TTIC 31210:
Advanced Natural Language Processing

Kevin Gimpel
Spring 2019

Lecture 1: Introduction
Course Overview

• Second time being offered (first was spring 2017)

• Prerequisite: TTIC 31190 (NLP)

• Aimed at second & third year PhD students

• My office hours (TTIC 531):
  – Mondays after class until 3:15pm
  – Wednesdays after class until 4pm
  – or by appointment

• Teaching assistant: Mingda Chen, 3rd year TTIC PhD student
  – TA office hours: Mondays 3-4pm, TTIC library (4th floor)
Course Web Page

https://ttic.uchicago.edu/~kgimpel/teaching/31210-s19/index.html

TTIC 31210: Advanced Natural Language Processing  lectures  assignments

This is the course webpage for the Spring 2019 version of TTIC 31210: Advanced Natural Language Processing. For the Spring 2017 course, go here.

Quarter: Spring 2019
Time: Monday/Wednesday 1:30-2:50pm
Location: Room 526 (fifth floor), TTIC

Instructor: Kevin Gimpel
Instructor Office Hours: Mondays 2:50-3:15pm, Wednesdays 2:50-4pm, Room 531
Teaching Assistant: Mingda Chen
Teaching Assistant Office Hours: Mondays 3-4pm, TTIC Library (fourth floor)

Prerequisites: TTIC 31190 or permission of the instructor.

Contents:
Textbooks
Grading
Topics
Collaboration Policy
Roadmap

• intro (today)
• deep learning for NLP (5 lectures)
• structured prediction: sequence labeling, syntactic and semantic parsing, dynamic programming (4 lectures)
• generative models, latent variables, unsupervised learning, variational autoencoders (2 lectures)
• Bayesian methods in NLP (2 lectures)
• Bayesian nonparametrics in NLP (2 lectures)
• review & other topics (1 lecture)

• I will be away at a conference (NAACL) the last week of classes (June 3-5), so we will cancel those two classes
Assignments

• Mini-research projects: implementation, experimentation, analysis, developing new methods

• Assignment 1 has been posted; due April 16
Assignments

• 1 (due ~4/16):
  – language modeling: loss function comparison, error analysis

• 2 (due ~4/30):
  – attention in text classification, self-attention, multiple heads

• 3 (due ~5/14):
  – exact and approximate decoding for hidden Markov models for part-of-speech tagging

• 4 (due ~5/29):
  – Gibbs sampling for inference in hidden Markov models and unsupervised part-of-speech tagging

• 5 (due ~6/12):
  – unsupervised tokenization with Bayesian nonparametrics
Grading

• 5 assignments (15% for each)
  – 5th assignment will be due during finals week
  – no final exam

• class participation (25%)
  – includes coming to class, participating, submitting occasional handouts, rare in-class quizzes

• if you have good reason to miss class, let me know!
Collaboration Policy

• You are welcome to discuss assignments with others in the course, but solutions and code must be written individually.
Lateness Policy

• If you turn in an assignment late, a penalty will be assessed (2% per hour late)

• You will have 4 late days to use as you wish during the quarter

• Late days must be used in whole increments
  – e.g., if you turn in an assignment 6 hours late and want to use a late day to avoid penalty, it will cost an entire late day to do so
Optional Textbooks (1/3)

- Jurafsky & Martin. *Speech and Language Processing*, 2nd Ed. & 3rd Ed.
- Many chapters of 3rd edition are online
- Copies of 2nd edition available in TTIC library
Optional Textbooks (2/3)

- Goldberg. *Neural Network Methods for Natural Language Processing*. 
- Earlier draft (from 2015) available online 
- Two copies on reserve in TTIC library
Optional Textbooks (3/3)

• Cohen. *Bayesian Analysis in Natural Language Processing*.
• Available in TTIC library
TTIC 31190 Topics

- words, morphology, lexical semantics
- text classification
- simple neural methods for NLP
- language modeling and word embeddings
- recurrent/recursive/convolutional networks in NLP
- sequence labeling, HMMs, dynamic programming
- syntax and syntactic parsing
- semantics, compositionality, semantic parsing
- machine translation and other NLP tasks
What is natural language processing?
What is natural language processing?

an experimental computer science research area that includes problems and solutions related to the understanding of human language
User-Facing Applications

Supporting Technologies

Language Understanding Capabilities
User-Facing Applications

Supporting Technologies

Language Understanding Capabilities
Text Classification

- spam / not spam
- priority level
- category
- sentiment
Sentiment Analysis

**Searched Term:** starbucks

<table>
<thead>
<tr>
<th>Positive Tweets</th>
<th>Neutral Tweets</th>
<th>Negative Tweets</th>
<th>Total Tweets</th>
</tr>
</thead>
<tbody>
<tr>
<td>708</td>
<td>4495</td>
<td>234</td>
<td>5437</td>
</tr>
</tbody>
</table>

**Sentiment Breakdown:**

- **13.02% Positive**
- **82.67% Neutral**
- **4.30% Negative**

Examples:

- Positive: "I like how that girl @ starbucks tonight let me stand in line for 10 mins w/ another dude in front of me, before saying “oh. I’m closed…”"
- Neutral: "sleep so i can do a ton of darkroom tomorrow i have to resist the starbucks though if i want enough money for the bus"
- Negative: "@macoy sore throat from the dark roast cheesecake? @rom have you tried the dark roast cheesecake at starbucks? its my addiction for the week..."

**Tweets on 2008-10-23:** Sitting in Starbucks, drinking Verona, and writing a sermon about the pure in heart... http://tinyurl.com/57ztx2d

...i'm really really thinking about not showing up for work tomorrow... or ever again... god i'm so pissed... i hate starbucks..."
Auto-Complete

To: Phil Schiller

Subject: WWDC rehearsal

Tomorrow we’re supposed to talk about the screen content.
The meeting was |

cancelled  rescheduled  moved

Q W E R T Y U I O P
A S D F G H J K L
Z X C V B N M

123 space return
Turkey!

dcorrado 5:37 PM
to me

Hi all,
We wanted to invite you to join us for an early Thanksgiving on November 22nd, beginning around 2PM. Please bring your favorite dish! RSVP by next week.

Dave

Server issues

Dan Mané 5:22 PM
to me

Hi team,
The server appears to be dropping about 10% of requests (see attached dashboards). There hasn’t been a new release since last night, so I’m not sure what’s going on. Is anyone looking into this?

Dan Mané 5:22 PM
Reply

I’ll check on it.

I’ll see if I can find out.

I’m on it.

Count us in! We’ll be there! Sorry, we won’t be able to make it.
Machine Translation

Where is the train station?

¿Dónde está la estación de tren?
The Apple Watch has drawbacks. There are other smartwatches that offer more capabilities.
Question Answering
Question Answering

“Alexa, who was President when Barack Obama was nine?”

“Alexa, how’s my commute?”

“Alexa, what’s the weather?”

“Alexa, did the 49ers win?”
User-Facing Applications

Supporting Technologies

Language Understanding Capabilities
## Word Sense Disambiguation

<table>
<thead>
<tr>
<th>input</th>
<th>output</th>
</tr>
</thead>
<tbody>
<tr>
<td>he’s a <strong>bass</strong> in the choir .</td>
<td><strong>bass_3</strong></td>
</tr>
<tr>
<td>our <strong>bass</strong> is line-caught from the Atlantic .</td>
<td><strong>bass_4</strong></td>
</tr>
</tbody>
</table>

Set of possible outputs = \{**bass_1**, **bass_2**, ..., **bass_8**\}

- **S:** (n) **bass** (the lowest part of the musical range)
- **S:** (n) **bass**, **bass part** (the lowest part in polyphonic music)
- **S:** (n) **bass**, **basso** (an adult male singer with the lowest voice)
- **S:** (n) **sea bass**, **bass** (the lean flesh of a saltwater fish of the family Serranidae)
- **S:** (n) **freshwater bass**, **bass** (any of various North American freshwater fish with lean flesh (especially of the genus Micropterus))
- **S:** (n) **bass**, **bass voice**, **basso** (the lowest adult male singing voice)
- **S:** (n) **bass** (the member with the lowest range of a family of musical instruments)
- **S:** (n) **bass** (nontechnical name for any of numerous edible marine and freshwater spiny-finned fishes)
Some questioned if Tim Cook’s first product would be a breakaway hit for Apple.
Some questioned if Tim Cook’s first product would be a breakaway hit for Apple.
Constituency Parsing

(S (NP the man) (VP walked (PP to (NP the park))))

Key:
S = sentence
NP = noun phrase
VP = verb phrase
PP = prepositional phrase
DT = determiner
NN = noun
VBD = verb (past tense)
IN = preposition

the man walked to the park
Cons constituency	
  Parsing

S = sentence
NP = noun phrase
VP = verb phrase
PP = prepositional phrase
DT = determiner
NN = noun
VBD = verb (past tense)
IN = preposition

the man walked to the park
Dependency Parsing

Key:
NSUBJ = nominal subject
DET = determiner
PREP = propositional modifier
POBJ = object of a preposition

The sentence 'The man walked to the park' is parsed as follows:

- **NSUBJ**: man
- **DET**: the
- **PREP**: to
- **POBJ**: the
- **VBD**: walked

The dependency relationships show the structure of the sentence:
- The man is the subject of the sentence.
- The man is followed by the prepositional phrase 'to the park'.
- 'The' is the determiner of 'man'.
- 'The' is the determiner of 'park'.

This diagram illustrates how dependency parsing breaks down a sentence into its grammatical components.
Semantic Parsing

• semantic role labeling (SRL)
• frame-semantic parsing
• semantic dependency formalisms
• abstract meaning representation (AMR)
Semantic Role Labeling

yesterday at the park the man gave crumbs to the birds

ARG0 = usually *agent*

ARG1 = typically *patient* or *theme*

ARG2 = often *beneficiary*
Semantic Role Labeling

yesterday at the park the man gave crumbs to the birds

ARG0 = usually agent
ARG1 = typically patient or theme
ARG2 = often beneficiary
Some questioned if Tim Cook’s first product would be a breakaway hit for Apple.
Some questioned if Tim Cook’s first product would be a breakaway hit for Apple.

Tim Cook

For other people named Tim Cook, see Tim Cook (disambiguation).

Timothy Donald Cook (born November 1, 1960) is an American business executive, industrial engineer, and developer. Cook is the Chief Executive Officer of Apple Inc., previously serving as the company’s Chief Operating Officer, under its founder Steve Jobs.[4]

Cook joined Apple in March 1998

Apple Inc.

Apple Inc. is an American multinational technology company headquartered in Cupertino, California, that designs, develops, and sells consumer electronics, computer software, and online services. The company’s hardware products include the iPhone smartphone, the iPad tablet computer, the Mac personal computer, the iPod portable...
Coreference Resolution

The boy threw some bread to a group of birds. They fought over it as he watched.
User-Facing Applications

Supporting Technologies

Language Understanding Capabilities
User-Facing Applications

Supporting Technologies

Language Understanding Capabilities
The man couldn't lift his son because he was so weak.

The man couldn't lift his son because he was so heavy.
The man couldn't lift his son because he was so weak.

The man couldn't lift his son because he was so heavy.
Natural Language Inference

• A **entails** B:
  A: yesterday’s game was canceled due to the rain.
  B: it rained yesterday.

• B **contradicts** A:
  A: yesterday’s game was canceled due to the rain.
  B: it didn’t rain yesterday.

• A and B are **neutral**:
  A: yesterday’s game was canceled due to the rain.
  B: since I had the day free, I cleaned my basement.
Once there was a boy named Fritz who loved to draw. He drew everything. In the morning, he drew a picture of his cereal with milk. His papa said, “Don’t draw your cereal. Eat it!”

After school, Fritz drew a picture of his bicycle. His uncle said, “Don't draw your bicycle. Ride it!”
Once there was a boy named Fritz who loved to draw. He drew everything. In the morning, he drew a picture of his cereal with milk. His papa said, “Don’t draw your cereal. Eat it!” After school, Fritz drew a picture of his bicycle. His uncle said, “Don't draw your bicycle. Ride it!”

What did Fritz draw first?

A) the toothpaste
B) his mama
C) cereal and milk
D) his bicycle
Once there was a boy named Fritz who loved to draw. He drew everything. In the morning, he drew a picture of his cereal with milk. His papa said, “Don’t draw your cereal. Eat it!”

After school, Fritz drew a picture of his bicycle. His uncle said, “Don't draw your bicycle. Ride it!”

What did Fritz draw first?

A) the toothpaste
B) his mama
C) cereal and milk
D) his bicycle
A Turing machine is a mathematical model of a general computing machine. It is a theoretical device that manipulates symbols contained on a strip of tape. Turing machines are not intended as a practical computing technology, but rather as a thought experiment representing a computing machine—anything from an advanced supercomputer to a mathematician with a pencil and paper. It is believed that if a problem can be solved by an algorithm, there exists a Turing machine that solves the problem. Indeed, this is the statement of the Church-Turing thesis. Furthermore, it is known that everything that can be computed on other models of computation known to us today, such as a RAM machine, Conway's Game of Life, cellular automata or any programming language can be computed on a Turing machine. Since Turing machines are easy to analyze mathematically, and are believed to be as powerful as any other model of computation, the Turing machine is the most commonly used model in complexity theory.

What is the term for a mathematical model that theoretically represents a general computing machine?

*Ground Truth Answers:* A Turing machine

*Prediction:* A Turing machine

It is generally assumed that a Turing machine can solve anything capable of also being solved using what?

*Ground Truth Answers:* an algorithm

*Prediction:* RAM machine, Conway's Game of Life, cellular automata or any programming language

What is the most commonplace model utilized in complexity theory?

*Ground Truth Answers:* the Turing machine

*Prediction:* Turing machine

What does a Turing machine handle on a strip of tape?

*Ground Truth Answers:* symbols

*Prediction:* general computing machine
<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Other ways are needed.</td>
<td>4.4</td>
</tr>
<tr>
<td>We must find other ways.</td>
<td></td>
</tr>
<tr>
<td>Pakistan bomb victims’ families end protest</td>
<td>2.6</td>
</tr>
<tr>
<td>Pakistan bomb victims to be buried after protest ends</td>
<td></td>
</tr>
<tr>
<td>I absolutely do believe there was an iceberg in those waters.</td>
<td>1.2</td>
</tr>
<tr>
<td>I don't believe there was any iceberg at all anywhere near the Titanic.</td>
<td></td>
</tr>
</tbody>
</table>
he bent down and searched the large container, trying to find anything else hidden in it other than the _____
he turned to one of the cops beside him. “search the entire coffin.” the man nodded and bustled forward towards the coffin.

he bent down and searched the large container, trying to find anything else hidden in it other than the ______
Why is NLP hard?

**ambiguity**: one form can mean many things

**variability**: many forms can mean the same thing
Why is NLP hard?

**ambiguity**: one form can mean many things

**variability**: many forms can mean the same thing

many different kinds of variability and ambiguity

each NLP task must address distinct kinds
Example: Hyperlinks in Wikipedia

Wikipedia Articles

bar (law)
bar (establishment)
bar association
... bar (unit)
medal bar
... bar (music)
...
Example: Hyperlinks in Wikipedia

Wikipedia Articles

bar

- bar (law)
- bar (establishment)
- bar association
- ... bar (unit)
- medal bar
- ... bar (music)

Ambiguity
Example: Hyperlinks in Wikipedia

Wikipedia Articles

- bar (law)
- bar (establishment)
- bar association
- bar (unit)
- medal bar
- bar (music)

Ambiguity

Variability

- bar
- bars
- saloon
- saloons
- lounge
- pub
- sports bar
What is a classifier?

• a function from inputs $x$ to outputs $y$

• one simple type of classifier:
  
  – for any input $x$, assign a score to each output $y$, parameterized by parameters $w$:

    $$\text{score}(x, y, w)$$

  – classify by choosing highest-scoring output:

    $$\text{classify}(x, w) = \arg\max_y \text{score}(x, y, w)$$
NotaBon

**Notation**

\[ \mathbf{u} = \text{a vector} \]

\[ u_i = \text{entry } i \text{ in the vector} \]

\[ \mathbf{W} = \text{a matrix} \]

\[ w_{ij} = \text{entry } (i,j) \text{ in the matrix} \]

\[ \mathbf{x} = \text{a structured object} \]

\[ x_i = \text{item } i \text{ in the structured object} \]
Modeling, Inference, Learning

**Inference**: solve \( \text{argmax} \)

\[
\text{classify}(x, w) = \text{argmax}_y \text{score}(x, y, w)
\]

**Modeling**: define score function

**Learning**: choose \( w \)
## Applications of our Classifier Framework

<table>
<thead>
<tr>
<th>task</th>
<th>input (x)</th>
<th>output (y)</th>
<th>output space ($\mathcal{L}$)</th>
<th>size of $\mathcal{L}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>text classification</td>
<td>a sentence</td>
<td>gold standard label for $x$</td>
<td>pre-defined, small label set (e.g., {positive, negative})</td>
<td>2-10</td>
</tr>
<tr>
<td>word sense disambiguation</td>
<td>instance of a particular word (e.g., bass) with its context</td>
<td>gold standard word sense of target word</td>
<td>pre-defined sense inventory from WordNet for bass</td>
<td>2-30</td>
</tr>
<tr>
<td>learning skip-gram word embeddings</td>
<td>instance of a word in a corpus</td>
<td>a word in the context of $x$ in a corpus</td>
<td>vocabulary</td>
<td>$</td>
</tr>
<tr>
<td>part-of-speech tagging</td>
<td>a sentence</td>
<td>gold standard part-of-speech tags for $x$</td>
<td>all possible part-of-speech tag sequences with same length as $x$</td>
<td>$</td>
</tr>
</tbody>
</table>
## Applications of our Classifier Framework

<table>
<thead>
<tr>
<th>task</th>
<th>input ((x))</th>
<th>output ((y))</th>
<th>output space ((\mathcal{L}))</th>
<th>size of (\mathcal{L})</th>
</tr>
</thead>
<tbody>
<tr>
<td>text classification</td>
<td>a sentence</td>
<td>gold standard label for (x)</td>
<td>pre-defined, small label set (e.g., {positive, negative})</td>
<td>2-10</td>
</tr>
<tr>
<td>word sense disambiguation</td>
<td>instance of a particular word (e.g., bass) with its context</td>
<td>gold standard word sense of (x)</td>
<td>pre-defined sense inventory from (\text{WordNet})</td>
<td>2-30</td>
</tr>
<tr>
<td>learning skip-gram word embeddings</td>
<td>instance of a particular word in a corpus</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>part-of-speech tagging</td>
<td>a sentence</td>
<td>gold standard part-of-speech tags for (x)</td>
<td>all possible part-of-speech tag sequences with same length as (x)</td>
<td>(</td>
</tr>
</tbody>
</table>
## Applications of Classifier Framework (continued)

<table>
<thead>
<tr>
<th>task</th>
<th>input (x)</th>
<th>output (y)</th>
<th>output space ((\mathcal{L}))</th>
<th>size of (\mathcal{L})</th>
</tr>
</thead>
<tbody>
<tr>
<td>named entity recognition</td>
<td>a sentence</td>
<td>gold standard named entity labels for (x) (BIO tags)</td>
<td>all possible BIO label sequences with same length as (x)</td>
<td>(</td>
</tr>
<tr>
<td>constituency parsing</td>
<td>a sentence</td>
<td>gold standard constituent parse (labeled bracketing) of (x)</td>
<td>all possible labeled bracketings of (x)</td>
<td>exponential in length of (x) (Catalan number)</td>
</tr>
<tr>
<td>dependency parsing</td>
<td>a sentence</td>
<td>gold standard dependency parse (labeled directed spanning tree) of (x)</td>
<td>all possible labeled directed spanning trees of (x)</td>
<td>exponential in length of (x)</td>
</tr>
<tr>
<td>machine translation</td>
<td>a sentence</td>
<td>a translation of (x)</td>
<td>all possible translations of (x)</td>
<td>potentially infinite</td>
</tr>
</tbody>
</table>
Modeling

classify(\(x, w\)) = \text{argmax}_{y} \text{score}(x, y, w)
Linear Models

• parameters are arranged in a vector $\mathbf{w}$
• score function is linear in $\mathbf{w}$:

$$\text{score}(\mathbf{x}, \mathbf{y}, \mathbf{w}) = \mathbf{w}^\top \mathbf{f}(\mathbf{x}, \mathbf{y}) = \sum_i w_i f_i(\mathbf{x}, \mathbf{y})$$

• $\mathbf{f}$: vector of feature functions
• each feature function can look at the entire input and output
Notation

\[ \mathbf{u} = \text{a vector} \]
\[ u_i = \text{entry } i \text{ in the vector} \]

\[ \mathbf{W} = \text{a matrix} \]
\[ w_{ij} = \text{entry } (i,j) \text{ in the matrix} \]

\[ \mathbf{x} = \text{a structured object} \]
\[ x_i = \text{item } i \text{ in the structured object} \]
## Stochastic/Generative Models

<table>
<thead>
<tr>
<th>model</th>
<th>tasks</th>
<th>context expansion</th>
</tr>
</thead>
<tbody>
<tr>
<td>n-gram language models</td>
<td>language modeling (for MT, ASR, etc.)</td>
<td>increase $n$</td>
</tr>
<tr>
<td>hidden Markov models</td>
<td>part-of-speech tagging, named entity recognition, word clustering</td>
<td>increase order of HMM (e.g., bigram HMM $\rightarrow$ trigram HMM)</td>
</tr>
<tr>
<td>probabilistic context-free grammars</td>
<td>constituency parsing</td>
<td>increase size of rules, e.g., flattening, parent annotation, etc.</td>
</tr>
</tbody>
</table>

- all use maximum likelihood estimation + smoothing (though different kinds)
- form of features dependent on “generative story”
- all assign probabilities to sentences (or to pairs of $<$sentence, something else$>$)
- trade-off between increasing “context” and needing more data / better smoothing
Model Families

• linear models
  – lots of freedom in defining features, though feature engineering required for best performance
  – learning uses optimization of a loss function
  – one can (try to) interpret learned feature weights

• stochastic/generative models
  – linear models with simple “features” (counts of events)
  – learning is easy: count & normalize (but smoothing needed)
  – easy to generate samples

• neural models
  – less feature engineering required (“features” are learned)
  – learning uses optimization of a loss function
  – works well; hard to interpret
Inference

\[
\text{classify}(x, w) = \arg\max_y \score(x, y, w)
\]
Inference for Structured Prediction

\[ \text{classify}(x, w) = \arg\max_y \text{score}(x, y, w) \]

• how do we efficiently search over the space of all structured outputs?

• this space may have size exponential in the size of the input, or be unbounded
Feature Locality

• feature locality: how “big” are your features?
• we need to be mindful of this to enable efficient inference
• features can be arbitrarily big in terms of the input
• but features cannot be arbitrarily big in terms of the output!
Inference in HMMs

\[ classify(x, w) = \arg\max_y p_w(x, y) \]

- since the output is a sequence, this argmax requires iterating over an exponentially-large set

- we can use **dynamic programming (DP)** to solve these problems exactly

- for HMMs (and other sequence models), the algorithm for solving this is the **Viterbi algorithm**
Learning

classify($x, w$) = argmax$_y$ score($x, y, w$)

**learning**: choose $w$
Cost Functions

• **cost function**: scores outputs against a gold standard
  \[ \text{cost} : \mathcal{L} \times \mathcal{L} \rightarrow \mathbb{R}_{\geq 0} \]

• should be as close as possible to the actual evaluation metric for your task

• typical cost for multi-class classification:
  \[ \text{cost}(y, y') = \mathbb{I}[y \neq y'] \]
Empirical Risk Minimization
(Vapnik et al.)

• replace expectation with sum over examples:

\[ \hat{w} = \arg\min_w \mathbb{E}_{P(x,y)} [\text{cost}(y, \text{classify}(x, w))] \]

\[ \hat{w} = \arg\min_w \sum_{i=1}^{|\mathcal{T}|} \text{cost}(y^{(i)}, \text{classify}(x^{(i)}, w)) \]
solution: replace “cost loss” (also called “0-1” loss) with a surrogate function that is easier to optimize

\[ \hat{w} = \arg\min_w \sum_{i=1}^{\mathcal{T}} \text{cost}(y^{(i)}, \text{classify}(x^{(i)}, w)) \]

generalize to permit any loss function

\[ \hat{w} = \arg\min_w \sum_{i=1}^{\mathcal{T}} \text{loss}(x^{(i)}, y^{(i)}, w) \]

cost loss / 0-1 loss: \[ \text{loss}_{\text{cost}}(x, y, w) = \text{cost}(y, \text{classify}(x, w)) \]
Surrogate Loss Functions

cost loss / 0-1 loss: \( \text{loss}_{\text{cost}}(x, y, w) = \text{cost}(y, \text{classify}(x, w)) \)

perceptron loss:
\[
\text{loss}_{\text{perc}}(x, y, w) = -\text{score}(x, y, w) + \max_{y' \in \mathcal{L}} \text{score}(x, y', w)
\]

hinge loss:
\[
\text{loss}_{\text{hinge}}(x, y, w) = -\text{score}(x, y, w) + \max_{y' \in \mathcal{L}} (\text{score}(x, y', w) + \text{cost}(y, y'))
\]

log loss:
\[
\text{loss}_{\text{log}}(x, y, w) = -\log p_w(y \mid x)
\]
Score $\rightarrow$ Probability

- can turn score into probability by exponentiating (to make it positive) and normalizing:

$$p_w(y \mid x) \propto \exp\{\text{score}(x, y, w)\}$$

$$p_w(y \mid x) = \frac{\exp\{\text{score}(x, y, w)\}}{\sum_{y' \in \mathcal{L}} \exp\{\text{score}(x, y', w)\}}$$

- this is often called a “softmax” function