Statistical Methods for NLP

Maximum Entropy Models

Sameer Maskey

Week 6, Feb 23, 2010

Topics for Today

- Logistic Regression/Maximum Entropy Models

Project

- Feb 23, 2010 (11:59pm) : Project Proposal
- March 23, 2010 (4pm) : Project Status Update
- April 20, 2010 (4pm) : Final Projects Due
- April 27, 2010 (4pm) : Class Presentations for the project

Maximum Entropy Model

- Maximum Entropy Model has shown to perform well in many NLP tasks
  - POS tagging [Ratnaparkhi, A., 1996]
  - Text Categorization [Nigam, K., et. al, 1999]
  - Named Entity Detection [Borthwick, A, 1999]
  - Parser [Charniak, E., 2000]
- Discriminative classifier
  - Conditional model P(c|d)
  - Maximize conditional likelihood
  - Can handle variety of features

Naïve Bayes vs. Maximum Entropy Models

Naïve Bayes Model

- Trained by maximizing likelihood of data and class
- Features are assumed independent
- Feature weights set independently

Maximum Entropy Model

- Trained by maximizing conditional likelihood of classes
- Dependency on features taken account by feature weights
- Feature weights are set mutually

Entropy

$$H(p) = -\sum_{x} p(x) log_2\ p(x)$$

- Measure of uncertainty
- Higher uncertainty equals higher entropy
- Degree of surprise of an event

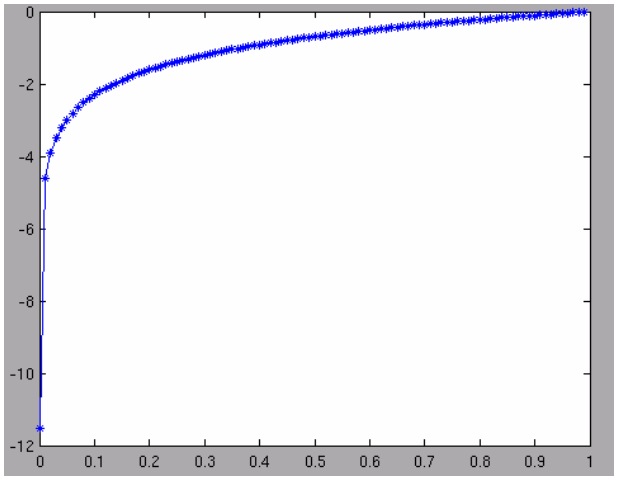
- Why this formula in particular? Why log?

Exploring the Entropy Formulation

- How much information received when observing a random variable ‘x’ ?
- Highly improbable event = received more information
- Highly probable event = received less information

- Need h(x) that express information content of p(x); we want
  1. Monotonic function of p(x)
  2. If p(x,y) = p(x). p(y) when x and y are unrelated, i.e. statistically independent then we want h(x,y) = h(x) + h(y) such that information gain by observing two unrelated events is their sum

Exploring Entropy Formulation (cont.)



Remember logarithm function

- What kind of h(x) satisfies two conditions mentioned previously

$$h(x) = - \log_2 p(x)$$

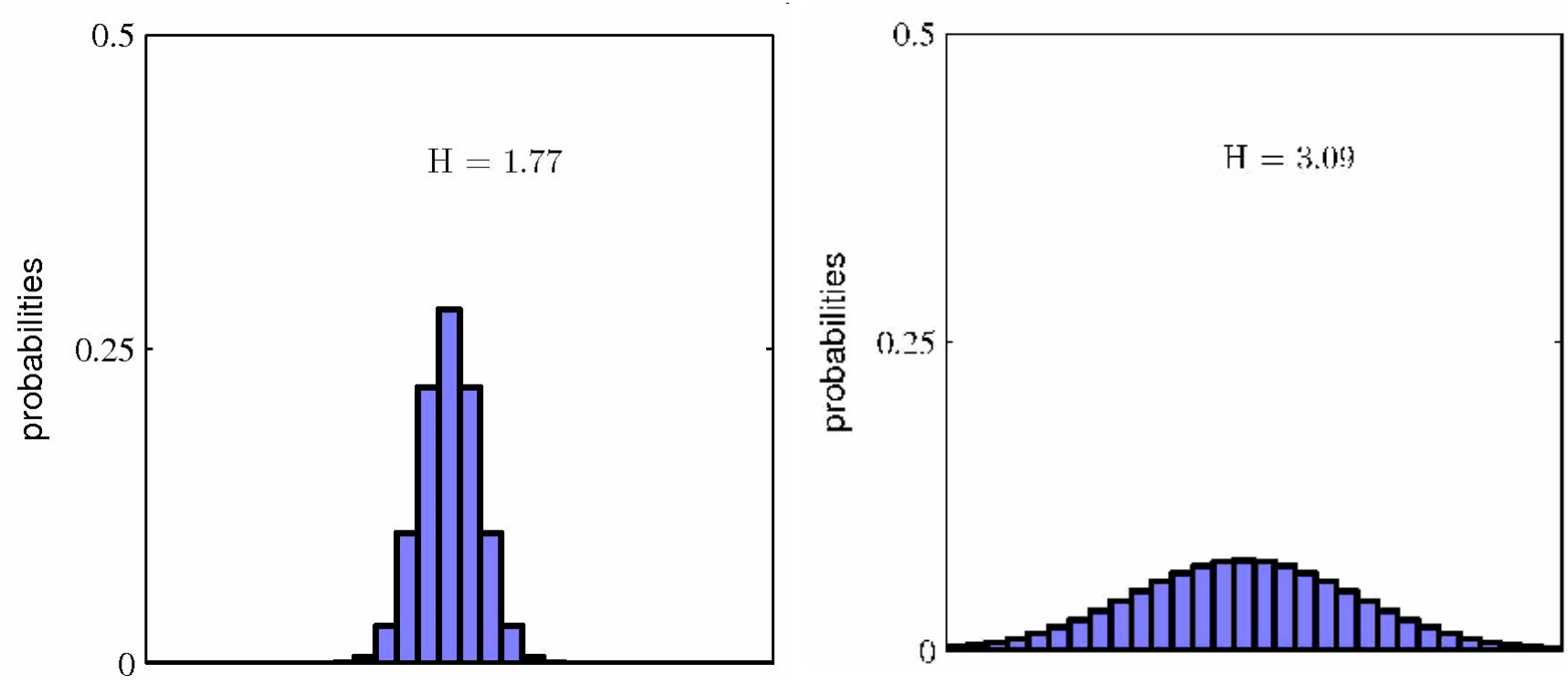
- Log of base 10 is ok as well

Entropy Formula

- h(x) = - $\log_2$ p(x) : information observed
- Expected amount of information observed can be found by taking expectation with respect to p(x)

$$H(p) = -\sum_x p(x) log_2 p(x)$$

Comparing Entropy Across Distributions



[1] Uniform distribution has higher entropy

Maximizing Entropy

- How can we find a distribution with maximum entropy?
- What about maximizing entropy of a distribution with a set of constraints?
- What does maximizing entropy has to do with classification task anyway?

- Let us first look at logistic regression to understand this

Remember Linear Regression

$$y_j = \theta_0 + \theta_1 x_j$$

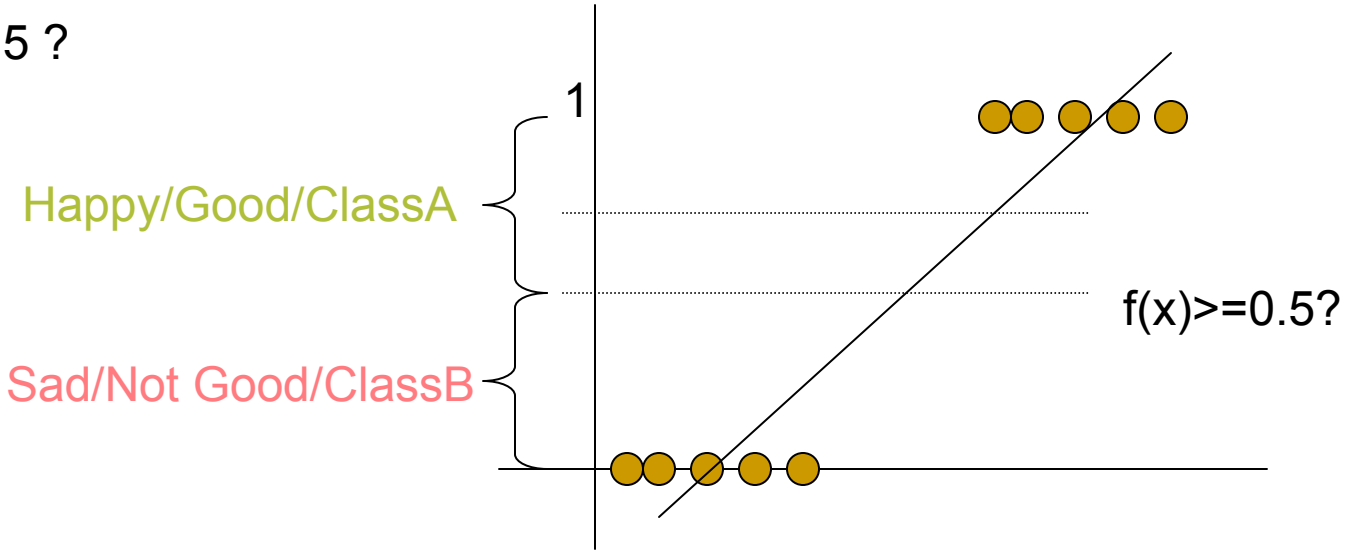
$$y_j = \sum_{i=0}^{N} \theta_i x_{ij} \qquad \text{where } \mathrm{x}_{0j} = 1$$

N is the number of dimensions where each input lives in

- We estimated theta by setting square loss function's derivative to zero

Regression to Classification

- We also looked at why linear regression may not work well if ‘y’ are binary
  - Output (-infinity to +infinity) is not limited to class labels (0 and 1)
  - Assumption of noise (errors) normally distributed
- Train Regression and threshold the output
  - If f(x) >= 0.7 CLASS1
  - If f(x) < 0.7 CLASS2
  - f(x) >= 0.5 ?



Ratio

- Instead of thresholding the output we can take the ratio of two probabilities
- Ratio is odds of predicting y=1 or y=0
  - E.g. for given ‘x’ if p(y=1) = 0.8 and p(y=0) = 0.2
  - Odds = 0.8/0.2 = 4
- Better?
  - We can make the linear model predict odds of y=1 instead of ‘y’ itself

$$\frac{p(y=true|\mathbf{x})}{p(y=false|\mathbf{x})} = \sum_{i=0}^{N} \theta_i \; x_i$$

Log Ratio

$$\frac{p(y=true|\mathbf{x})}{p(y=false|\mathbf{x})} = \sum_{i=0}^{N} \theta_i \, x_i$$

- LHS is between 0 and infinity, we want to be able to handle – infinity to +infinity which RHS can produce
- If we take log of LHS, it can also range between –infinity and +ve infinity

$$log(\frac{p(y=true|\mathbf{x})}{p(y=false|\mathbf{x})})$$

$$log(\frac{p(y=true|\mathbf{x})}{(1-p(y=true|\mathbf{x})})$$

$$logit(p(x)) = log\frac{p(x)}{1-p(x)}$$

Logistic Regression

$$log(\frac{p(y=true|\mathbf{x})}{(1-p(y=true|\mathbf{x})}) = \sum_{i=0}^{N} \theta_i \times x_i$$

- Logistic Regression: A Linear Model in which we predict logit of probability instead of probability

$$log(\frac{p(y=true|\mathbf{x})}{(1-p(y=true|\mathbf{x})}) = w \cdot f$$

Logistic Regression Derivation

$$log(\frac{p(y=true|\mathbf{x})}{(1-p(y=true|\mathbf{x})}) = w \cdot f$$

$$\frac{p(y=true|\mathbf{x})}{(1-p(y=true|\mathbf{x})} = exp(w \cdot f)$$

$$p(y = true|\mathbf{x}) = (1 - p(y = true|\mathbf{x})exp(w \cdot f)$$

$$p(y = true|\mathbf{x}) = exp(w \cdot f) - p(y = true|\mathbf{x})exp(w \cdot f)$$

$$p(y = true|\mathbf{x}) + p(y = true|\mathbf{x})exp(w \cdot f) = exp(w \cdot f)$$

$$p(y = true|\mathbf{x}) = \frac{exp(w \cdot f)}{1+exp(w \cdot f)}$$

Logistic Regression

$$p(y = true|\mathbf{x}) = \frac{exp(\sum_{i=0}^{N} \theta_i x_i)}{1+exp(\sum_{i=0}^{N} \theta_i x_i)}$$

$$p(y = false|\mathbf{x}) = \frac{1}{1+exp(\sum_{i=0}^{N} \theta_i x_i)}$$

For notation convenience for later part of the lecture replace theta with lambda and x with f where f is an indicator function

$$p(y = true|\mathbf{x}) = \frac{exp(\sum_{i=0}^{N} \lambda_i f_i)}{1+exp(\sum_{i=0}^{N} \lambda_i f_i)}$$

$$p(y = false|\mathbf{x}) = \frac{1}{1+exp(\sum_{i=0}^{N} \lambda_i f_i)}$$

Logistic Regression for Multiple Classes

- We can also have logistic regression for multiple classes
- Normalization has to take account of all classes

$$p(c|\mathbf{x}) = \frac{exp(\sum_{i=0}^{N} \lambda_{ci} f_i)}{\sum_{c' \in C} exp(\sum_{i=0}^{N} \lambda_{c'i} f_i)}$$

Exponential Models

- Turns out logistic regression is just a type of exponential model
  - Linear combination of weights and features to produce a probabilistic model

$$p(c|\mathbf{x}) = \frac{exp(\sum_{i=0}^{N} \lambda_{ci} f_i)}{\sum_{c' \in C} exp(\sum_{i=0}^{N} \lambda_{c'i} f_i)}$$

How Can We Estimate Weights

- How to estimate weights (Lambdas)
- For linear regression computed loss function and found derivative to zero
- We can estimate weights by maximizing (conditional) likelihood of data according to the model

So why did we talk all about logistic regression when we were trying to learn Maximum Entropy Models?

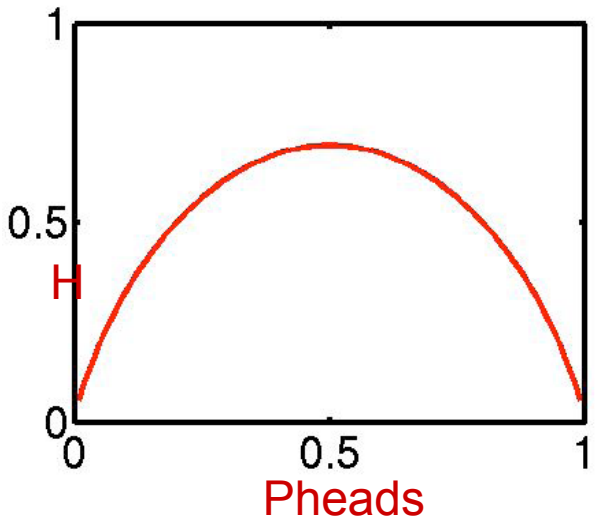
Let's find out

Maximum Entropy

- We saw what entropy is

$$H(p) = -\sum_x p(x) log_2 p(x)$$

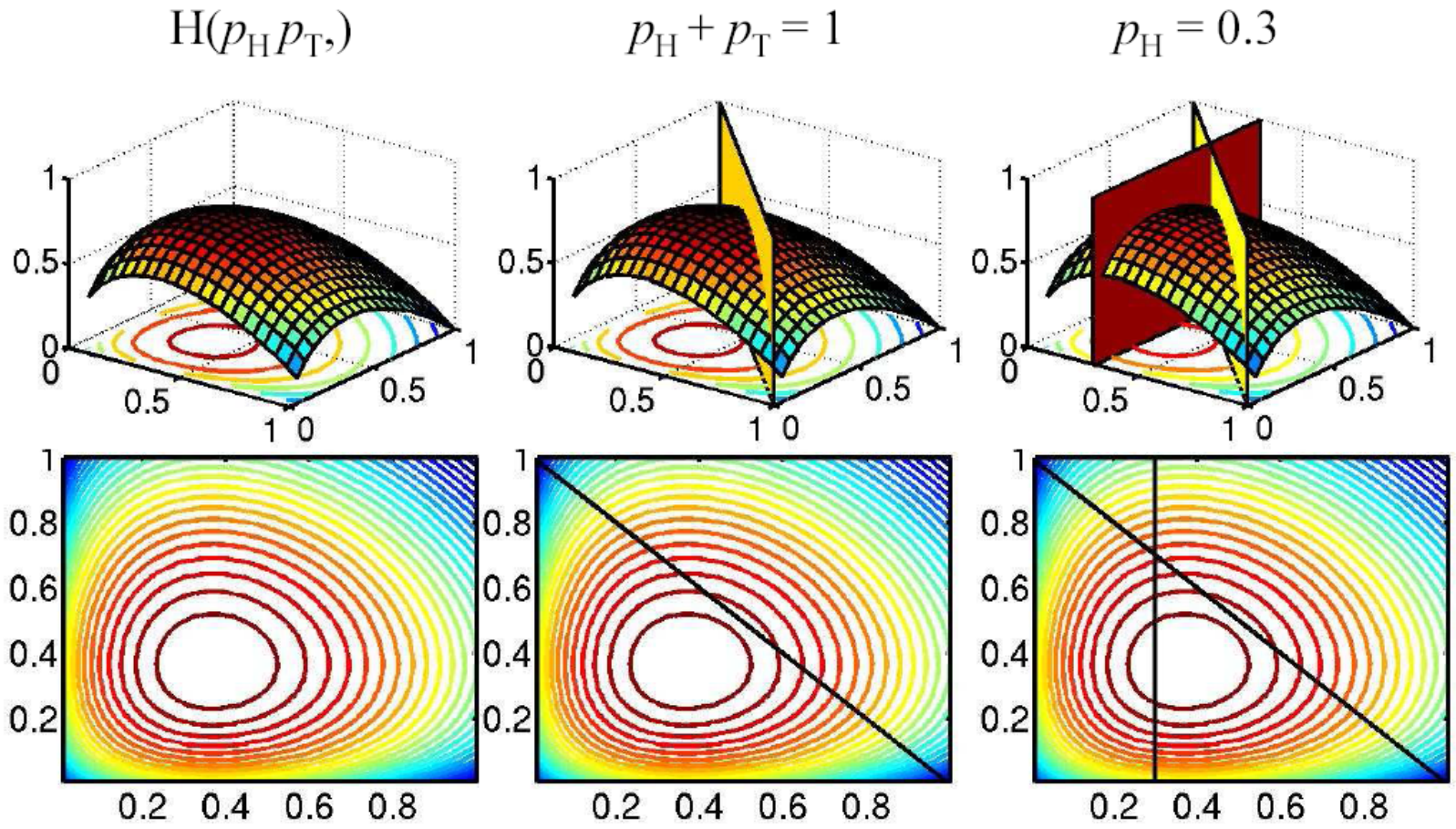
- We want to maximize entropy
  - Maximize subject to feature-based constraints
  - Feature based constraints help us bring the model distribution close to empirical distribution (data)
  - In other words it increases maximum likelihood of data given the model but makes the distribution less uniform



Fair coin has the highest entropy

Constraints on a Entropy Function

Figure below is from Klein, D. and Manning, C., Tutorial [1]



Features

- We have seen many different types of features
  - Count of words, length of docs, etc
- We can think of features as indicator functions that represent co-occurrence relation between input phenomenon and the class we are trying to predict

$$f_i(c_d) = \phi(d) \wedge c_d = c_i$$

Example: Features for POS Tagging

- $f_1(c,d)$ = { c=NN Λ curword(d)=book Λ prevword(d)=to}
- $f_2(c,d)$ = { c=VB Λ curword(d)=book Λ prevword(d)=to}
- $f_3(c,d)$ = { c=VB Λ curword(d)=book Λ Λ prevClass(d)=ADJ}

Maximum Entropy Example

Given Event space

NN	JJ	NNS	VB

Maximum Entropy Distribution

1/4	1/4	1/4	1/4

Add a constraint P(NN) + P(JJ) + P(NNS) = 1

1/3	1/3	1/3	0

Add another constraint P(NN) + P(NNS) = 8/10

4/10	2/10	4/10	0

Expectation of a Feature

- We can count the features from the labeled set of data

$$Empirical(f_i) = \sum_{(c,d) \in observed(C,D)} f_i(c,d)$$

- Expectation of a feature given the trained model

$$E(f_i) = \sum_{(c,d) \in (C,D)} p(c,d) f_i(c,d)$$

Maximization with Constraints

$$max_{p(x)} H(p) = -\sum_x p(x) log p(x)$$

$$s.t. \sum_x p(x) f_i(x) = \sum_x p(\tilde{x}) f_i(x), i = 1...N$$

$$\sum_x p(x) = 1$$

Solving MaxEnt

- MaxEnt is a convex optimization problem with concave objective function and linear constraints
- We have seen such optimization problems before
  - Solved with Lagrange multipliers

Lagrange Equation

$$L(p,\lambda) = -\sum_x p(x) log p(x) + \lambda_0 [\sum_x p(x) - 1] +$$
$$\sum_{i=1}^{N} \lambda_i [\sum_x p(x) f_i(x) - \sum_x p(\tilde{x}) f_i(x)]$$

Lagrangian gives us unconstrained optimization as constraints are built into the equation. We can now solve it by setting derivatives to zero

Maximum Entropy and Logistic Regression

- This unconstrained optimization problem is a dual problem equivalent to estimating maximum likelihood of logistic regression model we saw before

Maximizing entropy subject to our constraints
Is equivalent to
Maximum likelihood estimation over exponential family of $p_\lambda(x)$

Maximum Entropy and Logistic Regression

"Exponential Model for Multinomial Logistic Regression, when trained according to the maximum likelihood criterion, also finds the Maximum Entropy Distribution subject to the constraints from the feature function" [2]

Finding Maximum Likelihood of our Conditional Models (Multinomial Logistic Regression)

$$(C|D,\lambda) = \prod_{(c,d)\in(C,D)} p(c|d,\lambda)$$

$$P(C|D,\lambda) = \sum_{(c,d)\in(C,D)} logp(c|d,\lambda)$$

$$P(C|D,\lambda) = \sum_{(c,d)\in(C,D)} log\frac{exp\sum_i \lambda_i f_i(c,d)}{\sum_{c'} exp\sum_i \lambda_i f_i(c',d)}$$

Maximizing Conditional Log Likelihood

$$P(C|D,\lambda) = \sum_{(c,d)\in(C,D)} log exp \sum_i \lambda_i f_i(c,d)$$

$$- \sum_{(c,d)\in(C,D)} log \sum_{c'} exp \sum_i \lambda_i f_i(c',d)$$

Taking derivative and setting it to zero

$$\frac{\partial log(P|C,\lambda)}{\partial \lambda_i} = \sum_{(c,d)\in(C,D)} f_i(c,d) - \sum_{(c,d)\in(C,D)} \sum_{c'} P(c'|d,\lambda) f_i(c',d)$$

Empirical count (f_i, c)

Predicted count (f_i, λ)

Optimal parameters are obtained when empirical expectation equal predicted expectation

Finding Model Parameters

- We saw that optimum parameters are obtained when empirical expectation of a feature equals predicted expectation
- We are finding a model having maximum entropy and satisfying constraints for all features fj

$$E_p(f_j) = E_{\tilde{(p)}}(f_j)$$

- Hence finding the parameters of maximum entropy model entails to maximizing conditional log-likelihood and solving it
  - Conjugate Gradient Descent
  - Quasi Newton's Method
  - A simple iterative scaling
    - Features are non-negative (indicator functions are non-negative)
    - Add a slack feature $f_{m+1}(d,c) = M - \sum_{j=1}^{m} f_j(d,c)$
    - where $M = max_{i,c} \sum_{j=1}^{m} f_j(d_i,c)$

Generalized Iterative Scaling

- Empirical Expectation $E_{\tilde{p}}(f_j) = \frac{1}{N} \sum_{i=1}^{N} f_j(d_i, c_i)$
- Initialize m+1 lambdas to 0
- Loop Until Converged

$$E_{p^t}(f_j) = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} P(c_k|d_i) f_j(d_i, c_k)$$

$$\lambda_j^{t+1} = \lambda_j^t + \frac{1}{M} log(\frac{E_{\tilde{p}}(f_j)}{E_{p^t}(f_j)})$$

- End loop

Summary

- Logistic Regression
  - Maximize conditional log-likelihood to estimate parameters
- Maximum Entropy Model
  - Maximize entropy with feature constraints
  - Constrained maximization
- Solving for H(p) with maximum entropy is equivalent to maximizing conditional log-likelihood for our exponential model

References

- [1] Klein, D., Manning C., “Maximum Models, Conditional Estimation and Optimization” ACL 2003
- [2] Jurafsky, D. and Martin, J., J&M Book, 2nd Edition
- [3] http://webdocs.cs.ualberta.ca/~swang/me.html