Recall: Feature Extraction

On Sunday, Thomas and Mary went to the farmer’s market

- Feature extractor: function from (sentence, position) => sparse feature vector describing that position in the sentence
  - “Current word”: what is the word at this index?
  - “Previous word”: what is the word that precedes the index?

\[
\begin{bmatrix}
\text{currWord=Thomas} \\
\text{currWord=Mary} \\
\text{prevWord = and}
\end{bmatrix}
\]

\[
f(x, i=4) = \begin{bmatrix} 0 & 1 & 1 & \ldots \end{bmatrix}
\]

- Feature vector only has 2 nonzero entries out of 10k+ possible
- All features coexist in the same space! Other feats (char level, ...) possible

Recall: Binary Classification

Logistic regression:

\[
P(y = 1|x) = \frac{\exp \left( \sum_{i=1}^{n} w_i x_i \right)}{1 + \exp \left( \sum_{i=1}^{n} w_i x_i \right)}
\]

- These sums are sparse!

Decision rule:

\[
P(y = 1|x) \geq 0.5 \iff w^T x \geq 0
\]

Gradient: differentiate the log likelihood:

\[
x(y - P(y = 1|x))
\]

- This is the gradient of a single example. Can then apply stochastic gradient (or related optimization methods like Adagrad, etc.)

- ML pipeline: input -> feature representation, train model on labeled data (with stochastic gradient methods), then test on new data

Administrivia

- Course enrollment
- Mini 1 due Thursday at midnight (submit writeup on Gradescope + code/output on Canvas)
This Lecture

- Sentiment analysis
- Multiclass fundamentals
- Feature extraction
- Multiclass logistic regression

Sentiment Analysis

- This movie was great! I would watch again
- The movie was gross and overwrought, but I liked it
- 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

- Simple feature sets can do pretty well!

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

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

**Naive Bayes is doing well!**

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

**Best systems now: large pretrained networks**

**90 -> 97 over the last 2 years**

<table>
<thead>
<tr>
<th>Method</th>
<th>RT-s</th>
<th>MPQA</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNB-uni</td>
<td>77.9</td>
<td>85.3</td>
</tr>
<tr>
<td>MNB-bi</td>
<td>79.0</td>
<td>86.5</td>
</tr>
<tr>
<td>SVM-uni</td>
<td>76.2</td>
<td>86.1</td>
</tr>
<tr>
<td>SVM-bi</td>
<td>77.7</td>
<td>86.7</td>
</tr>
<tr>
<td>NBSVM-uni</td>
<td>78.1</td>
<td>85.3</td>
</tr>
<tr>
<td>NBSVM-bi</td>
<td>79.4</td>
<td>86.3</td>
</tr>
<tr>
<td>RAE</td>
<td>76.8</td>
<td>85.7</td>
</tr>
<tr>
<td>RAE-pretrain</td>
<td>77.7</td>
<td>86.4</td>
</tr>
<tr>
<td>Voting-w/Rev.</td>
<td>63.1</td>
<td>81.7</td>
</tr>
<tr>
<td>Rule</td>
<td>62.9</td>
<td>81.8</td>
</tr>
<tr>
<td>BoF-noDic.</td>
<td>75.7</td>
<td>81.8</td>
</tr>
<tr>
<td>BoF-w/Rev.</td>
<td>76.4</td>
<td>84.1</td>
</tr>
<tr>
<td>Tree-CRF</td>
<td>77.3</td>
<td>86.1</td>
</tr>
<tr>
<td>Kim (2014) CNNs</td>
<td>81.5</td>
<td>89.5</td>
</tr>
</tbody>
</table>

---

### Multiclass Fundamentals

#### Text Classification

- **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 GIRA KIZIOT

- **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 FRED BONNEY

---

### Stanford Sentiment Treebank (SST)

- **Binary classification**
- **Best systems now: large pretrained networks**
- **90 -> 97 over the last 2 years**

- XLM-Net-Large (ensemble) (Yang et al., 2018) 98.1
- MT-CNN-ensemble (Li et al., 2019) 98.5
- Snorkel/Meta(Lsemble) (Ratner et al., 2018) 96.2
- MT-CNN (Li et al., 2019) 95.6
- Bidirectional Encoder Representations from Transformers (Devlin et al., 2019) 94.9
- BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding 93.8

---

**Official**

- Neural Semantic Encoder (Mukhlisov and Yu, 2017) 89.7
- Text Classification Improved by Integrating Bidirectional LSTM with Two-dimensional Max Pooling 89.6

---

**Official**

- XLM-Net: Generalized Autoencoders for Cross-lingual Pre-training 98.0
- Improving Multi-Task Deep Neural Networks via Knowledge Distillation for Natural Language Understanding 98.5
- Training Complex Models with Multi-Task Weak Supervision 96.2
- Multi-Task Deep Neural Networks for Natural Language Understanding 95.6
- BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding 94.9

---

https://github.com/sebastianruder/NLP-progress/blob/master/english/sentiment_analysis.md
### Image Classification

- Dog
- Car

Thousands of classes (ImageNet)

### Entailment

- Three-class task over sentence pairs
  - A soccer game with multiple males playing.
    - ENTAILS
    - Some men are playing a sport.
  - Not clear how to do this with simple bag-of-words features
  - A black race car starts up in front of a crowd of people.
    - CONTRADICTS
    - A man is driving down a lonely road
  - A smiling costumed woman is holding an umbrella.
    - NEUTRAL
    - A happy woman in a fairy costume holds an umbrella.

Bowman et al. (2015)

### Entity Linking

- Lance Edward Armstrong is an American former professional road cyclist
- Armstrong County is a county in Pennsylvania...

4,500,000 classes (all articles in Wikipedia)

### Reading Comprehension

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)

Richardson (2013)
Binary Classification

- Binary classification: one weight vector defines positive and negative classes

Multiclass Classification

- Can we just use binary classifiers here?

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

- Can separate 1 from 2+3 and 2 from 1+3 but not 3 from the others (with these features)
Multiclass Classification

- Binary classification: one weight vector defines both classes
- Multiclass classification: different weights and/or features per class

决策规则:

-Can also have one weight vector per class: \( \arg\max_{y \in Y} w_y^T f(x) \)

Different Weights vs. Different Features

- Different features: \( \arg\max_{y \in Y} w^T f(x, y) \)

  - Suppose \( Y \) is a structured label space (part-of-speech tags for each word in a sentence). \( f(x,y) \) extracts features over shared parts of these

- Different weights: \( \arg\max_{y \in Y} w_y^T f(x) \)

  - Generalizes to neural networks: \( f(x) \) is the first \( n-1 \) layers of the network, then you multiply by a final linear layer at the end

  - For linear multiclass classification with discrete classes, these are identical

Feature Extraction
**Block Feature Vectors**

- Decision rule: \( \text{argmax}_{y \in \mathcal{Y}} w^\top f(x, y) \)
  
  *too many drug trials, too few patients*

- Base feature function:
  
  \[
  f(x) = \begin{cases} 
  1 & \text{[contains drug]}, \text{[contains patients]}, \text{[contains baseball]} \end{cases} = [1, 1, 0]
  \]

  feature vector blocks for each label

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

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

  Equivalent to having three weight vectors in this case

  We are NOT looking at the gold label! Instead looking at the candidate label

**Making Decisions**

*too many drug trials, too few patients*

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

"word drug in Science article" = +1.1

\[
\begin{align*}
  w &= [+2.1, +2.3, -5, -2.1, -3.8, +5.2, +1.1, -1.7, -1.3] \\
  w^\top f(x, y) &= \text{Health: +4.4 \hspace{1cm} Sports: -5.9 \hspace{1cm} Science: -0.6} \\
  \text{argmax}
\end{align*}
\]

**Part-of-speech tagging as multiclass**

- Classify *blocks* as one of 36 POS tags
  
  *the router blocks the packets*

- Example is a (sentence, index) pair \((x, i=2)\): the word *blocks* in this sentence

- Extract features with respect to this word:
  
  \[
  f(x, y = \text{VBZ}) = \begin{cases} 
  1[\text{curr_word=blocks} \& \text{tag = VBZ}], \\
  1[\text{prev_word=router} \& \text{tag = VBZ}], \\
  1[\text{next_word=the} \& \text{tag = VBZ}],
  1[\text{curr_suffix=s} \& \text{tag = VBZ}]
  \end{cases}
  \]

  not saying that *the* is tagged as VBZ! saying that *the* follows the VBZ word

**Multiclass Logistic Regression**

- Next two lectures: sequence labeling
### Training

- Multiclass logistic regression $P_w(y|x) = \frac{\exp(w^T f(x, y))}{\sum_{y' \in Y} \exp(w^T f(x, y'))}$
- Likelihood $L(x, y^*_i) = w^T f(x, y^*_i) - \log \sum_y \exp(w^T f(x, y))$
- Learning: gradient ascent on the discriminative log-likelihood $f(x, y^*) - \mathbb{E}_y[f(x, y)] = f(x, y^*) - \sum_y [P_w(y|x)f(x, y)]$
  
  “towards gold feature value, away from what the model thinks”

### Multiclass Logistic Regression: Summary

- Model: $P_w(y|x) = \frac{\exp(w^T f(x, y))}{\sum_{y' \in Y} \exp(w^T f(x, y'))}$
- Inference: $\text{argmax}_y P_w(y|x)$
- Learning: gradient ascent on the discriminative log-likelihood $f(x, y^*) - \mathbb{E}_y[f(x, y)] = f(x, y^*) - \sum_y [P_w(y|x)f(x, y)]$
  
  “towards gold feature value, away from what the model thinks”
Generative vs. Discriminative Models

Learning in Probabilistic Models

- So far we have talked about discriminative classifiers (e.g., logistic regression which models $P(y|x)$)
- Cannot analytically compute optimal weights for such models, need to use gradient descent
- What about generative models?

Naive Bayes

- Data point $x = (x_1, ..., x_n)$, label $y \in \{0, 1\}$
- Formulate a probabilistic model that places a distribution $P(x, y)$
- Compute $P(y|x)$, predict $\text{argmax}_y P(y|x)$ to classify

$$P(y|x) = \frac{P(y)P(x|y)}{P(x)}$$

Bayes’ Rule

$\propto P(y)P(x|y)$

constant: irrelevant for finding the max

$= P(y) \prod_{i=1}^{n} P(x_i|y)$

“Naive” assumption:

Maximum Likelihood Estimation

- Data points $(x_j, y_j)$ provided ($j$ indexes over examples)
- Find values of $P(y)$, $P(x_i|y)$ that maximize data likelihood (generative):

$$\prod_{j=1}^{m} P(y_j, x_j) = \prod_{j=1}^{m} P(y_j) \prod_{i=1}^{n} P(x_{ji}|y_j)$$

data points ($j$) \quad features ($i$) \quad $i$th feature of $j$th example
Maximum Likelihood Estimation

Imagine a coin flip which is heads with probability $p$

Observe $(H, H, H, T)$ and maximize likelihood:

$$\prod_{j=1}^{m} P(y_j) = p^3(1 - p)$$

Easier: maximize log likelihood

$$\sum_{j=1}^{m} \log P(y_j) = 3 \log p + \log(1 - p)$$

Maximum likelihood parameters for binomial/multinomial = read counts off of the data + normalize

Data points $(x_j, y_j)$ provided ($j$ indexes over examples)

Find values of $P(y)$, $P(x_i | y)$ that maximize data likelihood (generative):

$$\prod_{j=1}^{m} P(y_j, x_j) = \prod_{j=1}^{m} P(y_j) \prod_{i=1}^{n} P(x_{ji} | y_j)$$

Equivalent to maximizing log of data likelihood:

$$\sum_{j=1}^{m} \log P(y_j, x_j) = \sum_{j=1}^{m} \left[ \log P(y_j) + \sum_{i=1}^{n} \log P(x_{ji} | y_j) \right]$$

Can do this by counting and normalizing distributions!

Summary

Next time: HMMs / POS tagging

- Locally-normalized generative models, so easy to estimate from data
  - First thing we have that we could plausibly sample real sentences from

In 2 lectures: CRFs (NER)

You’ve now seen everything you need to implement multi-class classification models