Natural Language Processing
CSE 325/425

Sihong Xie

<u>Lecture 3:</u>
- MLE for n-grams
- General MLE
- MAP

Estimating n-gram model using MLE

- Maximum Likelihood Estimation (MLE)
  - Let the count of word w be $c(w)$
  - Then given a corpus, we can count the number of times any word $w \in V$ appears.
  - Assumeing a unigram language model: words are selected independent of other words.
  - The likelihood of the corpus is

$$P(w_1, \ldots, w_n) \propto P(w_1) \times \cdots \times P(w_n)$$

$$\propto \prod_{w \in V} P(w)^{c(w)} = \underbrace{L(w_1, \ldots, w_n | \boldsymbol{\theta})}_{\text{likelihood}}$$

V: vocabulary

  - Parameters of the model: $\boldsymbol{\theta} = \{P(w) : w \in V\}$
  - Sufficient statistics: $\{c(w) : w \in V\}$

$$P(w) = \frac{c(w)}{n}$$

n : # of words in training corpus

$$\vec{\theta} = [\theta_1 \; \cdots \; \theta_{|V|}]$$

$$= [P(1) \; \cdots \; P(|V|)]$$

prob of seeing word 1

Estimating uni-gram model using MLE

- Maximum Likelihood Estimation (MLE)
  - By choosing the parameters $\boldsymbol{\theta} = \{P(w) : w \in V\}$ to maximize the likelihood

$$\max_{\boldsymbol{\theta}} \; L(w_1, \ldots, w_n | \boldsymbol{\theta})$$

$$\text{s.t.} \quad \sum_{w \in V} P(w) = 1$$

(unigram)

$= Pr(w_1) \times Pr(w_2) \times \cdots \times Pr(w_n)$

(bigram)

= HW 1

  - Using the Lagrangian method, you will find the MLE

$$P(w) = \frac{\text{Count of } w}{\text{Count of the total tokens}}$$

  - MLE is considered the "best" estimation of the model parameter.

"Laplacian" estimation to handle words in the V

but unseen in the corpus

Estimating n-gram model using MLE

- Maximum Likelihood Estimation (MLE)
  - Can Laplacian be derived using MLE?
  - Yes! Just add a hallucinated appearance of every word to the corpus.
  - Suppose your vocabulary is

    V = {<s>, </s>, I, am, Sam, do, not, like, green, eggs, and, ham, red}
  - Adding a hallucinated document containing each word:

The corpus = {

D0=[<s>, </s>, I, am, Sam, do, not, like, green, eggs, and, ham, red]

D1=[<s> I am Sam </s>]

D2=[<s> Sam I am </s>]

D3=[<s> I do not like green eggs and ham </s>]

}

$p(w) = ?$

LM
Prob CFG
unigram
bigram

RNN

General MLE

$$\boldsymbol{\theta} \xrightarrow{\Pr(\mathcal{D}|\boldsymbol{\theta})} \mathcal{D}$$

The corpus = { $\mathcal{D}$
D1=[<s> I am Sam </s>]
D2=[<s> Sam I am </s>]
D3=[<s> I do not like green eggs and ham </s>]
}

- Maximum Likelihood Estimation (MLE)
  - Given observed data $\mathcal{D}$ (training Data in ML)
    - can be unsupervised: a corpus
    - ~~supervised: documents labeled with sentiments {*positive, neutral, negative*}~~
  - Assume a probabilistic model generating the data
    - parametric model for likelihood of the data $\Pr(\mathcal{D}|\boldsymbol{\theta})$
  - In NLP, commonly found are multi-nomial models
    - Probabilities of seeing a word: $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_{|V|}]$ $\quad \sum_w \theta_w = 1$
    - Likelihood of a corpus using uni-gram:

$$\Pr(\mathcal{D}|\boldsymbol{\theta}) = \frac{1}{Z} \prod_{w=1}^{|V|} \theta_w^{c(w)}$$

sufficient Statistics

unknown parameter

$$1 = \sum_{D} \Pr(D|\theta)$$

How about MLE for bi-gram model?

General MLE

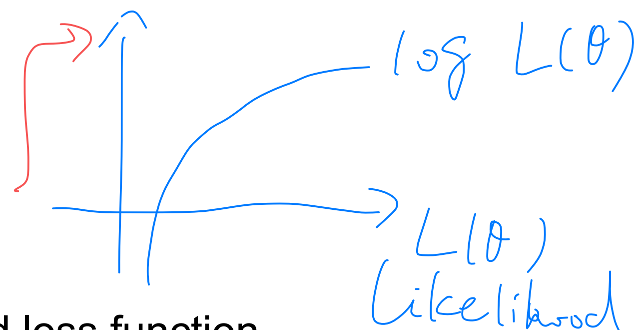
$$\boldsymbol{\theta} \xrightarrow{\Pr(\mathcal{D}|\boldsymbol{\theta})} \mathcal{D}$$

The corpus = {
D1=[<s> I am Sam </s>]
D2=[<s> Sam I am </s>]
D3=[<s> I do not like green eggs and ham </s>]
}

- Maximum Likelihood Estimation (MLE)
  - Given observed data $\mathcal{D}$
  - Parametric model for likelihood of the data $\Pr(\mathcal{D}|\boldsymbol{\theta})$ $\quad L(\boldsymbol{\theta}) = \Pr(\mathcal{D}|\boldsymbol{\theta})$
  - What's the most likely $\boldsymbol{\theta}$ that generates the data?
  - MLE is the constrained optimization problem

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \underbrace{\log L(\boldsymbol{\theta})}_{\text{log-likelihood}}$$

$$\text{s.t.} \quad \sum_w \theta_w = 1$$



  - Equivalent to minimizing the negative log-likelihood loss function

$$-\log L(\boldsymbol{\theta}) = J(\boldsymbol{\theta})$$

$$\theta^* = \arg\min_{\theta} -\log L(\theta) = \arg\min_{\theta} J(\theta)$$

optimized using gradient descent in general

General MLE

$$\boldsymbol{\theta} \xrightarrow{\Pr(\mathcal{D}|\boldsymbol{\theta})} \mathcal{D}$$

The corpus = {
D1=[<s> I am Sam </s>]
D2=[<s> Sam I am </s>]
D3=[<s> I do not like green eggs and ham </s>]
}

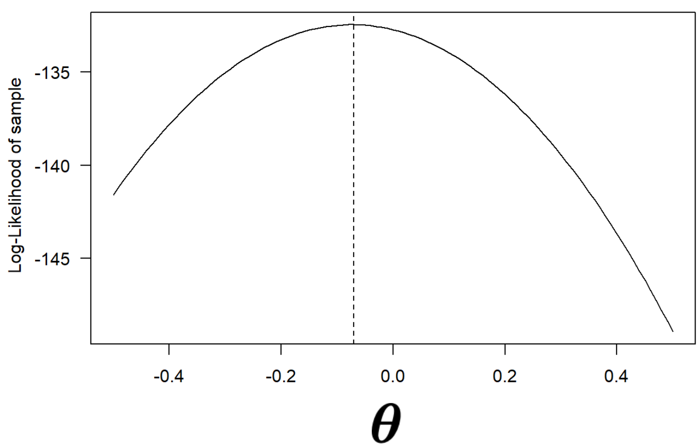
- Revisiting MLE for uni-gram model

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \left[\frac{1}{Z} \prod_{w=1}^{|V|} \theta_w^{c(w)} \right] = \arg\max_{\boldsymbol{\theta}} - \log Z + \sum_{w=1}^{|V|} c(w) \log \theta_w$$

objective function

$$J(\theta) = \arg\min_{\theta} - \sum_{w=1}^{|V|} c(w) \log \theta_w$$

s.t. $\sum_w \theta_w = 1$



But how to solve this problem?

Need two more tools:

- gradient descent
- matrix calculus
- Lagrangian methods

Alg GD

Initialize $\theta = \theta^{(0)}$ randomly

for $t = 1 \dots T$

$$\theta^{(t)} \leftarrow \theta^{(t-1)} - \eta \times \frac{\partial J(\theta)}{\partial \theta}\Big|_{\theta=\theta^{(t-1)}}$$

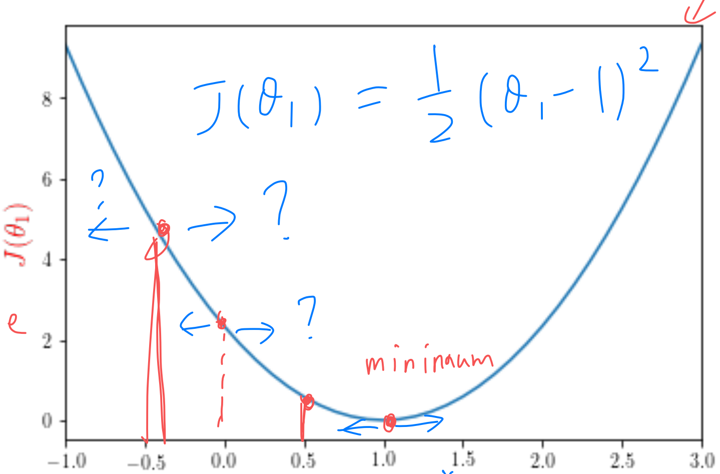
?

Gradient descent (minimal)

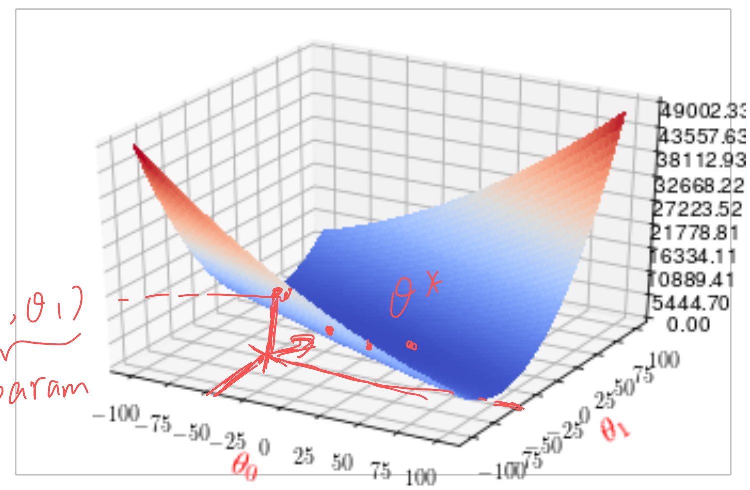
- The loss function is a scalar function of the vector $\boldsymbol{\theta}$

no a loss in NLP

When there is just one parameter



When there are two parameters



$J(\theta_0, \theta_1)$

2 param

negative log likelihood loss

$\theta^{(0)}$ $\theta^{(1)}$ $\theta^{(2)}$ θ^*

param

$$\frac{\partial J}{\partial \theta}\Big|_{\theta=\theta^*} = 0$$

$$\frac{\partial J}{\partial \theta}\Big|_{\theta=\theta^{(0)}} = (\theta_1 - 1)\Big|_{\theta_1=\theta^{(0)}} = (-0.5 - 1) = -1.5$$

$$\theta^{(1)} = \theta^{(0)} - \eta \times \frac{\partial J}{\partial \theta}\Big|_{\theta=\theta^{(0)}}$$

η = learning rate $\in (0, 0.1)$

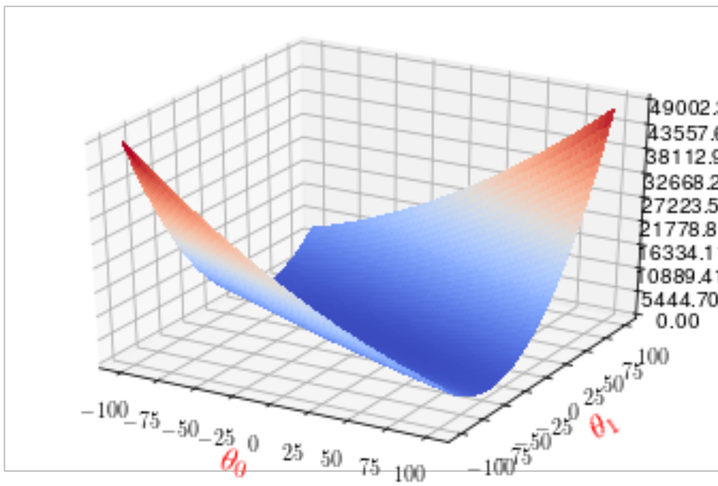
Matrix calculus (minimal)

unigram LM

$\theta = [\theta_1 \cdots \theta_{|V|}]$

- How to find the gradient of a scalar function with respect to a vector $\boldsymbol{\theta}$

When there are two parameters



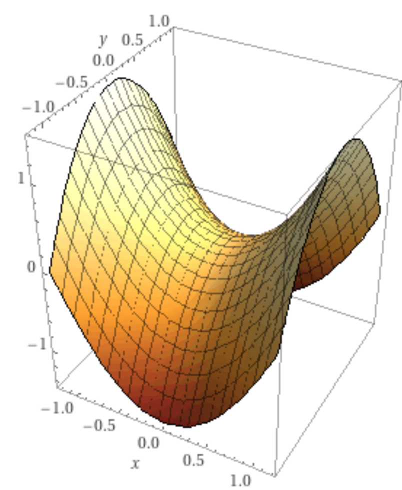
Multi-variate function:

$$f(\mathbf{x}) = f(x_1, \ldots, x_n)$$

$$f(x, y) = x^2 - y^2$$

Gradient:

$$\nabla_{\mathbf{x}} f = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix} \in \mathbb{R}^n$$



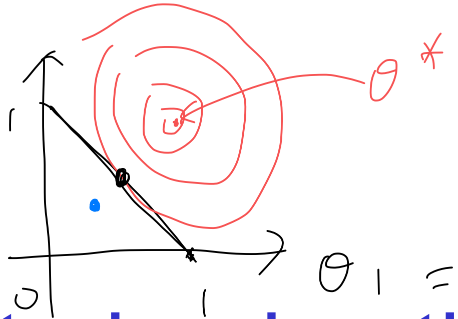
$\vec{x} \in \mathbb{R}^n$

$$\nabla_{(x,y)} f(x,y) \begin{bmatrix} \frac{\partial f(x,y)}{\partial x} \\ \frac{\partial f(x,y)}{\partial y} \end{bmatrix}$$

HW1

$\theta_{'the'} = \theta_2$

$V = \{'a', 'the'\}$



$\theta_1 = \theta_{'a'}$

$\begin{cases} \theta_1 + \theta_2 = 1 \\ \theta_1, \theta_2 \geq 0 \end{cases}$

$\min_{\theta} \max_{\lambda} L(\theta, \lambda)$

Constrained optimization

(used in estimating HMM parameters)

- Maximum Likelihood Estimation (MLE)

$$\theta^* = \arg\min_{\theta} -\frac{1}{Z}\prod_{w=1}^{|V|}\theta_w^{c(w)} = \arg\min_{\theta} \underbrace{\log Z - \sum_{w=1}^{|V|} c(w)\log\theta_w}_{J(\theta)}$$

$$\theta = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_{|V|} \end{bmatrix}$$

s.t. $\sum_w \theta_w = 1$

Take the gradient of the objective function with respect to $\boldsymbol{\theta}$

$$= \left[-\frac{c_1}{\theta_1}, -\frac{c_2}{\theta_2}, \ldots, -\frac{c_{|V|}}{\theta_{|V|}}\right]$$

$$\nabla_\theta J(\theta) = \left[\frac{\partial J}{\partial \theta_1} \cdots \frac{\partial J}{\partial \theta_{|V|}}\right] = \left[0 - c(1)\frac{1}{\theta_1}, \ldots, 0 - c(|V|)\frac{1}{\theta_{|V|}}\right]$$

Can you descend to the optimal $\boldsymbol{\theta}^*$? Don't forget the constraint. Need the Lagrangian method.

$$\mathcal{L}(\theta, \lambda) = \log Z - \sum_{w=1}^{|V|} c(w)\log\theta_w - \lambda\left[\sum_w \theta_w - 1\right]$$

$$\frac{\partial \mathcal{L}}{\partial \theta_w} = -\frac{c(w)}{\theta_w} - \lambda = 0 \Rightarrow \theta_w = -\frac{c(w)}{\lambda}$$

MLE of unigram $\Rightarrow \theta_w = \frac{c(w)}{\sum_u c(u)}$

$$\sum_{w=1}^{|V|}\theta_w = 1 \qquad \Rightarrow -\sum_{w=1}^{|V|}\frac{c(w)}{\lambda} = 1 \Rightarrow \lambda = -\sum_{w=1}^{|V|} c(w)$$

MAP

$$\boldsymbol{\alpha} \xrightarrow{P(\theta|\alpha)} \boldsymbol{\theta} \xrightarrow{\Pr(\mathcal{D}|\boldsymbol{\theta})} \mathcal{D}$$

The corpus = {
D1=[<s> I am Sam </s>]
D2=[<s> Sam I am </s>]
D3=[<s> I do not like green eggs and ham </s>]
}

- MAP (Maximum a Posterior) estimation
  - Assume a probabilistic model generating the data
    - parametric model for likelihood of the data $\Pr(\mathcal{D}|\boldsymbol{\theta})$ ← multinomial dist
    - can have prior distribution over the parameter $\Pr(\boldsymbol{\theta}|\boldsymbol{\alpha})$
  - Use the Bayes theorem to find the posterior

$$\Pr(\boldsymbol{\theta}|\mathcal{D}) = \frac{\Pr(\mathcal{D}|\boldsymbol{\theta}) \times \Pr(\boldsymbol{\theta}|\boldsymbol{\alpha})}{\Pr(\mathcal{D})}$$

α_w : Hallucinated appearance of word w

  - For convenience, the prior is a conjugage distribution of the likelihood.
    - with multi-nomial, the conjugage is the Dirichlet distribution

$$\Pr(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \frac{1}{Z}\prod_w \theta_w^{\alpha_w}$$

Apply "MLE" on $P(\theta|D)$

"Laplacian"

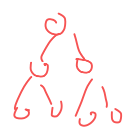
$$\Rightarrow \theta_w = [C(w)+\alpha_w] / \Sigma\left(C(w) + \alpha_w\right)$$

$$P(D|\theta) \times P(\theta|\alpha) = \frac{1}{Z}\prod_{w=1}^{|W|} \theta_w^{C(w)} \times \frac{1}{Z'}\prod_{w=1}^{|V|} \theta_w^{\alpha_w}$$

$$\propto \prod_{w=1}^{|V|} \theta_w^{\alpha_w + C(w)}$$

MLE in NLP

- Looking ahead, we will use MLE to learn a few models
  - Word embedding (glove) → word vectors
  - Hidden Markov Model (HMM) → sequences
  - Conditional random fields (CRF) → word vectors + seq.
  - Reccurent NN → NN version of CRF
  - Probabilistic CFG parser →



  - Recursive NN

→ NN on trees

Summary

- Maximum likelihood estimation
  - Probabilistic models: likelihood
- Tools:
  - gradient descent.
  - matrix calculus to find gradients.
  - constrained optimization.
- Maximum posterior estimation
  - Prior, conjugacy, and posterior using the Bayes theorem.
  - Laplacian smoothing as MAP