Metric Spaces Notes

Leon Lee

February 9, 2024

Contents

1 Section **3**
1.1 Subsection . . . 3

2 Week 3 **3**
2.1 Evaluating a Language Model . . . 3
2.2 Entropy . . . 3

3 Week 4 **3**
3.1 Bayes theorem stuff . . . 3
3.1.1 Naive Bayes . . . 3
3.1.2 Maximum Entropy . . . 4
3.1.3 MaxEnt Feature Example . . . 4
3.1.4 Classification with MaxEnt . . . 4
3.1.5 Training the model . . . 4
3.1.6 How does the gradient look like? . . . 4

1 Section

1.1 Subsection

2 Week 3

2.1 Evaluating a Language Model

Intuitively, a trigram model captures more context than a bigram model, so it should be a better model. How do we measure "better"?

Extrinsic Evaluation: measure performance on a downstream application

- For LM, plug it into a machine translation/ASR/etc. system
- The most reliable evaluation, but can be time-consuming.
- And of course, we still need an evaluation measure for the downstream system

Intrinsic Evaluation: design a measure that is inherent to the current task.

- Can be much quicker/easier during development cycle
- But not always easy to figure out what the right measure is: ideally, one that correlates well with extrinsic measures.

Let's introduce a way to define intrinsic measures for LMs

2.2 Entropy

Definition of the **entropy** of a random variable X:

$$H(X) = \sum_x -P(x) \log_2 P(x)$$

Also: the expected value of $-\log_2 P(X)$
Intuitively: it is a measure of uncertainty / disorder
Idea: If events are more likely, then the entropy is lower
Example 1: Equal odds → Medium entropy (kinda predictable)

$$\begin{aligned} P(a) &= 0.5 \\ P(b) &= 0.5 \end{aligned} \qquad \begin{aligned} H(X) &= -0.5 \log_2 0.5 - 0.5 \log_2 0.5 \\ &= -\log_2 0.5 \\ &= 1 \end{aligned}$$

Example 2: One term with higher odds → Lower entropy (more predictable)

$$\begin{aligned} P(a) &= 0.97 \\ P(b) &= 0.01 \\ P(c) &= 0.01 \\ P(d) &= 0.01 \end{aligned} \qquad \begin{aligned} H(X) &= -0.97 \log_2 0.97 - 3 \cdot (0.01 \log_2 0.01) \\ &= -0.97 \log_2 0.97 - 0.03 \log_2 0.01 \\ &= 0.25194 \end{aligned}$$

3 Week 4

3.1 Bayes theorem stuff

3.1.1 Naive Bayes

Naive Bayes assumption is naive Features are not usually independent given the class.

Adding multiple feature types (e.g. words and morphemes) often leads to even stronger correlations between features
Accuracy of classifier can sometimes still be okay, but it will be highly overconfident
A less naive model is "Maximum Entropy"

3.1.2 Maximum Entropy

Also called logistic regression
Most commonly, multinomial logistic regression

It is trained to **discriminate** correct vs incorrect values of c, given an input x. That's all it can do
Need not be probabilistic
Examples: artificial neural networks, decision trees, nearest neighbour methods, support vector machines.
Here we consider only one method: Maximum Entropy (MaxEnt) models

3.1.3 MaxEnt Feature Example

If we have three classes, our features will always come in groups of three. For example, we could have three binary features:

$$f_1 : \text{contains('ski')}\&c = 1$$
$$f_2 : \text{contains('ski')}\&c = 2$$
$$f_3 : \text{contains('ski')}\&c = 3$$

3.1.4 Classification with MaxEnt

Choose the class that has the highest probability according to

$$P(c|\vec{x}) = \frac{1}{Z}\exp\left(\sum_i w_i f_i(\bar{x}, c)\right)$$

where the nomalization constant $Z = \sum_{c'} \exp(\sum_i w_i f_i(\vec{x}, c'))$

- Inside brackets is just a dot product: $\vec{w} \cdot \vec{f}$
- $P(c|\vec{x})$ is a **monotonic function** of this dot product
- So, we will end up choosing the class for which $\vec{w} \cdot \vec{f}$ is highest

3.1.5 Training the model

Given annotated data, choose weights that make the class lavels most probable under the model
That is given examples $x^1 \dots x^N$ with labels $c^1 \dots c^N$, choose

$$\hat{w} = \text{argmax}_{\vec{w}} \sum_j \log P(c^j|x^j)$$

This is called **conditional maximum likelihood estimation** (CMLE) Like MLE, CMLE will overift, so we use tricks (regularization) to avoid that

3.1.6 How does the gradient look like?

bunch of maths lol