Classification &
The Noisy Channel Model

CMSC 473/673
UMBC

Some slides adapted from 3SLP
Outline

Recap: language modeling

Classification
  Why incorporate uncertainty
  Posterior decoding
  Noisy channel decoding

Evaluation
Chain Rule

+ Backoff (Markov assumption)

= n-grams
## N-Gram Terminology

How to (efficiently) compute $p($Colorless green ideas sleep furiously$)$?

<table>
<thead>
<tr>
<th>$n$</th>
<th>Commonly called</th>
<th>History Size (Markov order)</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>unigram</td>
<td>0</td>
<td>$p($furiously$)$</td>
</tr>
<tr>
<td>2</td>
<td>bigram</td>
<td>1</td>
<td>$p($furiously $</td>
</tr>
<tr>
<td>3</td>
<td>trigram (3-gram)</td>
<td>2</td>
<td>$p($furiously $</td>
</tr>
<tr>
<td>4</td>
<td>4-gram</td>
<td>3</td>
<td>$p($furiously $</td>
</tr>
<tr>
<td>$n$</td>
<td>n-gram</td>
<td>$n-1$</td>
<td>$p($$w_i$ $</td>
</tr>
</tbody>
</table>
Language Models & Smoothing

Maximum likelihood (MLE): simple counting
Laplace smoothing, add- $\lambda$
Interpolation models
Discounted backoff

Interpolated (modified) Kneser-Ney
Good-Turing
Witten-Bell
Evaluation Framework

What is “correct?”

What is working “well?”

- Training Data: acquire primary statistics for learning model parameters
- Dev Data: fine-tune any secondary (hyper)parameters
- Test Data: perform final evaluation

DO NOT ITERATE ON THE TEST DATA
Setting Hyperparameters

Use a development corpus

Choose $\lambda$s to maximize the probability of dev data:

- Fix the N-gram probabilities/counts (on the training data)
- Search for $\lambda$s that give largest probability to held-out set
Evaluating Language Models

What is “correct?”
What is working “well?”

Extrinsic: Evaluate LM in downstream task
   Test an MT, ASR, etc. system and see which LM does better
   Propagate & conflate errors

Intrinsic: Treat LM as its own downstream task
   Use perplexity (from information theory)
Perplexity

Lower is better: lower perplexity --> less surprised

\[ \text{perplexity} = \exp\left(\frac{-1}{M} \sum_{i=1}^{M} \log p(w_i | h_i)\right) \]
Implementation: Unknown words

Create an unknown word token <UNK>

Training:
1. Create a fixed lexicon L of size V
2. Change any word not in L to <UNK>
3. Train LM as normal

Evaluation:
Use UNK probabilities for any word not in training
Implementation: EOS Padding

Create an end of sentence (“chunk”) token <EOS>

Don’t estimate $p(<\text{BOS}> \mid <\text{EOS}>)$

Training & Evaluation:
1. Identify “chunks” that are relevant (sentences, paragraphs, documents)
2. Append the <EOS> token to the end of the chunk
3. Train or evaluate LM as normal
Outline

Recap: language modeling

Classification
  Why incorporate uncertainty
  Posterior decoding
  Noisy channel decoding

Evaluation
Probabilistic Classification

Directly model the posterior

\[ p(X \mid Y) = h(X; Y) \]

Discriminatively trained classifier

Model the posterior with Bayes rule

\[ p(X \mid Y) \propto p(Y \mid X) * p(X) \]

Generatively trained classifier
Generative Training:
Two Different Philosophical Frameworks

\[ p(X \mid Y) = \frac{p(Y \mid X) \ast p(X)}{p(Y)} \]

- posterior probability
- likelihood
- prior probability
- marginal likelihood

Posterior Classification/Decoding
- maximum a posteriori

Noisy Channel Model Decoding

there are others too (CMSC 478/678)
Outline

Recap: language modeling

Classification

Why incorporate uncertainty
Posterior decoding
Noisy channel decoding

Evaluation
Three people have been fatally shot, and five people, including a mayor, were seriously wounded as a result of a Shining Path attack today against a community in Junin department, central Peruvian mountain region.
Three people have been fatally shot, and five people, including a mayor, were seriously wounded as a result of a Shining Path attack today against a community in Junin department, central Peruvian mountain region.
Electronic alerts have been used to assist the authorities in moments of chaos and potential danger: after the Boston bombing in 2013, when the Boston suspects were still at large, and last month in Los Angeles, during an active shooter scare at the airport.
Electronic alerts have been used to assist the authorities in moments of chaos and potential danger: after the Boston bombing in 2013, when the Boston suspects were still at large, and last month in Los Angeles, during an active shooter scare at the airport.
Classify with Uncertainty

\[
\text{best label} = \ arg \ max \ P(\text{label}|\text{example})
\]

Use probabilities
Classify with Uncertainty

\[ \text{best label} = \arg \max_{\text{label}} P(\text{label}|\text{example}) \]

*Use probabilities*

*There are non-probabilistic ways to handle uncertainty... but probabilities sure are handy!*
Electronic alerts have been used to assist the authorities in moments of chaos and potential danger: after the Boston bombing in 2013, when the Boston suspects were still at large, and last month in Los Angeles, during an active shooter scare at the airport.

<table>
<thead>
<tr>
<th>Category</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Politics</td>
<td>0.05</td>
</tr>
<tr>
<td>Terrorism</td>
<td>0.48</td>
</tr>
<tr>
<td>Sports</td>
<td>0.0001</td>
</tr>
<tr>
<td>Tech</td>
<td>0.39</td>
</tr>
<tr>
<td>Health</td>
<td>0.0001</td>
</tr>
<tr>
<td>Finance</td>
<td>0.0002</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>
Text Classification

Assigning subject categories, topics, or genres
Spam detection
Authorship identification

Age/gender identification
Language Identification
Sentiment analysis
...

Text Classification

Assigning subject categories, topics, or genres
Spam detection
Authorship identification

Age/gender identification
Language Identification
Sentiment analysis

Input:
- a document
- a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$

Output: a predicted class $c$ from $C$
Text Classification

Assigning subject categories, topics, or genres
Spam detection
Authorship identification

Age/gender identification
Language Identification
Sentiment analysis
...

Input:
- a document linguistic blob
- a fixed set of classes \( C = \{c_1, c_2, \ldots, c_J\} \)

Output: a predicted class \( c \) from \( C \)
Text Classification: Hand-coded Rules?

Assigning subject categories, topics, or genres
Spam detection
Authorship identification

Age/gender identification
Language Identification
Sentiment analysis

Rules based on combinations of words or other features
spam: black-list-address OR (“dollars” AND “have been selected”)

Accuracy can be high
If rules carefully refined by expert

Building and maintaining these rules is expensive

Can humans faithfully assign uncertainty?
Text Classification: Supervised Machine Learning

Assigning subject categories, topics, or genres
Spam detection
Authorship identification
Age/gender identification
Language Identification
Sentiment analysis
...

**Input:**
- a document \(d\)
- a fixed set of classes \(C = \{c_1, c_2, \ldots, c_J\}\)
- A training set of \(m\) hand-labeled documents \((d_1, c_1), \ldots, (d_m, c_m)\)

**Output:**
- a learned classifier \(\gamma\) that maps documents to classes
Text Classification: Supervised Machine Learning

Assigning subject categories, topics, or genres
Spam detection
Authorship identification

Age/gender identification
Language Identification
Sentiment analysis

Input:
- a document $d$
- a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$
- A training set of $m$ hand-labeled documents $(d_1, c_1), \ldots, (d_m, c_m)$

Output:
- a learned classifier $\gamma$ that maps documents to classes

Naïve Bayes
Logistic regression
Support-vector machines
k-Nearest Neighbors

...
Text Classification: Supervised Machine Learning

Assigning subject categories, topics, or genres
Spam detection
Authorship identification
Age/gender identification
Language Identification
Sentiment analysis

Input:
- a document $d$
- a fixed set of classes $C = \{c_1, c_2, ..., c_J\}$
- A training set of $m$ hand-labeled documents $(d_1, c_1), ..., (d_m, c_m)$

Output:
- a learned classifier $\gamma$ that maps documents to classes

Naïve Bayes
Logistic regression
Support-vector machines
k-Nearest Neighbors

...
Outline

Recap: language modeling

Classification
- Why incorporate uncertainty
- Posterior decoding
- Noisy channel decoding

Evaluation
Generative Training: Two Different Philosophical Frameworks

\[ p(X \mid Y) = \frac{p(Y \mid X) \ast p(X)}{p(Y)} \]

- **Posterior Classification/Decoding**
  - maximum a posteriori

- **Noisy Channel Model Decoding**
  - likelihood
  - prior probability
  - marginal likelihood (probability)
Probabilistic Text Classification

Assigning subject categories, topics, or genres
Spam detection
Authorship identification

Age/gender identification
Language Identification
Sentiment analysis
...

\[ p(X \mid Y) = \frac{p(Y \mid X) \ast p(X)}{p(Y)} \]
Probabilistic Text Classification

Assigning subject categories, topics, or genres
Spam detection
Authorship identification

Age/gender identification
Language Identification
Sentiment analysis
...

\[ p(X \mid Y) = \frac{p(Y \mid X) \ast p(X)}{p(Y)} \]

- class-based likelihood (language model)
- prior probability of class
- observation likelihood (averaged over all classes)
- observed data
- class
Probabilistic Text Classification

Assigning subject categories, topics, or genres
Spam detection
Authorship identification

Age/gender identification
Language Identification
Sentiment analysis

\[ p(X \mid Y) = \frac{p(Y \mid X) \ast p(X)}{p(Y)} \]

class-based likelihood (language model)
observation likelihood (averaged over all classes)
prior probability of class
observed data
class
Classification with Bayes Rule

\[ \text{argmax}_X p(X \mid Y) \]
Classification with Bayes Rule

\[
\text{argmax}_X \frac{p(Y \mid X) \ast p(X)}{p(Y)}
\]
Classification with Bayes Rule

\[ \text{argmax}_X \frac{p(Y \mid X) \ast p(X)}{p(Y)} \]

constant with respect to \( X \)
Classification with Bayes Rule

$\text{argmax}_X p(Y \mid X) \ast p(X)$
My Hobby:

Sitting down with grad students and timing how long it takes them to figure out that I'm not actually an expert in their field.

**Engineering:**

Our big problem is heat dissipation. Have you tried logarithms?

**Linguistics:**

Ah, so does this Finno-Ugric family include, say, Klingon?

**Sociology:**

Yeah, my latest work is on ranking people from best to worst.

**Literary Criticism:**

You see, the deconstruction is inextricable from not only the text, but also the self.

8 papers and 2 books and they haven't caught on.
MY HOBBY:
SITTING DOWN WITH GRAD STUDENTS AND TIMING HOW LONG IT TAKES THEM TO FIGURE OUT THAT I'M NOT ACTUALLY AN EXPERT IN THEIR FIELD.

ENGINEERING:
OUR BIG PROBLEM IS HEAT DISSIPATION
HAVE YOU TRIED LOGARITHMS?
48 SECONDS

LINGUISTICS:
AH, SO DOES THIS FINNOUGRIC FAMILY INCLUDE, SAY, KLINGON?

SOCIOLOGY:
YEAH, MY LATEST WORK IS ON RANKING PEOPLE FROM BEST TO WORST.
63 SECONDS

LITERARY CRITICISM:
YOU SEE, THE DECONSTRUCTION IS INEXTRICABLE FROM NOT ONLY THE TEXT, BUT ALSO THE SELF.
4 MINUTES

EIGHT PAPERS AND TWO BOOKS AND THEY HAVEN'T CAUGHT ON.
Classification with Bayes Rule

$$\text{argmax}_X \log p(Y \mid X) + \log p(X)$$
Classification (labels) with Bayes Rule

\[
\text{argmax}_X \log p(Y \mid X) + \log p(X)
\]

how likely is label X overall?

how well does text Y represent label X?

For “simple” or “flat” labels:
* iterate through labels
* evaluate score for each label, keeping only the best (n best)
* return the best (or n best) label and score
Classification/Decoding with Bayes Rule

\[
\arg\max_X \log p(Y \mid X) + \log p(X)
\]

-how likely is text (complex output) \(X\) overall?

-how well does text (complex input) \(Y\) represent text (complex output) \(X\)?

If \(X\) is a string (or some complex structure), this can be complicated

* iterate through labels
* evaluate score for each label, keeping only the best (n best)
* return the best (or n best) label and score
Outline

Recap: language modeling

Classification

Why incorporate uncertainty
Posterior decoding
Noisy channel decoding

Evaluation
Generative Training: 
Two Different Philosophical Frameworks

\[ p(X \mid Y) = \frac{p(Y \mid X) \ast p(X)}{p(Y)} \]

- posterior probability
- likelihood
- prior probability
- marginal likelihood (probability)

---

**Posterior Classification/Decoding**
- maximum a posteriori

**Noisy Channel Model Decoding**
Noisy Channel Model
what I want to tell you "sports"
Noisy Channel Model

what I want to tell you “sports”

what you actually see “The Os lost again…”
what I want to tell you “sports”

what you actually see “The Os lost again...”

hypothesized intent “sad stories” “sports”
Noisy Channel Model

what I want to tell you “sports”

what you actually see “The Os lost again…”

hypothesized intent “sad stories” “sports”

reweight according to what’s likely “sports”
Noisy Channel

Machine translation
Speech-to-text
Spelling correction
Text normalization

Part-of-speech tagging
Morphological analysis
Image captioning

\[ p(X \mid Y) = \frac{p(Y \mid X) \ast p(X)}{p(Y)} \]

possible (clean) output
observed (noisy) text

translation/decode model
(clean) language model

observation (noisy) likelihood
Noisy Channel

- Machine translation
- Speech-to-text
- Spelling correction
- Text normalization
- Part-of-speech tagging
- Morphological analysis
- Image captioning

\[
p(X \mid Y) = \frac{p(Y \mid X) \cdot p(X)}{p(Y)}
\]

possible (clean) output

observed (noisy) text

translation/decode model

(clean) language model

observation (noisy) likelihood
Language Model

Use any of the language modeling algorithms we’ve learned

Unigram, bigram, trigram

Add-λ, interpolation, backoff

(Later: Maxent, RNNs, hierarchical Bayesian LMs, ... )
Probabilistic Classification

\[ p(X \mid Y) = h(X; Y) \]

*Discriminatively trained classifier*

\[ p(X \mid Y) \propto p(Y \mid X) \times p(X) \]

*Generatively trained classifier*
Outline

Recap: language modeling

Classification
  Why incorporate uncertainty
  Posterior decoding
  Noisy channel decoding

Evaluation
Evaluation: the 2-by-2 contingency table

<table>
<thead>
<tr>
<th></th>
<th>Actually Correct</th>
<th>Actually Incorrect</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Selected/Guessed</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Not selected/not guessed</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Evaluation: the 2-by-2 contingency table

<table>
<thead>
<tr>
<th></th>
<th>Actually Correct</th>
<th>Actually Incorrect</th>
</tr>
</thead>
<tbody>
<tr>
<td>**Selected/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Guessed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>**Not selected/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>not guessed</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Classes/Choices
Evaluation: the 2-by-2 contingency table

<table>
<thead>
<tr>
<th></th>
<th>Actually Correct</th>
<th>Actually Incorrect</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Selected/ Guessed</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>True Positive (TP)</td>
<td>Blue Circle: Correct</td>
<td>Blue Circle: Guessed</td>
</tr>
<tr>
<td>Not selected/ not guessed</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Classes/Choices**
Evaluation: the 2-by-2 contingency table

<table>
<thead>
<tr>
<th>Selected/ Guessed</th>
<th>Actually Correct</th>
<th>Actually Incorrect</th>
</tr>
</thead>
<tbody>
<tr>
<td>True Positive</td>
<td>(TP) Correct</td>
<td>False Positive</td>
</tr>
<tr>
<td></td>
<td>(TP) Guessed</td>
<td>(FP) Correct</td>
</tr>
<tr>
<td>Not selected/ not guessed</td>
<td></td>
<td>False Positive Guessed</td>
</tr>
</tbody>
</table>
Evaluation: the 2-by-2 contingency table

<table>
<thead>
<tr>
<th></th>
<th>Actually Correct</th>
<th>Actually Incorrect</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Selected/ Guessed</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>True Positive</td>
<td>Correct</td>
<td>Guessed</td>
</tr>
<tr>
<td>False Negative</td>
<td>Correct</td>
<td>Guessed</td>
</tr>
<tr>
<td><strong>Not selected/ not guessed</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>False Positive</td>
<td>Correct</td>
<td>Guessed</td>
</tr>
</tbody>
</table>

**Classes/Choices**
### Evaluation: the 2-by-2 contingency table

<table>
<thead>
<tr>
<th>Selected/ Guessed</th>
<th>Actually Correct</th>
<th>Actually Incorrect</th>
</tr>
</thead>
<tbody>
<tr>
<td>True Positive (TP)</td>
<td>Correct</td>
<td>Guessed</td>
</tr>
<tr>
<td>Correct</td>
<td></td>
<td></td>
</tr>
<tr>
<td>False Negative (FN)</td>
<td>Correct</td>
<td>Guessed</td>
</tr>
<tr>
<td>Correct</td>
<td></td>
<td></td>
</tr>
<tr>
<td>False Positive (FP)</td>
<td>Correct</td>
<td>Guessed</td>
</tr>
<tr>
<td>Correct</td>
<td></td>
<td></td>
</tr>
<tr>
<td>True Negative (TN)</td>
<td>Correct</td>
<td>Guessed</td>
</tr>
<tr>
<td>Correct</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Accuracy, Precision, and Recall

**Accuracy**: % of items correct

\[
\frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}}
\]

<table>
<thead>
<tr>
<th>Selected/Guessed</th>
<th>Actually Correct</th>
<th>Actually Incorrect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selected/Guessed</td>
<td>True Positive (TP)</td>
<td>False Positive (FP)</td>
</tr>
<tr>
<td>Not select/not guessed</td>
<td>False Negative (FN)</td>
<td>True Negative (TN)</td>
</tr>
</tbody>
</table>
**Accuracy, Precision, and Recall**

**Accuracy**: % of items correct
\[
\frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}}
\]

**Precision**: % of selected items that are correct
\[
\frac{\text{TP}}{\text{TP} + \text{FP}}
\]

<table>
<thead>
<tr>
<th></th>
<th>Actually Correct</th>
<th>Actually Incorrect</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Selected/Guessed</strong></td>
<td>True Positive (TP)</td>
<td>False Positive (FP)</td>
</tr>
<tr>
<td><strong>Not select/not guessed</strong></td>
<td>False Negative (FN)</td>
<td>True Negative (TN)</td>
</tr>
</tbody>
</table>
Accuracy, Precision, and Recall

**Accuracy**: % of items correct
\[
\frac{TP + TN}{TP + FP + FN + TN}
\]

**Precision**: % of selected items that are correct
\[
\frac{TP}{TP + FP}
\]

**Recall**: % of correct items that are selected
\[
\frac{TP}{TP + FN}
\]

<table>
<thead>
<tr>
<th>Selected/Guessed</th>
<th>Actually Correct</th>
<th>Actually Incorrect</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Actually Correct</strong></td>
<td>True Positive (TP)</td>
<td>False Positive (FP)</td>
</tr>
<tr>
<td><strong>Not select/not guessed</strong></td>
<td>False Negative (FN)</td>
<td>True Negative (TN)</td>
</tr>
</tbody>
</table>
A combined measure: $F$

Weighted (harmonic) average of Precision & Recall

$$F = \frac{1}{\alpha \frac{1}{P} + (1 - \alpha) \frac{1}{R}}$$
A combined measure: F

Weighted (harmonic) average of Precision & Recall

\[ F = \frac{1}{\alpha \frac{1}{P} + (1 - \alpha) \frac{1}{R}} = \frac{(1 + \beta^2) \times P \times R}{(\beta^2 \times P) + R} \]

algebra (not important)
A combined measure: $F$

Weighted (harmonic) average of Precision & Recall

$$F = \frac{(1 + \beta^2) \cdot P \cdot R}{(\beta^2 \cdot P) + R}$$

Balanced F1 measure: $\beta=1$

$$F_1 = \frac{2 \cdot P \cdot R}{P + R}$$
Micro- vs. Macro-Averaging

If we have more than one class, how do we combine multiple performance measures into one quantity?

Macroaveraging: Compute performance for each class, then average.

Microaveraging: Collect decisions for all classes, compute contingency table, evaluate.
# Micro- vs. Macro-Averaging: Example

<table>
<thead>
<tr>
<th>Class 1</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Truth: yes</td>
<td>Truth: no</td>
</tr>
<tr>
<td>Classifier: yes</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Classifier: no</td>
<td>10</td>
<td>970</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Class 2</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Truth: yes</td>
<td>Truth: no</td>
</tr>
<tr>
<td>Classifier: yes</td>
<td>90</td>
<td>10</td>
</tr>
<tr>
<td>Classifier: no</td>
<td>10</td>
<td>890</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Micro Ave. Table</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Truth: yes</td>
<td>Truth: no</td>
</tr>
<tr>
<td>Classifier: yes</td>
<td>100</td>
<td>20</td>
</tr>
<tr>
<td>Classifier: no</td>
<td>20</td>
<td>1860</td>
</tr>
</tbody>
</table>

Macroaveraged precision: \( \frac{0.5 + 0.9}{2} = 0.7 \)

Microaveraged precision: \( \frac{100}{120} = 0.83 \)

Microaveraged score is dominated by score on common classes
Outline

Recap: language modeling

Classification
  Why incorporate uncertainty
  Posterior decoding
  Noisy channel decoding

Evaluation