TTIC 31190: Natural Language Processing

Lecture 4: NLP Datasets & Text Classification

Fall 2023

Recap

- Words/subwords: tokenization
- Vectorized representations:

  TF-IDF, PMI, continuous bags of words, skip grams

Schedule

Date	Topic	Instructor
W, 9/27	Introduction	Freda
M, 10/2	Word	Joe
W, 10/4	Distributional Semantics	Joe
M, 10/9	**Classification**	**Freda**
W, 10/11	Classification	Freda
M, 10/16	Neural Networks	Freda
W, 10/18	Neural Networks & Sequence Labeling	Freda
M, 10/23	Sequence Labeling	Freda
W, 10/25	Language Modeling	Joe
M, 10/30	Seq2Seq	Freda
W, 11/1	Seq2Seq & Syntax	Freda
M, 11/6	Syntax	Freda
W, 11/8	Semantics	Joe
M, 11/13	Semantics	Joe
W, 11/15	Pragmatics	Freda
M, 11/20	Thanksgiving Break	
W, 11/22	Thanksgiving Break	
M, 11/27	LLM: Pretraining and Finetuning	Joe
W, 11/29	LLM: Prompting and Multilingualism	Freda
M, 12/4	Reading Period	
TBD	Final Exam	

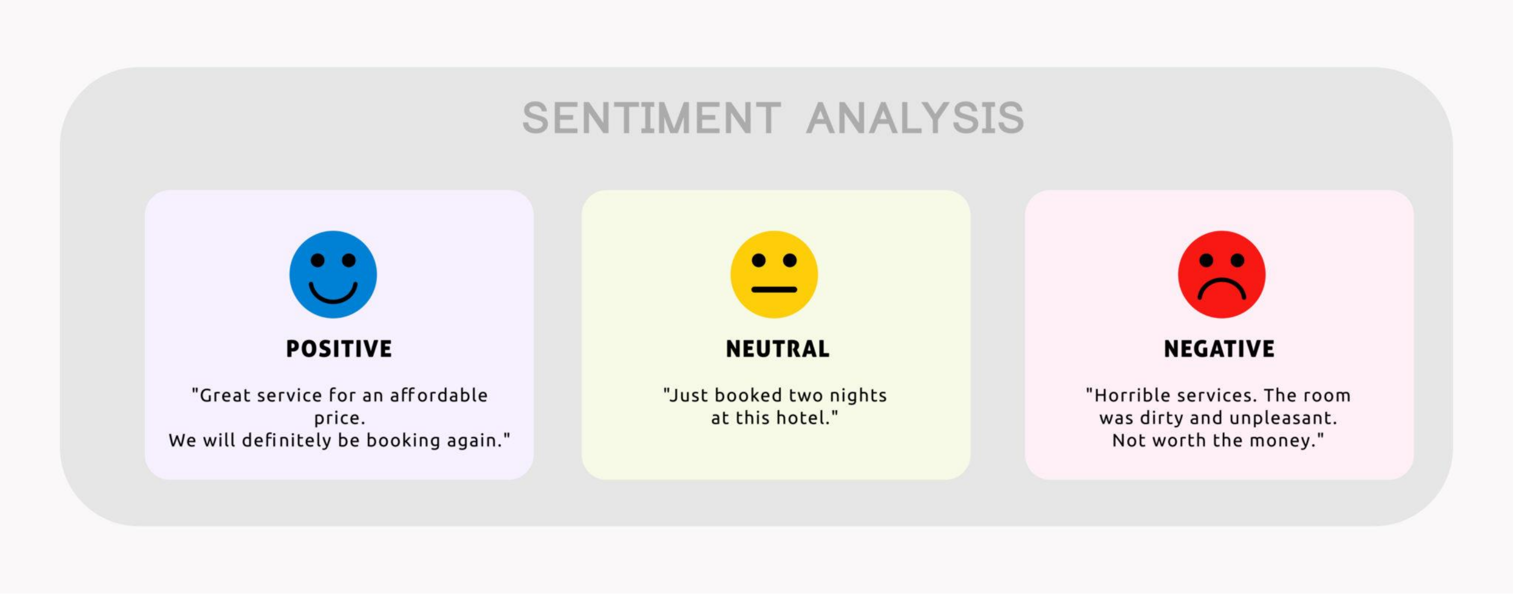
This Lecture: Beyond Word

- **NLP datasets**

- Text classification
  - Rule-based methods
  - Naïve Bayes
  - Logistic regression
  - Support vector machines

NLP Datasets

- NLP datasets typically include inputs (usually text) and outputs (usually some sort of annotation)



"Great service for an affordable price.
We will definitely be booking again."

"Just booked two nights at this hotel."

"Horrible services. The room was dirty and unpleasant.
Not worth the money."

Annotation

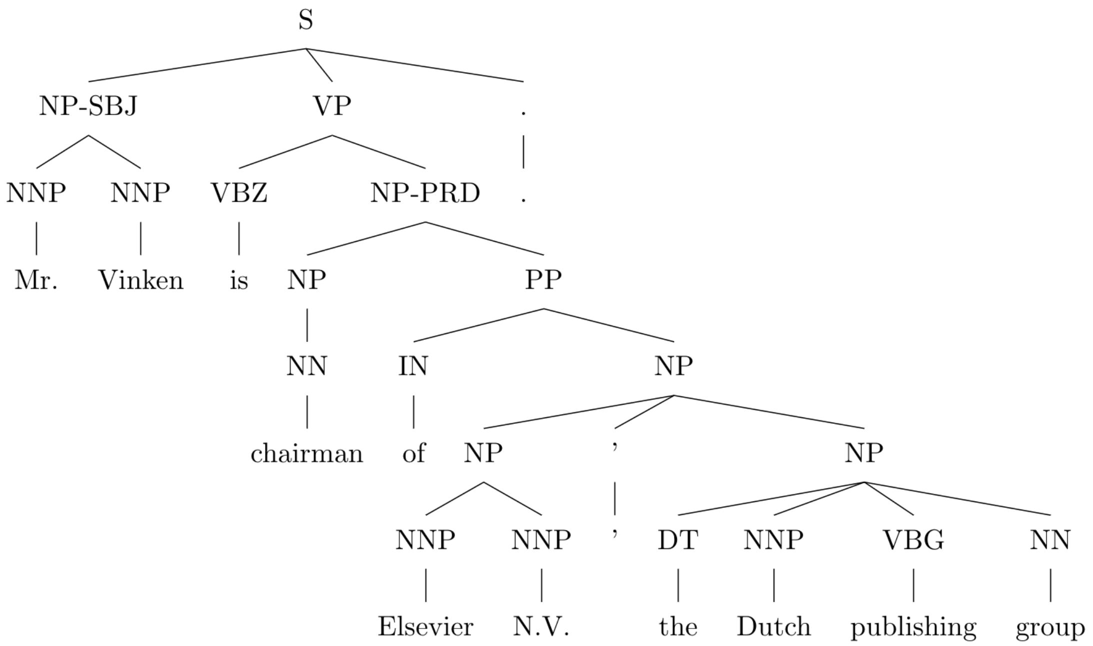
- Supervised machine learning needs labeled datasets, where labels are called ground truth
- In NLP, most labels are annotations provided by humans

- There is always some disagreement among annotators, even for simple tasks
- These annotations are called a gold standard, not ground truth

How are NLP datasets developed?

- Option 1: Paid, trained human annotators
  - Traditional approach
  - Researchers write annotation guidelines, recruit & pay the annotators
  - More consistent annotations, but costly to scale

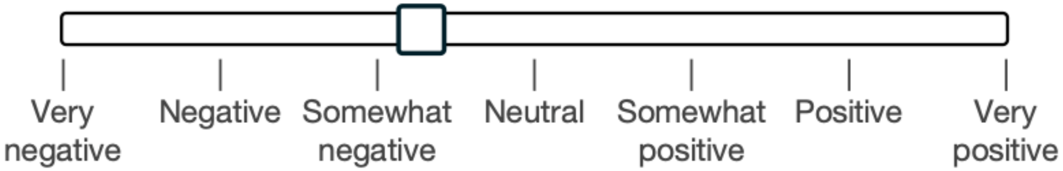
  - Example: Penn Treebank (1993)

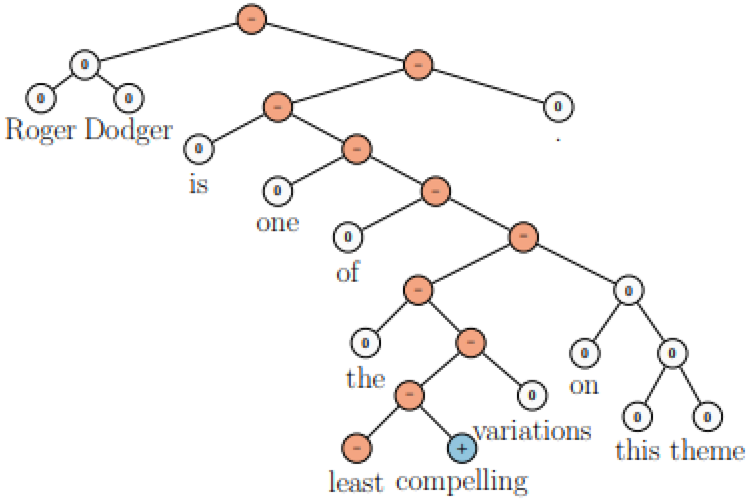


How are NLP datasets developed?

- Option 2: Crowdsourcing (e.g., Amazon Mechanical Turk)
  - More recent trend
  - Can’t really train annotators, but easier to get multiple annotations for each input (which can then be averaged)

  - Example: Stanford Sentiment Treebank





Ethics in Crowdsourcing

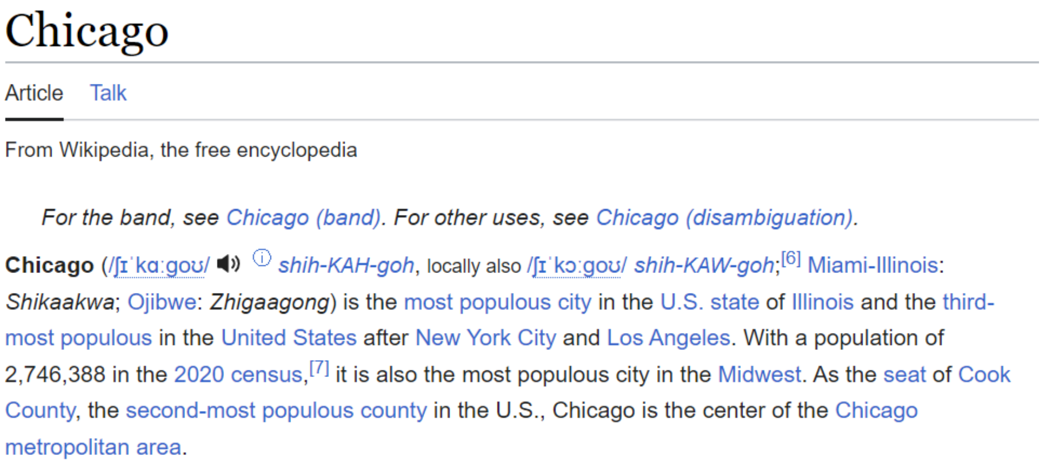
- A few questions to think about
  - Will you exclude some participants based on some characteristics?
  - Will the participants interact with each other?
  - How will the participants be paid?
  - Will you collect more data than needed?
  - How will the data be stored?

  …

How are NLP datasets developed?

- Option 3: Naturally-occurring annotation
  - Doesn’t require human annotation for the specific purpose
  - Could be noisy

- Example: named entity recognition



Chicago

Article Talk

From Wikipedia, the free encyclopedia

For the band, see Chicago (band). For other uses, see Chicago (disambiguation).

Chicago (/ʃɪˈkɑːɡoʊ/ 🔊 ⓘ *shih-KAH-goh*, locally also /ʃɪˈkɔːɡoʊ/ *shih-KAW-goh*;[6] Miami-Illinois: *Shikaakwa*; Ojibwe: *Zhigaagong*) is the most populous city in the U.S. state of Illinois and the third-most populous in the United States after New York City and Los Angeles. With a population of 2,746,388 in the 2020 census,[7] it is also the most populous city in the Midwest. As the seat of Cook County, the second-most populous county in the U.S., Chicago is the center of the Chicago metropolitan area.

Annotator Agreement

- Given annotations from two annotators, how should we measure inter-annotator agreement?
- Agreement percentage

$$p_o = \frac{\sum_{i=1}^{n} \mathbb{1}\left[a_i = b_i\right]}{n}$$

n: number of examples

Annotator Agreement

- Given annotations from two annotators, how should we measure inter-annotator agreement?
- Agreement percentage
- Cohen's Kappa (Cohen, 1960) accounts for agreement by chance

$$\kappa = \frac{p_o - p_e}{1 - p_e}$$

p_e: agreement by chance

A\B	Y	N
Y	80	5
N	5	10

$$p_A(Y) = 0.85$$
$$p_A(N) = 0.15$$
$$p_B(Y) = 0.85$$
$$p_B(N) = 0.15$$

$$p_e = p_A(Y)p_B(Y) + p_A(N)p_B(N) = 0.745$$
$$p_o = 0.9$$
$$\kappa = \frac{0.9 - 0.745}{1 - 0.745} = 0.608$$

Cohen’s Kappa (cont.)

$$\kappa = \frac{p_o - p_e}{1 - p_e}$$

A\B	Y	N
Y	80	5
N	5	10

$$\kappa = \frac{0.9 - 0.745}{1 - 0.745} = 0.608$$

A\B	Y	N
Y	45	5
N	5	45

$$p_A(Y) = 0.5$$
$$p_A(N) = 0.5$$
$$p_B(Y) = 0.5$$
$$p_B(N) = 0.5$$

$$p_e = p_A(Y)p_B(Y) + p_A(N)p_B(N) = 0.5$$

$$p_o = 0.9$$

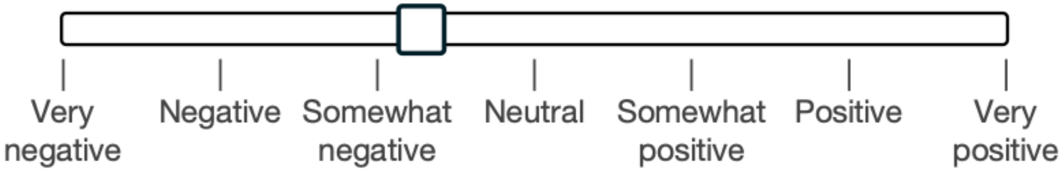
$$\kappa = \frac{0.9 - 0.5}{1 - 0.5} = 0.8$$

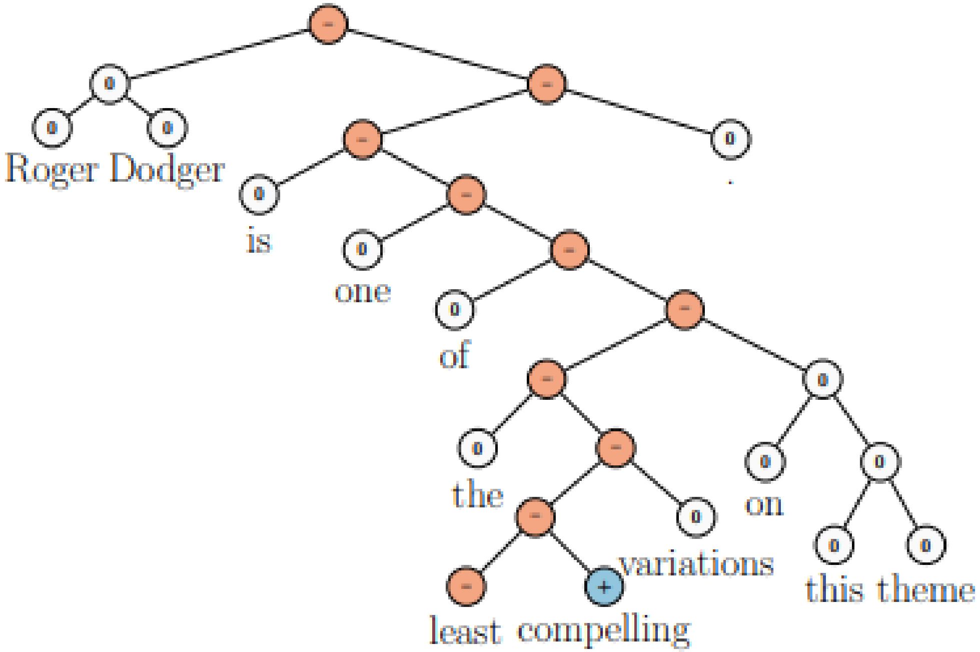
Annotator Agreement

- Given annotations from two annotators, how should we measure inter-annotator agreement?
- Agreement percentage
- Cohen’s Kappa (Cohen, 1960) accounts for agreement by chance
- Generalization exists for more than two annotators (Fleiss, 1970)

Data for Text Classification

- Stanford sentiment treebank: fine-grained sentiment analysis of movie reviews





Data for Text Classification

- Stanford sentiment treebank: fine-grained sentiment analysis of movie reviews
- Subjectivity/objectivity sentence classification

Subjectivity/Objectivity

the hulk is an anger fueled monster with incredible strength and resistance to damage .	Objective
in trying to be daring and original , it comes off as only occasionally satirical and never fresh .	Subjective
solondz may well be the only one laughing at his own joke	
obstacles pop up left and right , as the adventure gets wilder and wilder .	

Subjectivity/Objectivity

the hulk is an anger fueled monster with incredible strength and resistance to damage .	Objective
in trying to be daring and original , it comes off as only occasionally satirical and never fresh .	Subjective
solondz may well be the only one laughing at his own joke	Subjective
obstacles pop up left and right , as the adventure gets wilder and wilder .	Objective

- How was this dataset generated?
  - IMDB plot summaries: objective
  - Rotten Tomatoes snippets: subjective
- Be mindful to potential bias: movie reviews ≠ subjective wordings

Data for Text Classification

- Stanford sentiment treebank: fine-grained sentiment analysis of movie reviews
- Subjectivity/objectivity sentence classification
- TREC question type classification

Question Type

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

- Help QA system identify the answer type.

Data for Text Classification

- Stanford sentiment treebank: fine-grained sentiment analysis of movie reviews
- Subjectivity/objectivity sentence classification
- TREC question type classification
- Linguistic acceptability judgment

Linguistic Acceptability

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

Linguistic Acceptability

The more books I ask to whom he will give, the more he reads.	Unacceptable
The jeweller 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**

- Understand to what extent human judgments can be predicted.
- Caveat: humans don’t even agree on some examples.

Data for Text Classification

- Stanford sentiment treebank: fine-grained sentiment analysis of movie reviews
- Subjectivity/objectivity sentence classification
- TREC question type classification
- Linguistic acceptability judgment

- **Please look at your data in experiments!**

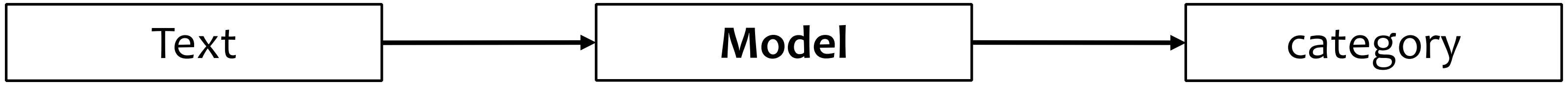
This Lecture

- NLP datasets

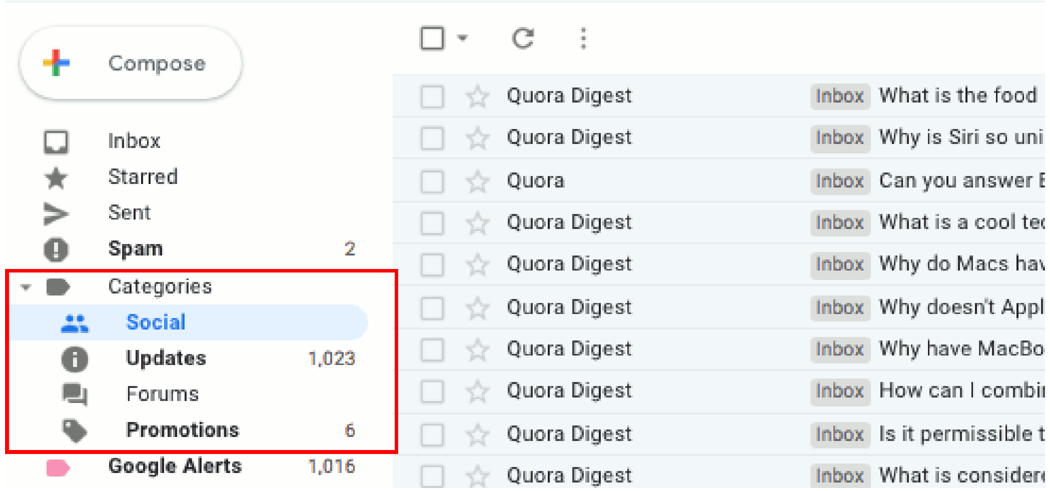
- **Text classification**
  - Rule-based methods
  - Naïve Bayes
  - Logistic regression
  - Support vector machines

What is text classification?

- Simplest user-facing NLP application

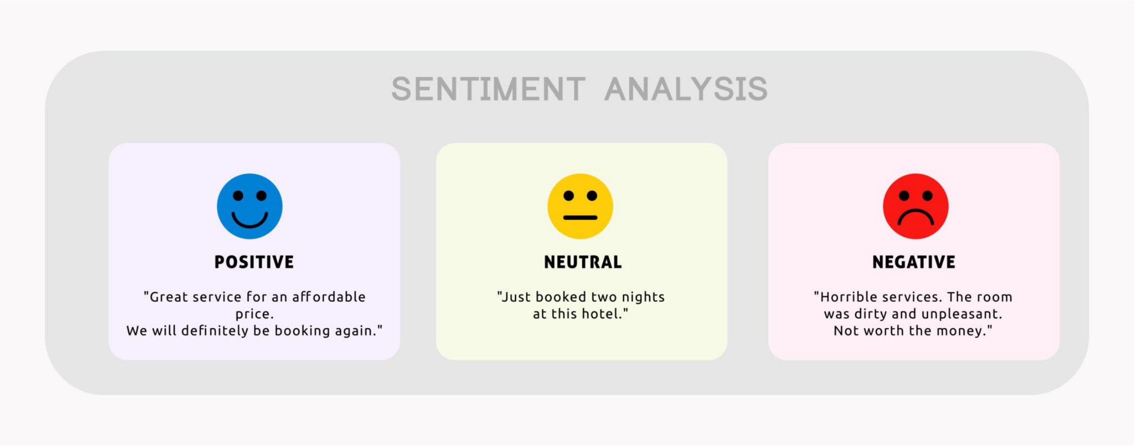


Text → **Model** → category



Inbox
Starred
Sent
Spam 2
Categories
Social
Updates 1,023
Forums
Promotions 6
Google Alerts 1,016

Quora Digest	Inbox	What is the food
Quora Digest	Inbox	Why is Siri so uni
Quora	Inbox	Can you answer E
Quora Digest	Inbox	What is a cool tec
Quora Digest	Inbox	Why do Macs hav
Quora Digest	Inbox	Why doesn't Appl
Quora Digest	Inbox	Why have MacBo
Quora Digest	Inbox	How can I combi
Quora Digest	Inbox	Is it permissible t
Quora Digest	Inbox	What is consider



SENTIMENT ANALYSIS

POSITIVE

"Great service for an affordable price.
We will definitely be booking again."

NEUTRAL

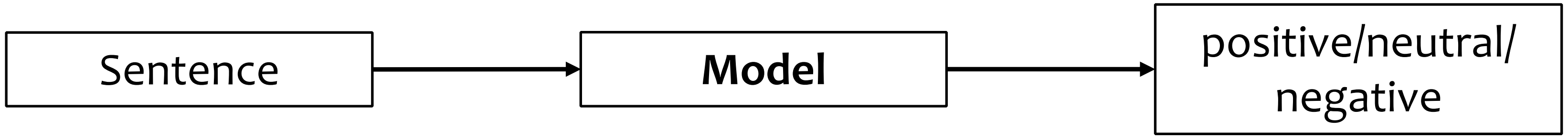
"Just booked two nights at this hotel."

NEGATIVE

"Horrible services. The room was dirty and unpleasant.
Not worth the money."

Task Formulation

- Sentiment classification



- Input
  - sentence/a list of tokens s
  - A set of categories Y
- Output
  - Predicted category $y_s \in Y$ for sentence s

Modeling: Rule-based text classification

- If s contains words in [*good, excellent, extraordinary, …*] return positive
- If s contains words in [*bad, terrible, awful, …*] return negative

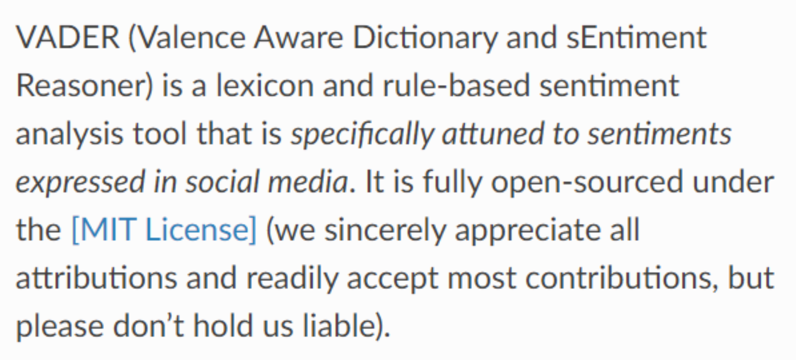
👍 Nice interpretability

👍 Can be very accurate (with carefully refined rules)

👎 Rules are difficult to define

👎 System can be very complicated

👎 Hardly generalizable



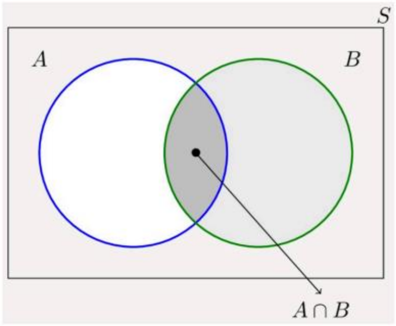
VADER (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is *specifically attuned to sentiments expressed in social media*. It is fully open-sourced under the [MIT License] (we sincerely appreciate all attributions and readily accept most contributions, but please don't hold us liable).

Modeling: Statistical Classifier

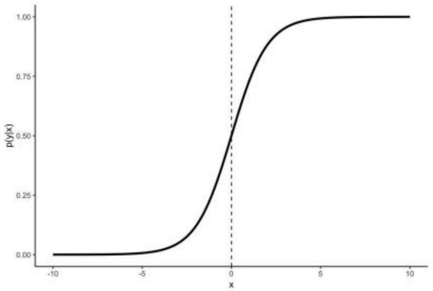
- What we have
  - A set of categories Y
  - A set of labeled sentences $\mathcal{D} = \{(s_1, y_1), (s_2, y_2), \dots (s_n, y_n)\}$
- What we want
  - Classifier $f_\Theta : \mathcal{S} \rightarrow Y$

  - Modeling: What is the form of f?
  - Training: How can we learn f?
  - Inference: How can we use f?

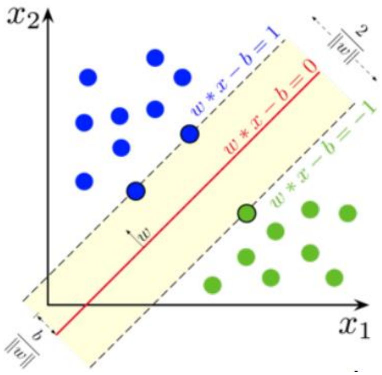
Types of Statistical Classifiers



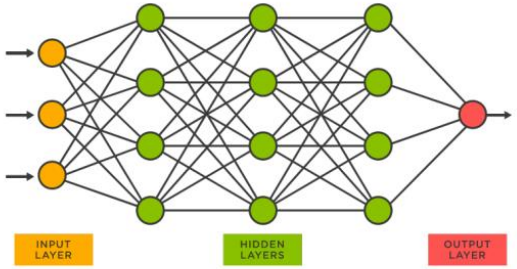
Naïve Bayes



Logistic regression



Support vector machines



Neural networks

Probabilistic Modeling for Classification

- Input: A set of labeled sentences $\mathcal{D} = \{(s_1, y_1), (s_2, y_2), \dots (s_n, y_n)\}$

- We can model the probability of label conditioned on sentence
  For sentence s, the result is

$$\arg\max_{y} P(y \mid s)$$

- What's the main challenge here?

  In NLP, text should be converted to numerical representations.

Naïve Bayes: Principle

- Simple classification model making use of Bayes’ rule

s: sentence; y: class label

$$P(y \mid s) = \frac{P(y, s)}{P(s)} = \frac{P(y)P(s \mid y)}{P(s)}$$

Definition of marginal, conditional and joint probability

Naïve Bayes: Inference

s: sentence; y: class

$$y_{\text{MAP}} = \arg\max_{y \in Y} P(y \mid s)$$

MAP: maximum a posteriori

Naïve Bayes: Inference

s: sentence; y: class

$$y_{\text{MAP}} = \arg\max_{y \in Y} P(y \mid s)$$

MAP: maximum a posteriori

$$= \arg\max_{y \in Y} \frac{P(y)P(s \mid y)}{P(s)}$$

Bayes’ rule

Naïve Bayes: Inference

s: sentence; y: class

$$y_{\text{MAP}} = \arg\max_{y \in Y} P(y \mid s)$$

MAP: maximum a posteriori

$$= \arg\max_{y \in Y} \frac{P(y)P(s \mid y)}{P(s)}$$

Bayes' rule

$$= \arg\max_{y \in Y} P(y)P(s \mid y)$$

Dropping the denominator

- What we need: $P(y), P(s \mid y)$

How should we model $P(s \mid y)$?

- What we need: $P(y), P(s \mid y)$
- Option 1: memorize the probability of all word sequences

$$s = w_1, w_2, \ldots, w_k$$
$$P(s \mid y) = P(w_1, w_2, \ldots, w_k \mid y)$$

  - There are too many sequences
  - Not generalizable: *This is a cute cat* vs. *This is a cute dog.*

How should we model $P(s \mid y)$?

- What we need: $P(y), P(s \mid y)$
- Option 1: memorize the probability of all word sequences
- Option 2: bag of words

$$
\begin{aligned}
s &= w_1, w_2, \ldots, w_k \\
P(s \mid y) &= P(w_1, w_2, \ldots, w_k \mid y) \\
&= P(w_1 \mid y) P(w_2 \mid y) \ldots P(w_k \mid y)
\end{aligned}
$$

  - Assumption: position doesn’t matter
  - Probability of each word is conditionally independent of others given class y

Bag of words features

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
…	

Bag of words features

- Caveat: Potential information loss

BoW(*a cat is drinking milk*) = BoW(*milk is drinking a cat*)

What tasks hurt the most from such information loss?

Naïve Bayes: Inference

s: sentence; y: class

$$\begin{aligned} y_{\text{MAP}} &= \arg\max_{y \in Y} P(y \mid s) \\ &= \arg\max_{y \in Y} P(y) P(s \mid y) \\ &= \arg\max_{y \in Y} P(y) \prod_{i=1}^{k} P(w_i \mid y) \end{aligned}$$

Naïve Bayes: Inference

s: sentence; y: class

$$\begin{aligned} y_{\text{MAP}} &= \arg\max_{y \in Y} P(y \mid s) \\ &= \arg\max_{y \in Y} P(y) P(s \mid y) \\ &= \arg\max_{y \in Y} P(y) \prod_{i=1}^{k} P(w_i \mid y) \\ &= \arg\max_{y \in Y} \log P(y) + \sum_{i=1}^{k} \log P(w_i \mid y) \end{aligned}$$

Naïve Bayes: Parameter Estimation

$$y_{\text{MAP}} = \arg\max_{y \in Y} P(y) \prod_{i=1}^{k} P(w_i \mid y)$$

- What we need: $P(y), P(s \mid y)$
- What we have: a set of labeled sentences $\mathcal{D} = \{(s_1, y_1), \ldots, (s_n, y_n)\}$

$$\hat{P}(y_j) = \frac{\text{count}(y_j)}{n}$$

$$\hat{P}(w_i \mid y_j) = \frac{\text{count}(w_i, y_j)}{\sum_w \text{count}(w, y_j)}$$

Naïve Bayes: Parameter Estimation

What is the time/memory complexity?

$$\hat{P}(y_j) = \frac{\text{count}(y_j)}{n} \qquad \hat{P}(w_i \mid y_j) = \frac{\text{count}(w_i, y_j)}{\sum_w \text{count}(w, y_j)}$$

Memory: $\mathcal{O}(|Y|V)$ -- |Y|: number of classes

-- V: number of **word types** (vocabulary size)

Time: $\mathcal{O}(W)$ -- W: number of **word tokens** in the training data

Word token vs. word type:

{(*the cat is cute, 1*), (*the snake is not cute, 0*)}

# word tokens: 9 # word types: 6

Data Sparsity

$$y_{\text{MAP}} = \arg\max_{y \in Y} P(y) \prod_{i=1}^{k} P(w_i \mid y)$$

If word *fantastic* doesn't exist in positive training examples, but appears once in negative training examples.

$$\text{count}(\textit{fantastic, positive}) = 0 \;\; \Rightarrow P(\textit{fantastic} \mid \textit{positive}) = 0$$

$$\text{count}(\textit{fantastic, negative}) = 1 \;\; \Rightarrow P(\textit{fantastic} \mid \textit{negative}) > 0$$

$P(s \mid \textit{positive}) = 0$ for all s containing *fantastic*, in the inference stage.

Addressing Data Sparsity: Laplace Smoothing

- Original:

$$P(w_i \mid y_j) = \frac{\text{count}(w_i, y_j)}{\sum_w \text{count}(w, y_j)}$$

- Smoothed:

$$P(w_i \mid y_j) = \frac{\text{count}(w_i, y_j) + \alpha}{\sum_w \text{count}(w, y_j) + \alpha|V|} \quad (\alpha > 0)$$

Naïve Bayes: Overall Process

- Input (training data): labeled sentences $\mathcal{D} = \{(s_1, y_1), \ldots (s_n, y_n)\}$
- Step 1: compute vocabulary V

Training (parameter estimation)

- Step 2: calculate $\hat{P}(y_j) = \frac{\text{count}(y_j)}{n}$ for each y_j
- Step 3: calculate $\hat{P}(w_i \mid y_j) = \frac{\text{count}(w_i, y_j)}{\sum_w \text{count}(w, y_j)}$ for each w_i, y_j

Inference (prediction)

- Step 4: $y_{\text{MAP}} = \arg\max_{y \in Y} P(y) \prod_{i=1}^{k} P(w_i \mid y)$

Naïve Bayes: Example

- **Prior from training set:**

$$P(+) = \frac{2}{5}, P(-) = \frac{3}{5}$$

- **$|V| = 20$ from training set**
- **Conditional probabilities $(\alpha = 1)$:**

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

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

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

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

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

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

Type	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

- **Scoring the examples**

$$P(+)P(\text{predictable} \mid +)P(\text{no} \mid +)P(\text{fun} \mid +) = 3.2 \times 10^{-5}$$

$$P(-)P(\text{predictable} \mid -)P(\text{no} \mid -)P(\text{fun} \mid -) = 6.1 \times 10^{-5}$$

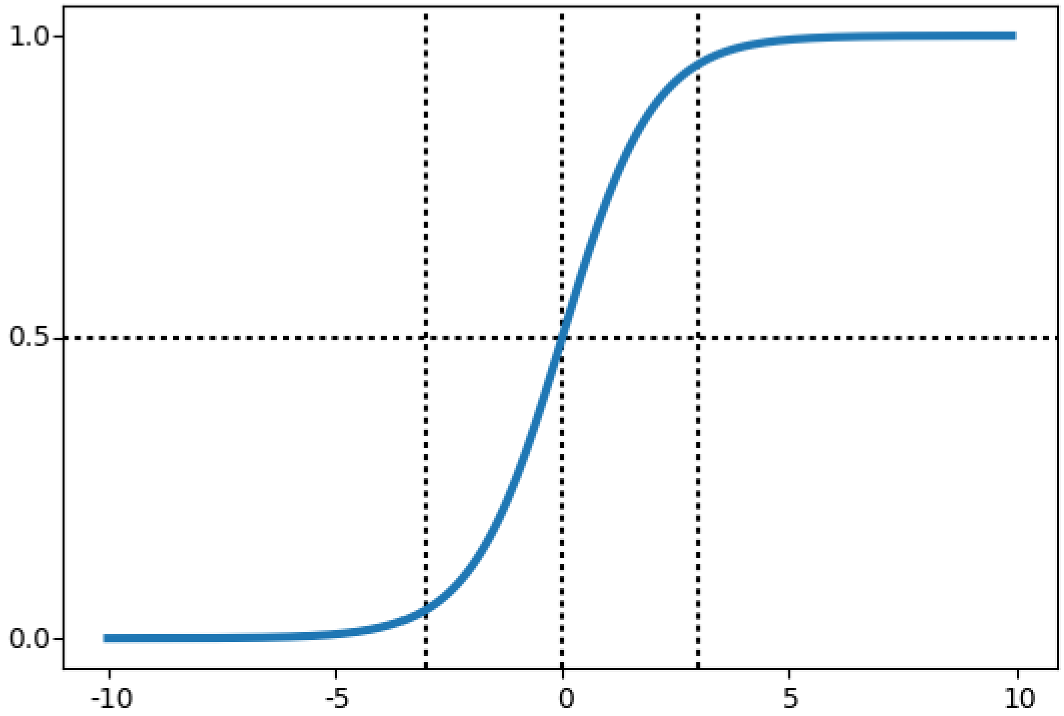
Logistic Regression

- Logistic function: $\sigma(x) = \dfrac{1}{1+e^{-x}}$

- Why logistic function?

$$\sigma : \mathbb{R} \to (0, 1)$$

Naturally models the probability for binary classification



Logistic Regression

- Logistic function: $\sigma(x) = \dfrac{1}{1+e^{-x}}$

- Suppose we can represent each sentence with a vector $\mathbf{x}$
  - How can we do this?

    Word counts, sum/average of word vectors, or more complicated features

$$P(y=1 \mid \mathbf{x}) = \sigma(\mathbf{w} \cdot \mathbf{x} + b)$$
$$P(y=0 \mid \mathbf{x}) = 1 - P(y=1 \mid \mathbf{x})$$

$\mathbf{w} \cdot \mathbf{x} + b$ can be expressed as $\mathbf{w} \cdot \mathbf{x}$ if $\mathbf{x}$ has a constant dimension.

Logistic Regression

- Input: A set of labeled sentences $\mathcal{D} = \{(s_1, y_1), (s_2, y_2), \ldots (s_n, y_n)\}$
- Feature engineering: $s_i \rightarrow \mathbf{x}_i$
- Probability of one example

$$P(y_i \mid \mathbf{x}_i; \mathbf{w}) = \begin{cases} \sigma(\mathbf{w} \cdot \mathbf{x}_i) & \text{if } y_i = 1 \\ 1 - \sigma(\mathbf{w} \cdot \mathbf{x}_i) & \text{if } y_i = 0 \end{cases}$$

- Assuming independence of examples, the dataset probability is

$$\prod_{i=1}^{n} \sigma(\mathbf{w} \cdot \mathbf{x}_i)^{y_i} \cdot (1 - \sigma(\mathbf{w} \cdot \mathbf{x}_i))^{1-y_i}$$

Logistic Regression

- Loss: take the negative logarithm of the probability

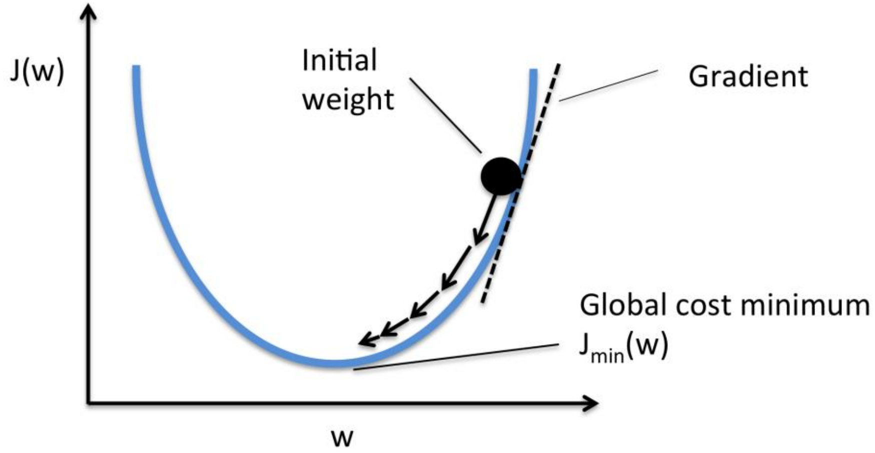
$$J(\mathbf{w}) = -\sum_{i=1}^{n} y_i \log \sigma(\mathbf{w} \cdot \mathbf{x}_i) + (1 - y_i) \log(1 - \sigma(\mathbf{w} \cdot \mathbf{x}_i))$$

- Our goal: estimate **w** to minimize the above loss (maximize likelihood)

- Method: gradient descent

$$\mathbf{w} = \mathbf{w} - \eta \cdot \frac{\partial J}{\partial \mathbf{w}}$$

η: learning rate



- Convex function:

$$\forall x_1, x_2 \in X, 0 \leq t \leq 1, f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2)$$

Logistic Regression

- What if there are more than 2 classes?

  1 vs. 1 for $\frac{|Y|\times(|Y|-1)}{2}$ class pairs and do voting

  1 vs. all for $|Y|$ classes and do argmax

- Probability interpretations (over classes) no longer hold

Generative vs. Discriminative Model

- Generative: $P(s, y) = P(\, s \mid y \,)P(y)$ – Naïve Bayes
- Discriminative: $P(y \mid s)$ or generally $score(y \mid s)$ – logistic regression

What are the differences?

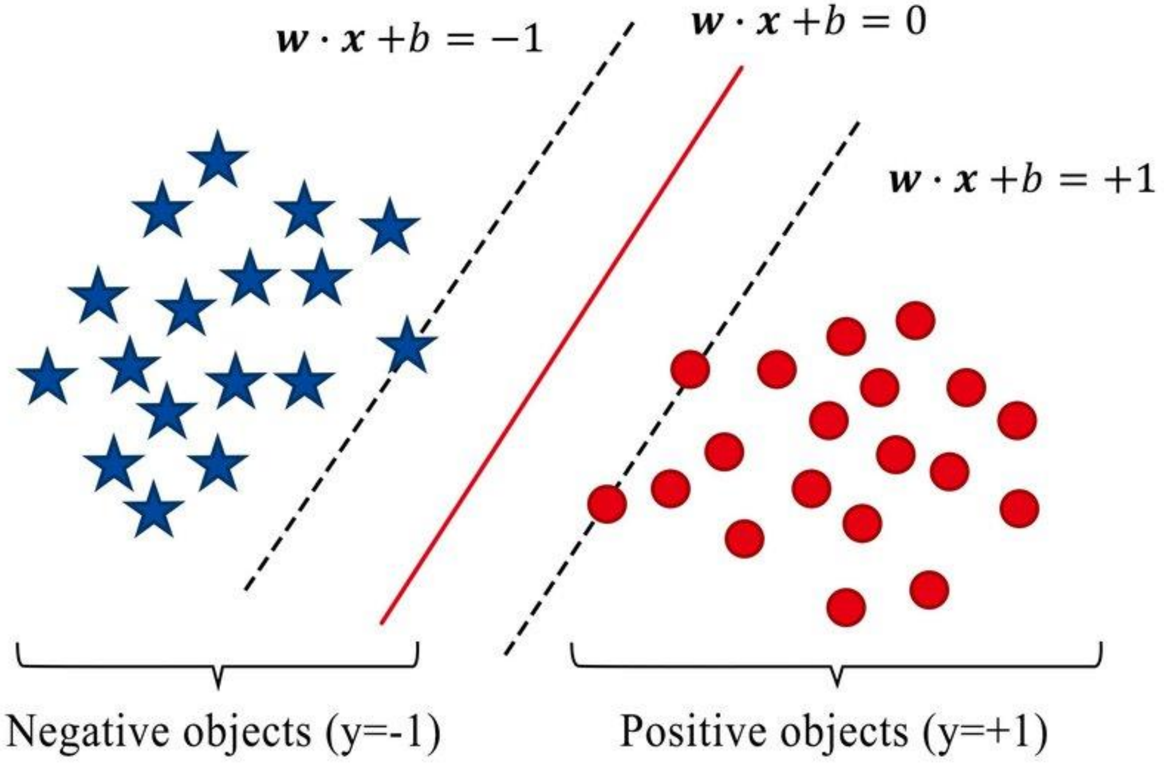
Support Vector Machines

- Input: A set of labeled sentences
  $\mathcal{D} = \{(s_1, y_1), (s_2, y_2), \dots (s_n, y_n)\}$

- Feature engineering $s_i \rightarrow \mathbf{x}_i$

- Support vectors: The data points at the forefront of a class closest to the opposite class.

- Find an optimal decision boundary that maximizes the distance between support vectors


$\boldsymbol{w} \cdot \boldsymbol{x} + b = -1$
$\boldsymbol{w} \cdot \boldsymbol{x} + b = 0$
$\boldsymbol{w} \cdot \boldsymbol{x} + b = +1$
Negative objects (y=-1)
Positive objects (y=+1)

Support Vector Machines

- We are interested in a large margin classifier

$$\arg\max_{\mathbf{w},b}\left\{\frac{1}{\|\mathbf{w}\|}\min_{i} y_i(\mathbf{w}\cdot\mathbf{x}_i+b)\right\}$$

$y_i \in \{-1, +1\}$ denotes the label

Support Vector Machines

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

Since we can scale $\mathbf{w}, b$ accordingly, let's assume the margin is 1, i.e.,

$$\min_i y_i(\mathbf{w}\cdot\mathbf{x}_i+b)=1$$

The problem turns to

$$\arg\min_{\mathbf{w}} ||\mathbf{w}||^2 \qquad \text{s.t. } y_i(\mathbf{w}\cdot\mathbf{x}_i+b)\geq 1, \forall i=1,\ldots,n$$

Support Vector Machines

$$\arg\min_{\mathbf{w}} ||\mathbf{w}||^2 \qquad \text{s.t. } y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1, \forall i = 1, \ldots, n$$

Representer theorem: the solution to the above problem can be represented as

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

Proof idea:

Let $\mathbf{w}^* = \mathbf{w}_X + \mathbf{w}_\perp$, where $\mathbf{w}_X \in \text{span}(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$, and $\mathbf{w}_\perp \cdot \mathbf{w}_X = 0$.

If $\mathbf{w}_\perp \neq \mathbf{0}$, we will be able to find a smaller $||\mathbf{w}||^2$.

SVM: Non-Separable Data

$$\arg\min_{\mathbf{w}} ||\mathbf{w}||^2 \qquad \text{s.t. } y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1, \forall i = 1, \ldots, n$$

- Slack “variables” ξ_i to handle non-separable data:

$$y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1 + \xi_i \geq 0, \qquad \xi_i \geq 0$$

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

- Minimize SVM loss with gradient descent

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

SVM: Non-Separable Data

- Minimize SVM loss with gradient descent

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

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

- Not differentiable when $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) = 1$?
- Solution: subgradient descent (next lecture).

SVM: Readings & Background

PRML: Christopher M. Bishop, Pattern Recognition and Machine Learning. Chapter 7.1

Tips: When you see kernel in the book, just think about the dot product between two vectors -- the simplest kernel in most cases.

Next Lecture

- Text classification: general formulation, features, and learning