Maximum Entropy

Natural Language Processing: Jordan Boyd-Graber
University of Colorado Boulder
SEPTEMBER 22, 2014

Adapted from material by Robert Malouf, Philipp Koehn, and Matthew Leingang

Roadmap

- Why we need more powerful probabilistic modeling formalism
- Introducing key concepts from information theory
- Maximum Entropy Models
  - Why the have the form they do
  - How to estimate them from data

Outline

Modeling Distributions

- Modeling Distributions
- Estimating from data
- Thus far, only counting
  - MLE
  - Priors
  - Backoff

Modeling Distributions

- Modeling Distributions
- Estimating from data
- Thus far, only counting
    - MLE
    - Priors
    - Backoff
- What about features?

Supervised Learning

- Problem Setup
  - Given: some annotated data
  - Goal: Build a model
  - Task: Apply it to unseen data
- Issues
  - More data help
  - How to represent the data

Supervised Learning

- Problem Setup
  - Given: some annotated data
  - Goal: Build a model
  - Task: Apply it to unseen data
- Issues
  - More data help
  - How to represent the data
- Better document labeling

Supervised Learning

- Problem Setup
  - Given: some annotated data (document categories)
  - Goal: Build a model (using some feature representation)
  - Task: Apply it to unseen data (document labeling)
- Issues
  - More data help
  - How to represent the data
- Better document labeling

Contrast: Naive Bayes

- NB useful and simple; two parameters
  - Prior distribution
  - Conditional emission
- Training is easy from tagged data (counting)
- Find best label using arithmetic

Contrast: Naive Bayes

- NB useful and simple; two parameters
    - Prior distribution
    - Conditional emission
- Training is easy from tagged data (counting)
- Find best label using arithmetic
- But it ignores important clues that could help

Motivating Example: Document Classification

- We can do better than counting words

Motivating Example: Document Classification

- We can do better than counting words
- Bigrams
- Metadata
- Morphology (Apple documents: how many words start with `i`)
- Style (length of sentences)
- Number of repeated sentences

Encoding Features

- Much more powerful and expressive than counting **single** observations
  - $\vec{f}(x)$ a vector with the feature count for observation x
  - $f_i(x)$: count of feature i in observation x

Encoding Features

- Much more powerful and expressive than counting **single** observations
  - $\vec{f}(x)$ a vector with the feature count for observation x
  - $f_i(x)$: count of feature i in observation x
- Typical example

$$f(w_1, w_2, w_3, w_4, \dots w_n) =$$

Encoding Features

- Much more powerful and expressive than counting **single** observations
  - $\vec{f}(x)$ a vector with the feature count for observation x
  - $f_i(x)$: count of feature i in observation x
- Typical example

$$f(w_1, w_2, w_3, w_4, \ldots w_n) =$$

$f_1(\vec{w})$ Number of times you see the word "dog"
$f_2(\vec{w})$ Number of times you see the bigram "dog house"

Where Maximum Entropy Models Fit

- Suppose we have some data-driven information about these features
- What distribution should we use to model these features?
- "Maximum Entropy" models provide a solution

Where Maximum Entropy Models Fit

- Suppose we have some data-driven information about these features
- What distribution should we use to model these features?
- “Maximum Entropy” models provide a solution ... but first quick refresher

Outline

Motivation: Document Classification

Expectation and Entropy

Constraints

Maximum Entropy Form

Entropy & Expectation

An *expectation* of a random variable is a weighted average:

$$\mathbb{E}\left[f(X)\right] = \sum_{x=1}^{\infty} f(x)\, p(x) \qquad \text{(discrete)}$$

Entropy is a measure of uncertainty that is associated with the distribution of a random variable:

$$\begin{aligned} H(X) &= -\mathrm{E}\left[\lg(p(X))\right] \\ &= -\sum_{x} p(x)\, \lg(p(x)) \qquad \text{(discrete)} \end{aligned}$$

Principles for Modeling Distributions

Maximum Entropy Principle (Jaynes)

All else being equal, we should prefer distributions that maximize the Entropy

Principles for Modeling Distributions

Maximum Entropy Principle (Jaynes)

All else being equal, we should prefer distributions that maximize the Entropy

- What additional constraints do we want to place on the distribution?
- How, mathematically, do we optimize the entropy?

Outline

The obvious one . . .

- We're attempting to model a probability distribution p
- By definition, our probability distribution must sum to one

$$\sum_x p(x) = 1 \tag{1}$$

Feature constraints

- We observe features across many outcomes
- We're modeling a distribution p over observations x. What is the correct model of features under this distribution?
- The whole point of this is that we **don't** want to count outcomes (we've discussed those methods)

Feature constraints

- We observe features across many outcomes
- We're modeling a distribution p over observations x. What is the correct model of features under this distribution?
- The whole point of this is that we **don't** want to count outcomes (we've discussed those methods)
- Ideally, the expected count of the features should be consistent with observations

Estimated Counts

$$\mathbb{E}_p\left[f_i(x)\right] = \sum_x p(x) f_i(x) \tag{2}$$

Empirical Counts

$$\hat{\mathbb{E}}_{\hat{p}}\left[f_i(x)\right] = \hat{p}(x) f_i(x) \tag{3}$$

- Empirical distribution is just what we've observed in data

Optimizing Constrained Functions

Theorem: Lagrange Multiplier Method

Given functions $f(x_1, \ldots x_n)$ and $g(x_1, \ldots x_n)$, the critical points of f restricted to the set $g = 0$ are solutions to equations:

$$\begin{aligned} \frac{\partial f}{\partial x_i}(x_1, \ldots x_n) =& \lambda \frac{\partial g}{\partial x_i}(x_1, \ldots x_n) \quad \forall i \\ g(x_1, \ldots x_n) =& 0 \end{aligned}$$

This is $n + 1$ equations in the $n + 1$ variables $x_1, \ldots x_n, \lambda$.

Lagrange Example

Maximize $f(x, y) = \sqrt{xy}$ subject to the constraint $20x + 10y = 200$.

- Compute derivatives

Lagrange Example

Maximize $f(x, y) = \sqrt{xy}$ subject to the constraint $20x + 10y = 200$.

- Compute derivatives

$$\frac{\partial f}{\partial x} = \frac{1}{2}\sqrt{\frac{y}{x}} \quad \frac{\partial g}{\partial x} = 20$$
$$\frac{\partial f}{\partial y} = \frac{1}{2}\sqrt{\frac{x}{y}} \quad \frac{\partial g}{\partial y} = 10$$

Lagrange Example

Maximize $f(x, y) = \sqrt{xy}$ subject to the constraint $20x + 10y = 200$.

- Compute derivatives

$$\frac{\partial f}{\partial x} = \frac{1}{2}\sqrt{\frac{y}{x}} \qquad \frac{\partial g}{\partial x} = 20$$
$$\frac{\partial f}{\partial y} = \frac{1}{2}\sqrt{\frac{x}{y}} \qquad \frac{\partial g}{\partial y} = 10$$

- Create new systems of equations

Lagrange Example

Maximize $f(x, y) = \sqrt{xy}$ subject to the constraint $20x + 10y = 200$.

- Compute derivatives

$$\frac{\partial f}{\partial x} = \frac{1}{2}\sqrt{\frac{y}{x}} \quad \frac{\partial g}{\partial x} = 20$$
$$\frac{\partial f}{\partial y} = \frac{1}{2}\sqrt{\frac{x}{y}} \quad \frac{\partial g}{\partial y} = 10$$

- Create new systems of equations

$$\frac{1}{2}\sqrt{\frac{y}{x}} = 20\lambda$$
$$\frac{1}{2}\sqrt{\frac{x}{y}} = 10\lambda$$
$$20x + 10y = 200$$

Lagrange Example

- Dividing the first equation by the second gives us

$$\frac{y}{x} = 2 \tag{4}$$

- which means $y = 2x$, plugging this into the constraint equation gives:

$$\begin{aligned} 20x + 20(2x) &= 200 \\ x = 5 &\Rightarrow y = 10 \end{aligned}$$

Outline

Objective Function

- We want a distribution p that maximizes

$$H(p) \equiv -\sum_x p(x) \log p(x) \tag{5}$$

- Under the constraints that

$$\sum_x p(x) = 1 \tag{6}$$

- and, for every feature f_i

$$\mathbb{E}_p [f_i] = \hat{\mathbb{E}}_{\hat{p}} [f_i] . \tag{7}$$

Augmented Objective Function

$$
\begin{aligned}
\Lambda(p, \lambda, \gamma) = & \\
& -\sum_x p(x) \log p(x) \\
& -\sum_i \lambda_i \left(\sum_x p(x) f_i(x) - \hat{\mathbb{E}}\left[f_i\right] \right) \\
& -\gamma \left(\sum_x p(x) - 1 \right)
\end{aligned}
$$

Plan for solution:

- Take derivative
- Set it equal to zero
- Solve for the $p(x)$ that optimizes equation
- This will give the functional form of our solution

Solution

$$0 = \frac{\partial \Lambda(p, \lambda, \gamma)}{\partial p(x)} \tag{8}$$

$$0 = -(1 + \log(p(x)) + \sum_i \lambda_i f_i(x) + \gamma \tag{9}$$

$$\log(p(x)) = \sum_i \lambda_i f_i(x) + \gamma - 1 \tag{10}$$

Solution

$$0 = \frac{\partial \Lambda(p, \lambda, \gamma)}{\partial p(x)} \tag{8}$$

$$0 = -(1 + \log(p(x)) + \sum_i \lambda_i f_i(x) + \gamma \tag{9}$$

$$\log(p(x)) = \sum_i \lambda_i f_i(x) + \gamma - 1 \tag{10}$$

Now solve for $p(x)$:

$$p(x) = \exp\{\gamma - 1\} \exp\left\{\sum_i \lambda_i f_i(x)\right\} \tag{11}$$

Solution

Substitute into normalization (Equation 6):

$$\sum_x p(x) = 1 \tag{12}$$

Solution

Substitute into normalization (Equation 6):

$$\sum_x p(x) = 1 \tag{12}$$

$$\sum_x \exp\{\gamma - 1\} \exp\left\{\sum_i \lambda_i f_i(x)\right\} = 1 \tag{13}$$

Solution

Substitute into normalization (Equation 6):

$$\sum_x p(x) = 1 \tag{12}$$

$$\sum_x \exp\{\gamma - 1\} \exp\left\{\sum_i \lambda_i f_i(x)\right\} = 1 \tag{13}$$

$$\exp\{\gamma - 1\} \sum_x \exp\left\{\sum_i \lambda_i f_i(x)\right\} = 1 \tag{14}$$

Solution

Substitute into normalization (Equation 6):

$$\sum_x p(x) = 1 \tag{12}$$

$$\sum_x \exp\{\gamma - 1\} \exp\left\{\sum_i \lambda_i f_i(x)\right\} = 1 \tag{13}$$

$$\exp\{\gamma - 1\} \sum_x \exp\left\{\sum_i \lambda_i f_i(x)\right\} = 1 \tag{14}$$

$$\exp\{\gamma - 1\} = \frac{1}{\sum_x \exp\{\sum_i \lambda_i f_i(x)\}} \tag{15}$$

Final steps!

Substitute Equation 12 into Equation 11:

$$p(x) = \exp\{\gamma - 1\} \exp\left\{\sum_i \lambda_i f_i(x)\right\} \tag{16}$$

Final steps!

Substitute Equation 12 into Equation 11:

$$p(x) = \exp\{\gamma - 1\} \exp\left\{\sum_i \lambda_i f_i(x)\right\} \tag{16}$$

Final steps!

Substitute Equation 12 into Equation 11:

$$p(x) = \exp\{\gamma - 1\} \exp\left\{\sum_i \lambda_i f_i(x)\right\} \tag{16}$$

$$p(x) = \frac{1}{\sum_x \exp\{\sum_i \lambda_i f_i(x)\}} \exp\left\{\sum_i \lambda_i f_i(x)\right\} \tag{17}$$

Final steps!

Substitute Equation 12 into Equation 11:

$$p(x) = \exp\{\gamma - 1\} \exp\left\{\sum_i \lambda_i f_i(x)\right\} \tag{16}$$

$$p(x) = \frac{1}{\sum_x \exp\{\sum_i \lambda_i f_i(x)\}} \exp\left\{\sum_i \lambda_i f_i(x)\right\} \tag{17}$$

More concretely:

$$p(x) = \frac{\exp\left\{\lambda^\top \vec{f}(x)\right\}}{\sum_{x'} \exp\left\{\lambda^\top \vec{f}(x)\right\}} \tag{18}$$

Form of Solution

- Other ways to arrive at same answer (monkeys throwing balls)
- Should remind you of logistic regression

$$p(x) = \frac{\exp\left\{\lambda^\top \vec{f}(x)\right\}}{\sum_{x'} \exp\left\{\lambda^\top \vec{f}(x)\right\}} \tag{19}$$

- Thus, distribution is parameterized by $\vec{\lambda}$ (one for each feature, used β before)

Finding Parameters

- Form is simple
- However, finding parameteris is difficult
- Solutions take iterative form
  1. Start with $\vec{\lambda}^{(0)} = \vec{0}$
  2. For $k = 1 \dots$
     2.1 Determine update $\vec{\delta}^{(k)}$
     2.2 $\vec{\lambda}^{(k)} \rightarrow \vec{\lambda}^{(k-1)} + \vec{\delta}^{(k)}$

Method for finding updates

- Our objective is a function of $\vec{\lambda}$

$$L(\lambda) = \sum_x \frac{\exp\left\{\lambda^\top f(x)\right\}}{\sum_{x'} \exp\left\{\lambda^\top f(x')\right\}} \tag{20}$$

(in practice, we typically use the log probability)

- Strategy: Move $\vec{\lambda}$ by walking up the gradient $G(\lambda^{(k)})$
- Gradient

$$G_i(\lambda) = \frac{\partial L(\lambda)}{\partial \lambda_i} = -\left[\left(\sum_x p_\lambda(x) f_i(x)\right) - \hat{\mathbb{E}}\left[f_i\right]\right] \tag{21}$$

Method for finding updates

- Set the update of the form

$$\delta^{(k)} = \alpha^{(k)} G(\lambda^{(k)}) \tag{22}$$

- Use the new parameter

$$\vec{\lambda}^{(k)} \rightarrow \vec{\lambda}^{(k-1)} + \vec{\delta}^{(k)} \tag{23}$$

- What value of α?

Method for finding updates

- Set the update of the form

$$\delta^{(k)} = \alpha^{(k)} G(\lambda^{(k)}) \tag{22}$$

- Use the new parameter

$$\vec{\lambda}^{(k)} \rightarrow \vec{\lambda}^{(k-1)} + \vec{\delta}^{(k)} \tag{23}$$

- What value of α?
  - Try lots of different values, pick the one that optimizes $L(\lambda)$ (grid search)

Other parameter estimation techniques

- Iterative scaling
- Conjugate gradient methods
- Real difference is speed and scalability

Regularization / Priors

- We often want to prefer small parameters over large ones, all else being equal

$$L(\lambda) = \sum_x \frac{\exp\left\{\lambda^\top f(x)\right\}}{\sum_{x'} \exp\left\{\lambda^\top f(x')\right\}} - \sum_i \frac{\lambda^2}{\sigma^2} \tag{24}$$

- This is equivalent to having a Gaussian prior on the weights λ
- Also possible to use **informed** priors when you have an idea of what the weights should be (e.g. for domain adaptation)

All sorts of distributions

- We talked about a simple distribution $p(x)$
- But could just as easily be joint distribution $p(y, x)$

$$p(y, x) = \frac{\exp\left\{\lambda^\top f(y, x)\right\}}{\sum_{y', x'} \exp\left\{\lambda^\top f(y', x')\right\}} \tag{25}$$

- Or a conditional distribution $p(y|x)$

$$p(y|x) = \frac{\exp\left\{\lambda^\top f(y, x)\right\}}{\sum_{y'} \exp\left\{\lambda^\top f(y', x)\right\}} \tag{26}$$

Uses of MaxEnt Distributions

- POS Tagging (state of the art)
- Supervised classification: spam vs. not spam
- Parsing (head or not)
- Many other NLP applications

In class . . .

In class . . .

In class . . .

In class . . .

In class . . .