CS4705

Probability Review and
Naïve Bayes

Slides from Dragomir Radev and modified

Announcements

- Reading for today: C. 4, 4.5 NLP
- Reading for next class: C 3, NLP
- Next class will be taught by Chris Kedzie
- For new students in class:
  - No laptop policy
  - Class participation using PollEverywhere or in-class comments

Today

- SciKit Learn Tutorial
- Wrap up on optimization
- Generative methods

Regularization

- Consider the case where one or more documents are mis-labeled
  - Text from a novel may be mis-labeled as social media if posted as a quote
- The classifier will attempt to learn weights that promote words characteristic of novels as predictors of social media
- Overfitting can also occur when the social media documents in the training set are not representative

Loss

- To prevent overfitting, a regularization parameter R(Θ) is added:

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmin}} \left(\overbrace{\frac{1}{n} \sum_{i=1}^{n} L(f(\mathbf{x}_i; \Theta), \mathbf{y}_i)}^{loss} \quad + \quad \overbrace{\lambda R(\Theta)}^{regularization} \right)$$

Two Common regularizers

- L_2 regularization
  - Keeps sum of squares of parameter values low

$$R_{L_2}(\mathbf{W}) = ||\mathbf{W}||_2^2 = \sum_{i,j} (\mathbf{W}_{[i,j]})^2$$

  - Gaussian prior or weight decay (Here W is weights not including b)
  - Prefers to decrease parameter with high weight by 1 than 10 parameters with low weights
- L_1 regularization
  - Keeps sum of absolute value of parameters low

$$R_{L_1}(\mathbf{W}) = ||\mathbf{W}||_1 = \sum_{i,j} |\mathbf{W}_{[i,j]}|$$

Punished uniformly for high and low values

Gradient based optimization

- Repeat until L (Loss) < margin
  - Compute L over the training set
  - Compute gradients of Θ with respect to L
  - Move the parameters in the opposite direction of the gradient

Stochastic Gradient Descent

Algorithm 1 Online Stochastic Gradient Descent Training

Input:

- Function $f(\mathbf{x};\Theta)$ parameterized with parameters Θ.
- Training set of inputs $\mathbf{x_1},\ldots,\mathbf{x_n}$ and desired outputs $\mathbf{y_1},\ldots,\mathbf{y_n}$.
- Loss function L.

while stopping criteria not met **do**
    Sample a training example $\mathbf{x_i},\mathbf{y_i}$
    Compute the loss $L(f(\mathbf{x_i};\Theta),\mathbf{y_i})$
    $\hat{\mathbf{g}} \leftarrow$ gradients of $L(f(\mathbf{x_i};\Theta),\mathbf{y_i})$ w.r.t θ
    $\Theta \leftarrow \Theta - \eta_t\hat{\mathbf{g}}$
return Θ

Problem

- Error is calculated based on just one training sample
- May not be representative of corpus wide loss
- Instead calculate the error based on a set of training examples: *minibatch*
- -> Minibatch stochastic gradient descent

Computing Gradients

$$\frac{\partial L}{\partial \mathbf{b}_{[i]}} = \begin{cases} -1 & i = t \\ 1 & i = k \\ 0 & \text{otherwise} \end{cases}$$

$$\frac{\partial L}{\partial \mathbf{W}_{[i,j]}} = \begin{cases} \frac{\partial(-\mathbf{x}_{[i]} \cdot \mathbf{W}_{[i,t]})}{\partial \mathbf{W}_{[i,t]}} = -\mathbf{x}_{[i]} & j = t \\ \frac{\partial(\mathbf{x}_{[i]} \cdot \mathbf{W}_{[i,k]})}{\partial \mathbf{W}_{[i,k]}} = \mathbf{x}_{[i]} & j = k \\ 0 & \text{otherwise} \end{cases}$$

Summary

- Smoothing helps to account for zero valued n-grams
- Text classification using feature vectors representing n-grams and other properties
- Discriminative learning
- Methods for optimization, loss functions and regularization

Classification using a Generative Approach

- Start with Naïve Bayes and Maximum Likelihood Expectation

- But we need some background in probability first

Probabilities in NLP

- Very important for language processing
- Example in speech recognition:
  - "recognize speech" vs "wreck a nice beach"
- Example in machine translation:
  - "l'avocat general": "the attorney general" vs. "the general avocado"
- Example in information retrieval:
  - If a document includes three occurrences of "stir" and one of "rice", what is the probability that it is a recipe
- Probabilities make it possible to combine evidence from multiple sources systematically

Probabilities

- Probability theory
  - predicting how likely it is that something will happen
- Experiment (trial)
  - e.g., throwing a coin
- Possible outcomes
  - heads or tails
- Sample spaces
  - discrete (number of “rice”) or continuous (e.g., temperature)
- Events
  - Ω is the certain event
  - $\varnothing$ is the impossible event
  - event space - all possible events

Sample Space

- Random experiment: an experiment with uncertain outcome
  - e.g., flipping a coin, picking a word from text
- Sample space: all possible outcomes, e.g.,
  - Tossing 2 fair coins, Ω ={HH, HT, TH, TT}

Events

- Event: a subspace of the sample space
  - $E \subseteq \Omega$, E happens iff outcome is in E, e.g.,
    - E={HH} (all heads)
    - E={HH,TT} (same face)
- Probability of Event : $0 \leq P(E) \leq 1$, s.t.
  - $P(\Omega)=1$ (outcome always in Ω)
  - $P(A \cup B)=P(A)+P(B)$, if $(A \cap B)=\emptyset$ (e.g., A=same face, B=different face)

Example: Toss a Die

- Sample space: $\Omega = \{1,2,3,4,5,6\}$
- Fair die:
  - $p(1) = p(2) = p(3) = p(4) = p(5) = p(6) = 1/6$
- Unfair die: $p(1) = 0.3$, $p(2) = 0.2$, ...
- N-dimensional die:
  - $\Omega = \{1, 2, 3, 4, \ldots, N\}$
- Example in modeling text:
  - Toss a die to decide which word to write in the next position
  - Ω = {cat, dog, tiger, …}

Example: Flip a Coin

- Ω : {Head, Tail}
- Fair coin:
  - p(H) = 0.5, p(T) = 0.5
- Unfair coin, e.g.:
  - p(H) = 0.3, p(T) = 0.7
- Flipping two fair coins:
  - Sample space: {HH, HT, TH, TT}
- Example in modeling text:
  - Flip a coin to decide whether or not to include a word in a document
  - Sample space = {appear, absence}

Probabilities

- Probabilities
  - numbers between 0 and 1
- Probability distribution
  - distributes a probability mass of 1 throughout the sample space Ω.
- Example:
  - A fair coin is tossed three times.
  - What is the probability of 3 heads?

Probabilities

- Joint probability: $P(A \cap B)$, also written as $P(A, B)$
- Conditional Probability: $P(A|B)=P(A \cap B)/P(B)$
  - $P(A \cap B) = P(A)P(B|A) = P(B)P(A|B)$
  - So, $P(A|B) = P(B|A)P(A)/P(B)$ (Bayes' Rule)
  - For independent events, $P(A \cap B) = P(A)P(B)$, so $P(A|B)=P(A)$
- Total probability: If $A_1, \ldots, A_n$ form a partition of S, then
  - $P(B) = P(B \cap S) = P(B, A_1) + \ldots + P(B, A_n)$
  - So, $P(A_i|B) = P(B|A_i)P(A_i)/P(B)$

    $= P(B|A_i)P(A_i)/[P(B|A_1)P(A_1)+\ldots+P(B|A_n)P(A_n)]$
  - This allows us to compute $P(A_i|B)$ based on $P(B|A_i)$

Probabilities

- Joint probability: $P(A \cap B)$, also written as $P(A, B)$
- Conditional Probability: $P(A|B)=P(A \cap B)/\boxed{P(B)}$
  - $P(A \cap B) = P(A)P(B|A) = \boxed{P(B)P(A|B)}$
  - So, $P(A|B) = \boxed{P(B|A)P(A)}/\boxed{P(B)}$ (Bayes' Rule)
  - For independent events, $P(A \cap B) = P(A)P(B)$, so $P(A|B)=P(A)$
- Total probability: If $A_1, \ldots, A_n$ form a partition of S, then
  - $P(B) = P(B \cap S) = P(B, A_1) + \ldots + P(B, A_n)$
  - So, $P(A_i|B) = P(B|A_i)P(A_i)/P(B)$

    $= P(B|A_i)P(A_i)/[P(B|A_1)P(A_1)+\ldots+P(B|A_n)P(A_n)]$
  - This allows us to compute $P(A_i|B)$ based on $P(B|A_i)$

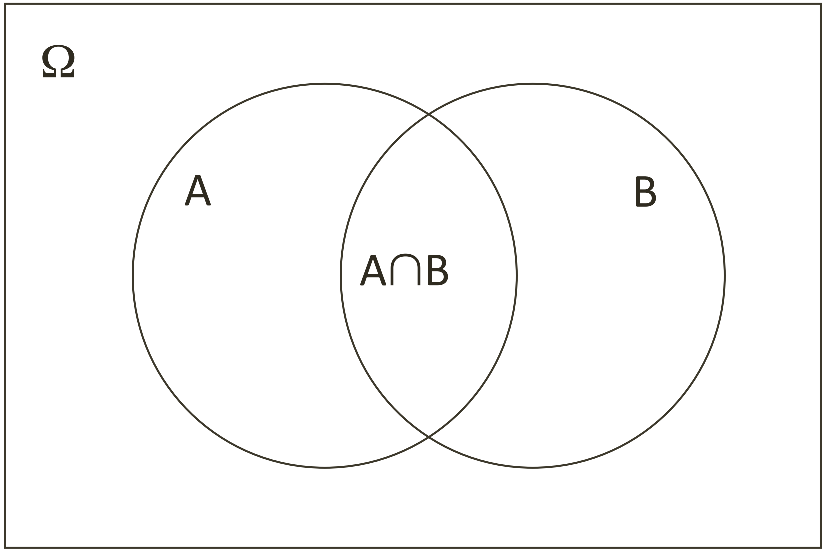
Properties of Probabilities

- $p(\emptyset) = 0$
- $P(\text{certain event})=1$
- $p(X) \leq p(Y)$, if $X \subseteq Y$
- $p(X \cup Y) = p(X) + p(Y)$, if $X \cap Y=\emptyset$

Conditional Probability

- Prior and posterior probability
- Conditional probability

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$



Conditional Probability

- Six-sided fair die
  - P(D even)=?
  - P(D>=4)=?
  - P(D even|D>=4)=?
  - P(D odd|D>=4)=?
- Multiple conditions
  - P(D odd|D>=4, D<=5)=?

P(D even) =?

1

5

.5

.3

None of the above

P(D even) =?

1

.25

.5

.3

None of the above

P (D even | D > 4)

- 2/3
- 1/2
- 1/4
- 0
- None of the above

P(D odd | D >= 4)

3/6

2/3

1/3

1/4

None of the above

P(D odd|D>=4, D<=5)=?

2/3

1/3

0/2

1/2

None of the above

Independence

- Two events are independent when

  $P(A \cap B) = P(A)P(B)$
- Unless $P(B)=0$ this is equivalent to saying that $P(A) = P(A|B)$
- If two events are not independent, they are considered dependent

Probability Theory Review

$$1 = \sum_a P(A = a)$$

Conditional Probability $$P(A|B) = \frac{P(AB)}{P(B)}$$

Chain Rule $$P(AB) = P(A|B)P(B)$$

Law of Total Probability $$P(A) = \sum_b P(A, B = b)$$

$$P(A) = \sum_b P(A|B = b)P(B = b)$$

Disjunction (Union) $$P(A \vee B) = P(A) + P(B) - P(AB)$$

Negation (Complement) $$P(\neg A) = 1 - P(A)$$

[slide from Brendan O'Connor]

Naïve Bayes Classifier

- We use Baye's rule:
  - $$P(C|D) = \frac{P(D|C)P(C)}{P(D)}$$

    Here C=Class, D=Document
- We can simplify and ignore P(D) since it is independent of class choice
  - $$P(C|D) \cong P(D|C)P(C) \cong P(C) \prod_{i=1,n} P(w_i|C)$$
  - This estimates the probability of D being in Class C assuming that D as n tokens and w is a token in D.

Use Labeled Training Data

- P(C) is equivalent to the number of labeled documents in the class / total number of documents:

  $P(C) = D_c/D$

$P(w_i|C)$ is equivalent to the number of times w_i occurs with label C / the number of times all words in the vocabulary (V) occur with label C

$$P(w_i|C) = Count(w_i C)/\sum_{v_i \varepsilon V} Count(v_i C)$$

Multinomial Naïve Bayes Independence Assumptions

$$P(w_1,\ldots w_n)$$

- Bag of Words assumption
  - Assume position doesn't matter
- Conditional Independence
  - Assume the feature probabilities $P(w_i|c)$ are independent given the class *c.*

$$P(w_1,\ldots w_n) = \prod_{i=1,n} P(w_i|C)$$

[Jurafsky and Martin]

Multinomial Naïve Bayes Classifier

- C_{MAP} = argmax $P(w_1 \ldots w_n | C)\, P(C)$

- C_{NB} = argmax $P(C_j)\, \boxed{\prod_{w \varepsilon W} P(w|C)}$

This is why it's naïve!

[Jurafsky and Martin]

Laplace Smoothing: Needed because counts may be zero

$$\hat{P}(w_i \mid c) = \frac{count(w_i, c)}{\sum_{w \in V} \left(count(w, c)\right)}$$

$$\hat{P}(w_i \mid c) = \frac{count(w_i, c) + 1}{\sum_{w \in V} \left(count(w, c) + 1\right)}$$

$$= \frac{count(w_i, c) + 1}{\left(\sum_{w \in V} count(w, c)\right) + |V|}$$

[Jurafsky and Martin]

Questions?

SciKit Learn