Announcement

Some clarification on unit tests:

- Unit test is a practical way to increase our confidence that future changes don't break the intended functionality of existing code.
- We can't exhaustively test all possible inputs, as long as the function may accept infinitely many inputs; that is, it's hard to measure the recall of your tests (in terms of catching errors).
- For grading purposes, we will only check the correctness of your test case; however, we encourage you to avoid trivial test cases.
- **General rule of thumb**: if something doesn't exactly match the CHAT manual description in that specific section, leave that part as is in the input; otherwise, modify it as required.

CS 784: Computational Linguistics
Lecture 7: Text Classification

Freda Shi

School of Computer Science, University of Waterloo
fhs@uwaterloo.ca

January 28, 2025

Taxonomy of NLP/CL

Subareas in Linguistics:

- **Morphology**
- Syntax
- Semantics
- Pragmatics

Modeling Approaches:

- Classification **Classification**
- Language modeling
- Sequence-to-sequence modeling
- Structured prediction

The course is organized in a nested structure: the linguistic subareas as the general framework, and the modeling approaches as the building blocks.

Tasks: Sentiment Analysis

Sentiment analysis is the task of determining the sentiment of a piece of text.

The sentiment can be positive, negative, or neutral, or it can be more fine-grained.

Text	**Sentiment**
Great service for an affordable price.	Positive
Just booked two nights at this hotel.	Neutral
Horrible services.	Negative

[Source: Socher et al., 2013]

Tasks: Subjectivity vs. Objectivity Judgment

Text	**Label**
The hulk is an anger fueled monster with incredible strength and resistance to damage.	Obj.
In trying to be daring and original, it comes off as only occasionally satirical and never fresh.	Subj.
Solondz may well be the only one laughing at his own joke.	Subj.
Obstacles pop up left and right, as the adventure gets wilder and wilder.	Obj.

[Source: Pang and Lee, 2004]

Caveat: The task itself is, to some extent, subjective.

How was the dataset generated?

- IMDB plot summaries: objective.
- Rotten Tomatoes snippets: subjective.

The data could be noisy, but we make this reasonable assumption based on some common wisdom about the platforms.

Tasks: Question Type Classification

Text	**Label**
Who invented baseball?	Human
CNN is an acronym for what?	Abbreviation
Which South American country is the largest?	Location
How many small businesses are there in the US?	Number
What would you add to the clay mixture to produce bone china?	Entity
What is the root of all evil?	Description

[Source: Li and Roth, 2002]

There has been a long history in question classification in the Text Retrieval Conference (TREC) series.

Classification results helps QA system to route the question to the appropriate answer extraction module.

Linguistic Acceptability Judgment

A more linguistic oriented task – to determine whether a sentence is acceptable or not (to native speakers).

Text	**Label**
The more books I ask to whom he will give, the more he reads.	Unacceptable
The jeweler inscribed the ring with the name.	Acceptable
The gardener planted roses in the garden.	Acceptable
Who do you think that will question Seamus first?	Unacceptable
Kim persuaded it to rain.	**Unacceptable**

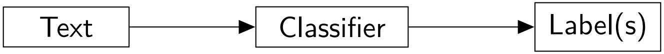
[Source: Warstadt et al., 2019]

Caveat: Humans don't even agree on some sentences.

For this task, why models predict the way they do is arguably more interesting than the prediction itself.

General tip: Look at your data **a lot** in experiments!

Text Classification: Task Formulation



At the inference stage,

- Input: text represented as a list of tokens $s = \langle w_1, \ldots, w_n \rangle$.
- Output: a set of categories $Y(s) \subseteq Y$, where Y is the set of all possible categories.

Possible types of classifiers:

- Rule-based classifier, e.g., list of rules, decision trees.
- Statistical classifier, e.g., naïve Bayes, logistic regression, support vector machines.
- Neural network–based classifier (modern statistical classifiers).

Rule-Based Classifiers

Taking sentiment classification as an example:

- If s contains words in $\{$good, excellent, nice, ...$\}$, then $Y(s) =$ Positive.
- If s contains words in $\{$bad, horrible, awful, ...$\}$, then $Y(s) =$ Negative.

Alternatively, we can train a decision tree to make the decision.

Pros 1: Perfect interpretability.

Pros 2: Can be very accurate with carefully refined rules.

Cons 1: Rules can be brittle and hard to maintain.

Cons 2: System can be very complicated when the number of rules grows.

Cons 3: Hard to generalize to unseen data.

Check out VADER (Valence Aware Dictionary and sEntiment Reasoner) for a rule-based sentiment analysis toolkit.

Statistical (or Data-Driven) Classifiers

Data-driven modeling of the text classification task.

At the **inference stage**,

- Input: text represented as a list of tokens $s = \langle w_1, \ldots, w_\ell \rangle$.
- Output: a set of categories $Y(s) \subseteq Y$, where Y is the set of all possible categories.

At the **training stage**,

- Input: a set of labeled examples $D = \{(s_1, Y(s_1)), \ldots, (s_n, Y(s_n))\}$.
- Output: a model that can predict the label(s) of text.

For simplicity, we assume that $Y(s_i)$ is a single category y_i.

We will discuss multi-label classification later.

$$\text{classify}(s) = \arg\max_y \mathit{score}(s, y; \mathbf{\Theta})$$

Statistical Classifier: Naïve Bayes

The Bayes' rule:

$$\underbrace{P(y \mid s)}_{\text{classfier}} = \frac{P(s \mid y)P(y)}{P(s)} \propto P(s \mid y)P(y)$$

$s:$ a sentence, $y:$ a category.

Estimate $P(s \mid y)$ and $P(y)$ for all s and y with training data D.

- $P(y)$: the probability of the category y.

$$P(y) = \frac{\text{count}_D(y)}{\sum_{y'} \text{count}_D(y')}$$

- $P(s \mid y)$: the probability of the sentence s given the category y.

$$P(s = \langle w_1, \ldots, w_\ell \rangle \mid y) = \prod_{i=1}^{\ell} P(w_i \mid y) \quad P(w \mid y) = \frac{\text{count}_D(w, y)}{\sum_{w'} \text{count}_D(w', y)}$$

Assumption: words in a sentence are conditionally independent given category.

The Bag-of-Words Assumption

The bag-of-words (BoW) assumption: the order of words in a sentence does not matter.

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!

love sweet humor
it are be it seen
and tale whimsical
and I'm I've it see
again friend it seen
recommend fairy to
anyone always have
hasn't genre humor...

it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
sweet	1
genre	1
fairy	1
humor	1
have	1
great	1
...	

Laplace Smoothing (Again)

$$P(y) = \frac{\text{count}_D(y)}{\sum_{y'} \text{count}_D(y')} \qquad P(w \mid y) \frac{\text{count}_D(w, y)}{\sum_{w'} \text{count}_D(w', y)}$$

If $\text{count}_D(w_i, y) = 0$ due to lack of presence in the training data, at the inference stage, $P(w_i \mid y) = 0 \Rightarrow P(s \mid y) = 0$.

This could be problematic: if there is an unseen word in a test sentence for certain category y, the model will assign $P(s \mid y) = 0$.

Solution: Laplace smoothing – add a small constant α to all the counts, and renormalize the probability.

$$P(w \mid y) = \frac{\text{count}_D(w, y) + \alpha}{\sum_{w'} \text{count}_D(w', y) + \alpha|V|}$$

$|V|$: the size of the vocabulary.

However, if there's a word that never appears in the training data, we may simply ignore it.

Naïve Bayes: Example

Split	**Category**	**Sentence**
Training	-	just plain boring
Training	-	entirely predictable and lacks energy
Training	-	no surprises and very few laughs
Training	+	very powerful
Training	+	the most fun film of the summer
Testing	?	predictable with no fun

Priors: $P(+) = 2/5$ $P(-) = 3/5$

Conditional Probabilities (with Laplace smoothing $\alpha = 1$):

$$|V| = 20$$

$$P(\text{predictable} \mid +) = \frac{0+1}{9+20} = \frac{1}{29}$$

$$P(\text{predictable} \mid -) = \frac{1+1}{14+20} = \frac{2}{34}$$

$$P(\text{no} \mid +) = \frac{0+1}{9+20} = \frac{1}{29}$$

$$P(\text{no} \mid -) = \frac{1+1}{14+20} = \frac{2}{34}$$

$$P(\text{fun} \mid +) = \frac{1+1}{9+20} = \frac{2}{29}$$

$$P(\text{fun} \mid -) = \frac{0+1}{14+20} = \frac{1}{34}$$

Naïve Bayes: Example (Cont.)

Priors: $P(+) = 2/5$ $P(-) = 3/5$

Conditional Probabilities (with Laplace smoothing $\alpha = 1$):

$$P(\text{predictable} \mid +) = \frac{0+1}{9+20} = \frac{1}{29} \qquad P(\text{no} \mid -) = \frac{1+1}{14+20} = \frac{2}{34}$$

$$P(\text{predictable} \mid -) = \frac{1+1}{14+20} = \frac{2}{34} \qquad P(\text{fun} \mid +) = \frac{1+1}{9+20} = \frac{2}{29}$$

$$P(\text{no} \mid +) = \frac{0+1}{9+20} = \frac{1}{29} \qquad P(\text{fun} \mid -) = \frac{0+1}{14+20} = \frac{1}{34}$$

$$P(+ \mid s) \propto P(+)P(\text{predictable} \mid +)P(\text{no} \mid +)P(\text{fun} \mid +)$$
$$= \frac{2}{5} \times \frac{1}{29} \times \frac{1}{29} \times \frac{2}{29} \approx 3.2 \times 10^{-5}$$

$$P(\text{-} \mid s) \propto P(-)P(\text{predictable} \mid -)P(\text{no} \mid -)P(\text{fun} \mid -)$$
$$= \frac{3}{5} \times \frac{2}{34} \times \frac{2}{34} \times \frac{1}{34} \approx 6.1 \times 10^{-5}$$

Statistical Classifier: Logistic Regression

Base case: binary classification for text classification.

Training data: $D = \{(s_1, y_1), \ldots, (s_n, y_n)\}, y_i \in \{0, 1\} (\forall i)$.

Each piece of text s_i will be represented as a feature vector $\mathbf{x}_i$.

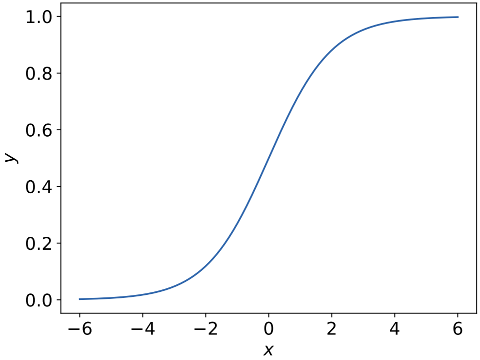
Train a model parameterized by $\mathbf{w}$ to predict the probability of y_i given $\mathbf{x}_i$ with the logistic function:

$$P(y_i = 1 \mid \mathbf{x}_i) = \sigma\left(\mathbf{w}^T \mathbf{x}_i\right) = \frac{1}{1 + \exp(-\mathbf{w}^T \mathbf{x}_i)}$$

$$P(y_i = 0 \mid \mathbf{x}_i) = 1 - \sigma\left(\mathbf{w}^T \mathbf{x}_i\right) = \frac{\exp(-\mathbf{w}^T \mathbf{x}_i)}{1 + \exp(-\mathbf{w}^T \mathbf{x}_i)}$$

$\sigma(\cdot)$: the sigmoid function; $\mathbf{w}, \mathbf{x}_i \in \mathbb{R}^d$.

The Sigmoid function



$$\sigma(x) = \frac{1}{1 + \exp(-x)}$$

It normalizes the output to $[0, 1]$, which has natural interpretation as probability.

Training Logistic Regression

- Training objective: maximize the likelihood of training data
  Assumption: each example is independent and identically distributed (i.i.d.)

$$\begin{aligned}\mathcal{L}(\mathbf{w}) &= \prod_{i=1}^{n} P(y_i \mid \mathbf{x}_i) \\ &= \prod_{i=1}^{n} \sigma(\mathbf{w}^T\mathbf{x}_i)^{y_i}(1-\sigma(\mathbf{w}^T\mathbf{x}_i))^{1-y_i}\end{aligned}$$

- Log converts multiplication to addition:

$$\log \mathcal{L}(\mathbf{w}) = \sum_{i=1}^{n}\left[y_i \log \sigma(\mathbf{w}^T\mathbf{x}_i) + (1-y_i)\log(1-\sigma(\mathbf{w}^T\mathbf{x}_i))\right]$$

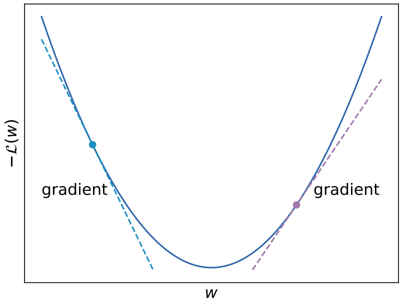
- Maximize the log-likelihood by **gradient ascent**, or minimize the negative log-likelihood loss by **gradient descent**:

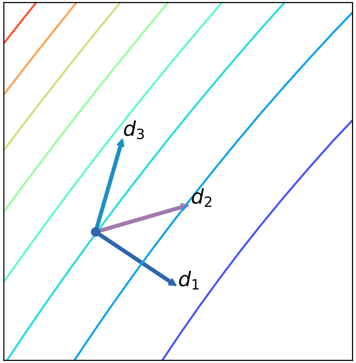
$$\mathbf{w}_{t+1} = \mathbf{w}_t + \eta \nabla_{\mathbf{w}} \log \mathcal{L}(\mathbf{w})$$

$$\nabla_{\mathbf{w}} \log \mathcal{L}(\mathbf{w}) = \left[\frac{\partial \log \mathcal{L}(\mathbf{w})}{\partial w_1}, \ldots, \frac{\partial \log \mathcal{L}(\mathbf{w})}{\partial w_d}\right]$$

η: learning rate, t: time step (iteration number).

Gradient Descent: The Idea





Find the direction of the steepest descent (i.e., the direction that is perpendicular to the contour lines), move slightly along that direction, and repeat until convergence.

See here for a formal proof.

Explain Logistic Regression

$$P(y = 1 \mid \mathbf{x}) = \sigma\left(\mathbf{w}^T\mathbf{x}\right) = \frac{1}{1 + \exp(\underbrace{-\mathbf{w}^T\mathbf{x}}_{\text{linear}})}$$

The weight vector **w** is a set of coefficients that are learned during training.

If each feature has the same variance, the weight vector **w** can be interpreted as the importance of each feature in the prediction.

Larger absolute values of a w_i means... the feature x_i has a larger impact on the prediction.

Generative vs. Discriminative Models

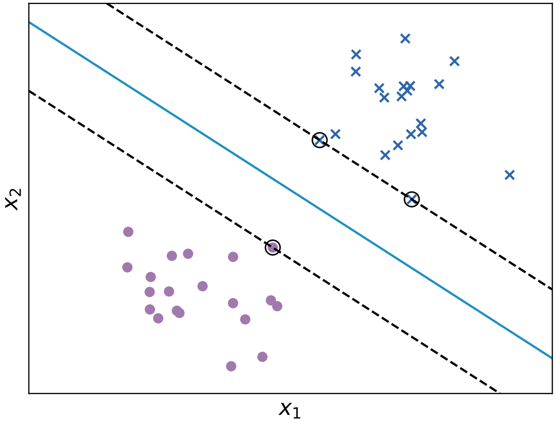
Generative models: model the joint distribution of the input and output $P(x, y)$. Once we have this distribution, we may **generate new data** by sampling from it.

Discriminative models: model the conditional distribution of the output given the input $P(y \mid x)$.

Naïve Bayes vs. logistic regression: which one is generative and which one is discriminative?

Support Vector Machines (SVMs)

Suppose the data is linearly separated, an SVM finds the hyperplane that maximizes the margin between the two classes.



SVM: Formulation

We have the training data $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, where $\mathbf{x}_i \in \mathbb{R}^d$ denotes features and $y_i \in \{-1, 1\}$ denotes a label.

We are interested in a large margin classifier:

$$\arg\max_{\mathbf{w},b} \left\{ \frac{1}{\|\mathbf{w}\|} \min_i y_i(\mathbf{w}^T \mathbf{x}_i + b) \right\}$$

Since we can scale $\mathbf{w}$ and b accordingly, we can assume $\min_i y_i(\mathbf{w}^T\mathbf{x}_i + b) = 1$ without loss of generality.

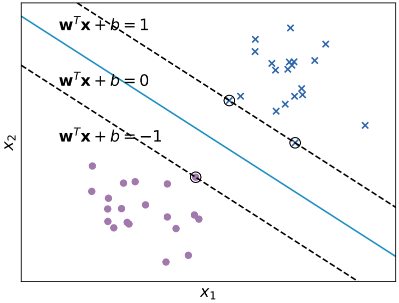
The problem turns to

$$\arg\min_{\mathbf{w},b} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 \right\}$$

$$\text{s.t. } \; y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \forall i$$

In the inference stage, we predict the label of a new data point $\mathbf{x}$ by $\text{sign}(\mathbf{w}^T\mathbf{x} + b)$.

SVM: The Idea



There are infinitely many (equivalent) ways to write down the decision boundary:

$$x_1 + x_2 - 5 = 0 \qquad \mathbf{w} = [1, 1]^T, b = -5$$

$$2x_1 + 2x_2 - 10 = 0 \qquad \mathbf{w} = [2, 2]^T, b = -10$$

$$\ldots$$

The **w** and b with $\min_i y_i \mathbf{w}^T\mathbf{x}_i + b = 1$ is just one representative within the equivalence class.

SVM: The Representor Theorem

The solution to the problem

$$\mathbf{w}^* = \arg\min_{\mathbf{w},b} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 \right\} \text{ s.t. } y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1, \forall i$$

can be represented as

$$\mathbf{w}^* = \sum_{i=1}^{n} \beta_i \mathbf{x}_i$$

Proof idea:

Let $\mathbf{w}^* = \mathbf{w}_\mathbf{X} + \mathbf{w}_\perp$, where $\mathbf{w}_\mathbf{X} \in \text{span}(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ and $\mathbf{w}_\perp \cdot \mathbf{w}_\mathbf{X} = 0$ is orthogonal to all $\mathbf{x}_i$.

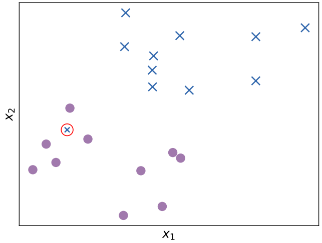
If $\mathbf{w}_\perp \neq 0$, we can always find a $\mathbf{w}$ with smaller $||\mathbf{w}||^2$ by removing the $\mathbf{w}_\perp$ component.

Practice: complete the proof.

SVM with Non-Separable Data

$$\mathbf{w}^* = \arg\min_{\mathbf{w},b} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 \right\} \text{ s.t. } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \forall i$$

What if the data is not linearly separable?



Add **slack variables** ξ_i to allow some misclassification:

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i \quad (\xi_i \geq 0)$$

$$\xi_i = \max\left\{0, 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b)\right\}$$

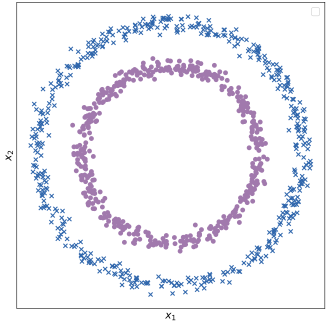
Minimize the SVM loss J with (sub)gradient descent.

$$J(\mathbf{w}, b) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i(\mathbf{w}, b)$$

C: balancing hyperparameter.

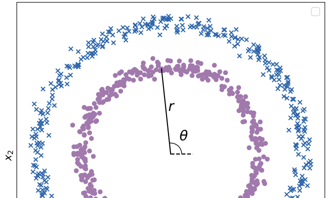
More about SVMs

Linear projection is sometimes not a good idea for the original data, but it could be after some (nonlinear) projection.



$$r = \sqrt{x_1^2 + x_2^2}$$

$$\theta = \arctan\left(\frac{x_2}{x_1}\right)$$



The **kernel trick**: We don't really need to know the projection function, as long as we can compute the dot product in the projected space.

Summary

The classifiers we've discussed so far

- Naïve Bayes

$$P(y \mid s) \propto P(s \mid y)P(y)$$

- Logistic regression

$$P(y=1 \mid \mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x}) \qquad\qquad y \in \{0,1\}$$

- SVM

$$\mathbf{w}^* = \arg\min_{\mathbf{w},b} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 \right\}$$
$$\text{s.t. } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \forall i (y_i \in \{-1,1\})$$

For logistic regression and SVM, we've only discussed binary classification.

Summary

$$\text{classify}(s) = \arg\max_{y} \mathit{score}(s, y; \mathbf{\Theta})$$

s: text, y: category, $\mathbf{\Theta}$: model parameters.

Model	**Probabilistic**	**G/D**
Naïve Bayes	Yes	Generative
Logistic Regression	Yes	Discriminative
SVM	No	Discriminative

Generalization to Multiple Classes

There are two common approaches to extend binary classifiers to support K classes:

- **One-vs-Rest**: Train K binary classifiers, one for each class.
  Each classifier distinguishes one class from the rest.
  During prediction, the class with the highest confidence score is chosen.
- **One-vs-One**: Train $\frac{K(K-1)}{2}$ binary classifiers, one for each pair of classes.
  During prediction, do majority voting among all classifiers.

If there is a tie, break it randomly or use any plausible strategy.

Next

Classification with Features from Advanced Neural-Net Structures

$$\begin{aligned}\text{logits}(\mathbf{x}) &= \text{NN}(\mathbf{x}) \in \mathbb{R}^K \\ P(y \mid \mathbf{x}) &= \text{softmax}(\text{logits}(\mathbf{x}))\end{aligned}$$

or

$$P(\mathbf{x}, y) \propto \text{NN}(\mathbf{x}, y)$$