CS395T: Structured Models for NLP
Lecture 3: Multiclass Classification

Greg Durrett

Some slides adapted from Vivek Srikumar, University of Utah
Administrivia

- Course enrollment

- Project 1 out next Tuesday
Recall: Binary Classification

- Logistic regression: \( P(y = 1|x) = \frac{\exp \left( \sum_{i=1}^{n} w_i x_i \right)}{1 + \exp \left( \sum_{i=1}^{n} w_i x_i \right)} \)

  Decision rule: \( P(y = 1|x) \geq 0.5 \Leftrightarrow w^\top x \geq 0 \)

  Gradient (unregularized): \( x(y - P(y = 1|x)) \)

- SVM: quadratic program to minimize weight vector norm w/slack

  Decision rule: \( w^\top x \geq 0 \)

  (Sub)gradient (unregularized): 0 if correct with margin of 1, else \( x(2y - 1) \)
Loss Functions

- Hinge (SVM)
- 0-1 (ideal)
- Logistic
- Perceptron

\[ w^T x \]
Sentiment Analysis

- Classify sentence as positive or negative sentiment

  - *this movie was* great!* would* watch again
    - Positive
  
  - *the movie was* gross and overwrought*, but I* liked it
    - Negative
  
  - *this movie was not really very* enjoyable

- Bag-of-words doesn’t seem sufficient (discourse structure, negation)

- There are some ways around this: extract bigram feature for “not X” for all X following the *not*

Bo Pang, Lillian Lee, Shivakumar Vaithyanathan (2002)
**Sentiment Analysis**

Simple feature sets can do pretty well!

<table>
<thead>
<tr>
<th>Features</th>
<th># of features</th>
<th>frequency or presence?</th>
<th>NB</th>
<th>ME</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) unigrams</td>
<td>16165</td>
<td>freq.</td>
<td>78.7</td>
<td>N/A</td>
<td>72.8</td>
</tr>
<tr>
<td>(2) unigrams</td>
<td></td>
<td>pres.</td>
<td>81.0</td>
<td>80.4</td>
<td>82.9</td>
</tr>
<tr>
<td>(3) unigrams+bigrams</td>
<td>32330</td>
<td>pres.</td>
<td>80.6</td>
<td>80.8</td>
<td>82.7</td>
</tr>
<tr>
<td>(4) bigrams</td>
<td>16165</td>
<td>pres.</td>
<td>77.3</td>
<td>77.4</td>
<td>77.1</td>
</tr>
<tr>
<td>(5) unigrams+POS</td>
<td>16695</td>
<td>pres.</td>
<td>81.5</td>
<td>80.4</td>
<td>81.9</td>
</tr>
<tr>
<td>(6) adjectives</td>
<td>2633</td>
<td>pres.</td>
<td>77.0</td>
<td>77.7</td>
<td>75.1</td>
</tr>
<tr>
<td>(7) top 2633 unigrams</td>
<td>2633</td>
<td>pres.</td>
<td>80.3</td>
<td>81.0</td>
<td>81.4</td>
</tr>
<tr>
<td>(8) unigrams+position</td>
<td>22430</td>
<td>pres.</td>
<td>81.0</td>
<td>80.1</td>
<td>81.6</td>
</tr>
</tbody>
</table>

Bo Pang, Lillian Lee, Shivakumar Vaithyanathan (2002)
## Sentiment Analysis

Wang and Manning (2012)

<table>
<thead>
<tr>
<th>Method</th>
<th>RT-s</th>
<th>MPQA</th>
<th>CR</th>
<th>Subj.</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNB-uni</td>
<td>77.9</td>
<td>85.3</td>
<td>79.8</td>
<td>92.6</td>
</tr>
<tr>
<td>MNB-bi</td>
<td><strong>79.0</strong></td>
<td><strong>86.3</strong></td>
<td>80.0</td>
<td>93.6</td>
</tr>
<tr>
<td>SVM-uni</td>
<td>76.2</td>
<td>86.1</td>
<td>79.0</td>
<td>90.8</td>
</tr>
<tr>
<td>SVM-bi</td>
<td>77.7</td>
<td><strong>86.7</strong></td>
<td>80.8</td>
<td>91.7</td>
</tr>
<tr>
<td>NBSVM-uni</td>
<td><strong>78.1</strong></td>
<td>85.3</td>
<td>80.5</td>
<td>92.4</td>
</tr>
<tr>
<td>NBSVM-bi</td>
<td><strong>79.4</strong></td>
<td><strong>86.3</strong></td>
<td><strong>81.8</strong></td>
<td><strong>93.2</strong></td>
</tr>
<tr>
<td>RAE</td>
<td>76.8</td>
<td>85.7</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>RAE-pretrain</td>
<td>77.7</td>
<td><strong>86.4</strong></td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Voting-w/Rev.</td>
<td>63.1</td>
<td>81.7</td>
<td>74.2</td>
<td>–</td>
</tr>
<tr>
<td>Rule</td>
<td>62.9</td>
<td>81.8</td>
<td>74.3</td>
<td>–</td>
</tr>
<tr>
<td>BoF-noDic.</td>
<td>75.7</td>
<td>81.8</td>
<td>79.3</td>
<td>–</td>
</tr>
<tr>
<td>BoF-w/Rev.</td>
<td>76.4</td>
<td>84.1</td>
<td><strong>81.4</strong></td>
<td>–</td>
</tr>
<tr>
<td>Tree-CRF</td>
<td>77.3</td>
<td>86.1</td>
<td><strong>81.4</strong></td>
<td>–</td>
</tr>
<tr>
<td>BoWSVM</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>90.0</td>
</tr>
</tbody>
</table>

Naive Bayes is doing well!

Ng and Jordan (2002) — NB can be better for small data

Before neural nets had taken off — results weren’t that great

Two years later Kim (2014) with neural networks
This Lecture

- Multiclass fundamentals
- Feature extraction
- Multiclass logistic regression
- Multiclass SVM
- Optimization
A Cancer Conundrum: Too Many Drug Trials, Too Few Patients

Breakthroughs in immunotherapy and a rush to develop profitable new treatments have brought a crush of clinical trials scrambling for patients.

By GINA KOLATA

Yankees and Mets Are on Opposite Tracks This Subway Series

As they meet for a four-game series, the Yankees are playing for a postseason spot, and the most the Mets can hope for is to play spoiler.

By FILIP BONDY

▶ 20 classes

Text Classification

Health

Sports
Image Classification

- Dog
- Car

- Thousands of classes (ImageNet)
Although he originally won the event, the United States Anti-Doping Agency announced in August 2012 that they had disqualified **Armstrong** from his seven consecutive Tour de France wins from 1999–2005.

---

- **4,500,000 classes (all articles in Wikipedia)**
One day, James thought he would go into town and see what kind of trouble he could get into. He went to the grocery store and pulled all the pudding off the shelves and ate two jars. Then he walked to the fast food restaurant and ordered 15 bags of fries. He didn't pay, and instead headed home.

3) Where did James go after he went to the grocery store?
   A) his deck
   B) his freezer
   C) a fast food restaurant
   D) his room

After about a month, and after getting into lots of trouble, James finally made up his mind to be a better turtle.

- Multiple choice questions, 4 classes (but classes change per example)
Binary classification: one weight vector defines positive and negative classes.
Can we just use binary classifiers here?
Multiclass Classification

- One-vs-all: train $k$ classifiers, one to distinguish each class from all the rest
- How do we reconcile multiple positive predictions? Highest score?
- Not all classes may even be separable using this approach.
Multiclass Classification

- All-vs-all: train \(\frac{n(n-1)}{2}\) classifiers to differentiate each pair of classes
- Again, how to reconcile?
Multiclass Classification

- Binary classification: one weight vector defines both classes
- Multiclass classification: one weight vector per class, decision is argmax
Multiclass Classification

- Formally: instead of two labels, we have an output space $\mathcal{Y}$ containing a number of possible classes
- Same machinery that we’ll use later for exponentially large output spaces, including sequences and trees

- Decision rule: $\arg\max_{y \in \mathcal{Y}} w^\top f(x, y)$
  - Multiple feature vectors, one weight vector
  - Can also have one weight vector per class: $\arg\max_{y \in \mathcal{Y}} w_y f(x)$
  - Why do we do with separate feature vectors? Let’s see!
Decision rule: \( \arg\max_{y \in Y} w^\top f(x, y) \)

*too many drug trials, too few patients*

Base feature function:

\[
f(x) = \mathbb{I}[\text{contains drug}], \mathbb{I}[\text{contains patients}], \mathbb{I}[\text{contains baseball}] = [1, 1, 0]
\]

Feature vector blocks for each label:

\[
f(x, y = \text{Health}) = [1, 1, 0, 0, 0, 0, 0, 0]
\]

\[
f(x, y = \text{Sports}) = [0, 0, 0, 1, 1, 0, 0, 0]
\]

Equivalent to having three weight vectors, but this formulation is more general if the features depend on \(y\)
Making Decisions

**too many drug trials, too few patients**

\[ f(x) = \mathbb{I}[\text{contains drug}], \mathbb{I}[\text{contains patients}], \mathbb{I}[\text{contains baseball}] \]

\[
\begin{align*}
  f(x, y = \text{Health}) &= [1, 1, 0, 0, 0, 0, 0, 0, 0] \\
  f(x, y = \text{Sports}) &= [0, 0, 0, 1, 1, 0, 0, 0, 0]
\end{align*}
\]

\[
w = [+2.1, +2.3, -5, -2.1, -3.8, +5.2, +1.1, -1.7, -1.3]
\]

\[
w^\top f(x, y) = \\
\begin{align*}
  \text{Health} &\quad +4.4 \\
  \text{Sports} &\quad -5.9 \\
  \text{Science} &\quad -1.9
\end{align*}
\]

"word drug in Science article" = +1.1
Multiclass Logistic Regression

\[ P(y|x) = \frac{\exp \left( w^T f(x, y) \right)}{\sum_{y' \in Y} \exp \left( w^T f(x, y') \right)} \]

sum over output space to normalize

- Compare to binary:

\[
P(y = 1|x) = \frac{\exp(w^T f(x))}{1 + \exp(w^T f(x))}
\]

negative class implicitly had \( f(x, y = 0) = \) the zero vector

- Training: maximize \( \mathcal{L}(x, y) = \sum_{j=1}^{n} \log P(y_j^*|x_j) \)

\[
= \sum_{j=1}^{n} \left( w^T f(x_j, y_j^*) - \log \sum_y \exp(w^T f(x_j, y)) \right)
\]
Training

- Multiclass logistic regression
  \[ P(y|x) = \frac{\exp \left( w^\top f(x, y) \right)}{\sum_{y' \in Y} \exp \left( w^\top f(x, y') \right)} \]

- Likelihood
  \[ \mathcal{L}(x_j, y_{j^*}) = w^\top f(x_j, y_{j^*}) - \log \sum_y \exp(w^\top f(x_j, y)) \]

\[
\frac{\partial}{\partial w_i} \mathcal{L}(x_j, y_{j^*}) = f_i(x_j, y_{j^*}) - \frac{\sum_y f_i(x_j, y) \exp(w^\top f(x_j, y))}{\sum_y \exp(w^\top f(x_j, y))}
\]

\[
\frac{\partial}{\partial w_i} \mathcal{L}(x_j, y_{j^*}) = f_i(x_j, y_{j^*}) - \sum_y f_i(x_j, y) P(y|x_j)
\]

\[
\frac{\partial}{\partial w_i} \mathcal{L}(x_j, y_{j^*}) = f_i(x_j, y_{j^*}) - \mathbb{E}_y[f_i(x_j, y)]
\]

model’s expectation of gold feature value from
Logistic Regression: Summary

- Model: 
  \[ P(y|x) = \frac{\exp \left( w^\top f(x, y) \right)}{\sum_{y' \in Y} \exp \left( w^\top f(x, y') \right)} \]

- Inference: 
  \[ \arg\max_y P(y|x) \]

- Learning: gradient ascent on the discriminative log-likelihood
  \[ \frac{\partial}{\partial w_i} \mathcal{L}(x_j, y_j^*) = f_i(x_j, y_j^*) - \mathbb{E}_y[f_i(x_j, y)] \]
  “towards gold feature value, away from expectation of feature value”
Training

- Are all decisions equally costly?

  - too many drug trials, too few patients

    Predicted Sports: bad error
    Predicted Science: not so bad

- We can define a loss function $\ell(y, y^*)$

  $\ell(\text{Sports}, \text{Health}) = 3$
  $\ell(\text{Science}, \text{Health}) = 1$
Multiclass SVM

Minimize $\lambda \|w\|^2_2 + \sum_{j=1}^{m} \xi_j$ 

s.t. $\forall j \xi_j \geq 0$

$$\forall j \ (2y_j - 1)(w^\top x_j) \geq 1 - \xi_j$$

$$\forall j \forall y \in Y \ w^\top f(x_j, y_j^*) \geq w^\top f(x_j, y) + \ell(y, y_j^*) - \xi_j$$

Correct prediction now has to beat every other class

Score comparison is more explicit now

The 1 that was here is replaced by a loss function

slack variables $> 0$ iff example is support vector
Multiclass SVM

Minimize $\lambda \|w\|^2 + \sum_{j=1}^{m} \xi_j$

s.t. $\forall j \xi_j \geq 0$

$\forall j \forall y \in Y \; w^\top f(x_j, y^*_j) \geq w^\top f(x_j, y) + \ell(y, y^*_j) - \xi_j$

- How does this quantification come into play?

- One slack variable per example, so it’s set to be whatever the most violated constraint is for that example

$$\xi_j = \max_{y \in Y} \left( w^\top f(x_j, y) + \ell(y, y^*_j) \right) - w^\top f(x_j, y^*_j)$$

- Plug in the gold $y$ and you get 0, so slack is always nonnegative!
Loss-Augmented Decoding

\[
\xi_j = \max_{y \in \mathcal{Y}} \left[ w^\top f(x_j, y) + \ell(y, y_j^*) \right] - w^\top f(x_j, y_j^*)
\]

too many drug trials, too few patients

<table>
<thead>
<tr>
<th></th>
<th>Health</th>
<th>2.4</th>
<th>0</th>
<th>2.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Health</td>
<td>+2.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sports</td>
<td>+1.3</td>
<td>3</td>
<td>4.3</td>
<td>← argmax</td>
</tr>
<tr>
<td>Science</td>
<td>+1.8</td>
<td>1</td>
<td>2.8</td>
<td></td>
</tr>
</tbody>
</table>

- **Sports** is most violated constraint, slack = 4.3 − 2.4 = 1.9
- Perceptron would make no update, regular SVM would pick **Science**
Computing the Subgradient

Minimize \( \lambda \|w\|^2 + \sum_{j=1}^{m} \xi_j \)

s.t. \( \forall j \quad \xi_j \geq 0 \)
\( \forall j \forall y \in \mathcal{Y} \quad w^T f(x_j, y) \geq w^T f(x_j, y^*) + \ell(y, y^*) - \xi_j \)

- If \( \xi_j = 0 \), the example is not a support vector, gradient is zero
- Otherwise, \( \xi_j = \max_{y \in \mathcal{Y}} w^T f(x_j, y) + \ell(y, y^*) - w^T f(x_j, y^*) \)
  \( \frac{\partial}{\partial w_i} \xi_j = f_i(x_j, y_{\text{max}}) - f_i(x_j, y^*) \)
  *(update looks backwards — we’re minimizing here!)*
- Perceptron-like, but we update away from *loss-augmented* prediction
Can we include a loss function in logistic regression?

\[ P(y|x) = \frac{\exp\left( w^\top f(x, y) + \ell(y, y^*) \right)}{\sum_{y'} \exp\left( w^\top f(x, y') + \ell(y', y_j^*) \right)} \]

Likelihood is artificially higher for things with high loss — training needs to work even harder to maximize the likelihood of the right thing!

Biased estimator for original likelihood, but better loss

Gimpel and Smith (2010)
Although he originally won the event, the United States Anti-Doping Agency announced in August 2012 that they had disqualified Armstrong from his seven consecutive Tour de France wins from 1999–2005.

Lance Edward Armstrong is an American former professional road cyclist.

Armstrong County is a county in Pennsylvania...

- 4.5M classes, not enough data to learn features like “Tour de France <-> en/wiki/Lance_Armstrong”
- Instead, features $f(x, y)$ look at the actual article associated with $y$
Although he originally won the event, the United States Anti-Doping Agency announced in August 2012 that they had disqualified Armstrong from his seven consecutive Tour de France wins from 1999–2005.

- \( \text{tf-idf}(\text{doc}, w) = \text{freq of } w \text{ in } \text{doc} \times \log(4.5M/\# \text{ Wiki articles } w \text{ occurs in}) \)
  - *the*: occurs in every article, \( \text{tf-idf} = 0 \)
  - *cyclist*: occurs in 1% of articles, \( \text{tf-idf} = \# \text{ occurrences} \times \log10(100) \)
- \( \text{tf-idf}(\text{doc}) = \text{vector of } \text{tf-idf}(\text{doc}, w) \text{ for all words in vocabulary (50,000)} \)
- \( f(x,y) = [\cos(\text{tf-idf}(x), \text{tf-idf}(y)), \ldots \text{ other features}] \)
Structured Prediction

- Four elements of a structured machine learning method:
  - Model: probabilistic, max-margin, deep neural network
  - Objective
  - Inference: just maxes so far, but will get harder
  - Training: gradient descent
Optimization

- Stochastic gradient *ascent*
  - Very simple to code up
  - “First-order” technique: only relies on having gradient
  - Difficult to tune step size
- Newton’s method
  - Second-order technique
  - Optimizes quadratic instantly
- Quasi-Newton methods: L-BFGS, etc.
  - Approximate inverse Hessian with gradients over time

\[ w \leftarrow w + \alpha g, \quad g = \frac{\partial}{\partial w} \mathcal{L} \]

\[ w \leftarrow w + \left( \frac{\partial^2}{\partial w^2} \mathcal{L} \right)^{-1} g \]

Inverse Hessian: \( n \times n \) mat, expensive!
AdaGrad

- Optimized for problems with sparse features
- Per-parameter learning rate: smaller updates are made to parameters that get updated frequently

\[
w_i \leftarrow w_i + \alpha \frac{1}{\sum_{\tau=1}^{t} g_{\tau,i}^2} g_{t,i}
\]

accumulate sum of squared gradients from previous updates

- Generally much more robust, requires little tuning of learning rates
- Other techniques for optimizing deep models — more later!

Duchi et al. (2011)
Structured Prediction

- Design tradeoffs need to reflect interactions:
  - Model and objective are coupled: probabilistic model $\leftrightarrow$ maximize likelihood
  - ...but not always: a linear model or neural network can be trained to minimize any differentiable loss function
  - Inference governs what learning: need to be able to compute expectations to use logistic regression
You’ve now seen everything you need to implement multi-class classification models

Next time: HMMs (POS tagging)

In 2 lectures: CRFs (NER)