
  0 & \text{if } \sigma(\mathbf{w} \cdot \mathbf{x}) < a \\
  \frac{\sigma(\mathbf{w} \cdot \mathbf{x}) - a}{b - a} & \text{otherwise}
\end{cases}
\]

i.e. \( \sigma(\mathbf{w} \cdot \mathbf{x}) \leq a \) mapped to 0 and \( \sigma(\mathbf{w} \cdot \mathbf{x}) \geq b \) mapped to 1

- When there are lots of sparse features then
  - B) prevents overfitting: worse average logistic loss on training set
  - better average logistic loss on test set
  - A) does not prevent overfitting
Questions

- Shrink/stretch hypothesis is linear combination of examples
- Unfair to compare to 1-norm regularization
- Does it beat 2-norm regularization (without early stopping)
- Is there a smarter way to change the label based on special knowledge about the examples and features
- Is there a dual version for minimizing logistic loss. In that case we have one variable per feature/constraint.

Idea: Only change those constraints for which
\[ \sum_{t: x_t, i=1} y_t \]
\[ \frac{1}{T} \] close to 0 or 1
Data from blog comment abuse (spam) detection, from 5 blog topics.

- Each topic had $\sim 10,000$ examples
  Divided into 50% train / 50% test set Roughly 20% positive (spam), 80% negative (not spam).
- Sparse, high dimensional data: $\sim 30,000$ total features, $\sim 100$ non-zero features per example.
- Some topics very noisy (e.g., news), others less noisy (e.g., cricket).
Experiments: Methods

- Used label shrinking, stretching
- Compared with 2-Norm regularization
- Used mean logistic loss as evaluation measure.
- Experimented with stochastic gradient descent (extremely slow), full-batch gradient descent (very slow), and chunk-batch gradient descent (just slow) for label shrinking, stretching and initial 2-Norm regularization.
- Also used iterative reweighted least squares solver with conjugate gradient descent (fast) for 2-Norm regularization.
Label Shrinking Results (no stretching)

Log. Reg. w/Label Shrinking: News data set

- Mean Logistic Loss (un-stretched)
- Shrink-stretch
- UCSC 33 / 39
2-Norm Regularization Results

Log. Reg. w/2-Norm Regularization: News data set

Mean Logistic Loss

Regularization Parameter: lambda

M. Warmuth (UCSC)
Experiments: Results

<table>
<thead>
<tr>
<th>topic</th>
<th>un-modified</th>
<th>label shrinking</th>
<th>shrink-stretch</th>
<th>2-norm regularize</th>
</tr>
</thead>
<tbody>
<tr>
<td>cricket</td>
<td>0.44</td>
<td>0.33</td>
<td>0.43</td>
<td>0.18</td>
</tr>
<tr>
<td>money</td>
<td>0.40</td>
<td>0.25</td>
<td>0.40</td>
<td>0.18</td>
</tr>
<tr>
<td>movies</td>
<td>0.57</td>
<td>0.45</td>
<td>0.57</td>
<td>0.20</td>
</tr>
<tr>
<td>news</td>
<td>0.89</td>
<td>0.44</td>
<td>0.66</td>
<td>0.34</td>
</tr>
<tr>
<td>sports</td>
<td>1.39</td>
<td>0.51</td>
<td>1.11</td>
<td>0.51</td>
</tr>
</tbody>
</table>

Mean logistic loss results w. best parameter settings
Experiments: Results

- 2-norm regularization appears to be clear winner
- Label Shrinking always helps (versus not shrinking) Label Shrinking/stretching not helpful on this data set
- Is the comparison between 2-Norm Regularization and Label Shrinking fair?
  - Yes: Both predict with linear combination of examples
  - ????: What about implicit regularization via early stopping
Experiments: Results

<table>
<thead>
<tr>
<th>1-norm of gradient</th>
<th>label shrinking</th>
<th>2-norm regularization</th>
</tr>
</thead>
<tbody>
<tr>
<td>cricket</td>
<td>0.0060</td>
<td>0.50</td>
</tr>
<tr>
<td>money</td>
<td>0.0004</td>
<td>0.80</td>
</tr>
<tr>
<td>movies</td>
<td>0.0020</td>
<td>0.87</td>
</tr>
<tr>
<td>news</td>
<td>0.0003</td>
<td>1.76</td>
</tr>
<tr>
<td>sports</td>
<td>0.0002</td>
<td>0.81</td>
</tr>
</tbody>
</table>

Label shrinking:

$$\nabla_w = \frac{1}{T} \sum_t (\sigma(w \cdot x_t) - y'_t)x_t,$$

where $y_t$ are shrunk labels

2-norm regularization:

$$\nabla_w = \frac{1}{\eta}w + \frac{1}{T} \sum_t (\sigma(w \cdot x_t) - y_t)x_t$$
The label shrinking results were gained with chunk-batch gradient descent, using 1-Norm of Gradient as a stopping criteria. The 2-norm regularization results were gained with an IRLS solver, using a different stopping criteria, and benefitted from early stopping.
Regularization Effect of Early Stopping
to 1-Norm of Gradient

Vanilla Logistic regression, no regularization, original labels