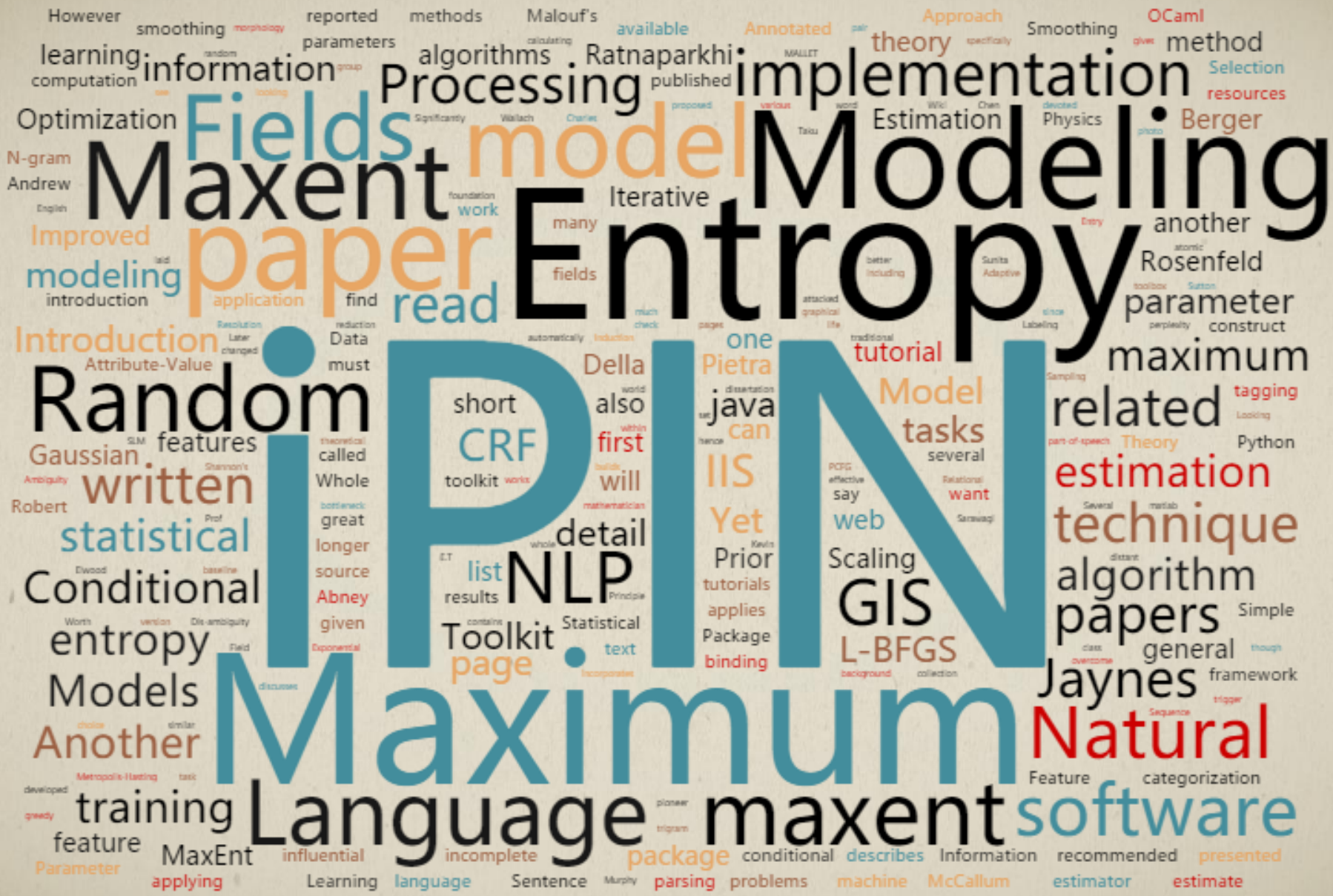

However
smoothing
reported
methods
Malouf's
available
Annotated
Approach
OCaml
Smoothing
learning
parameters
algorithms
Ratnaparkhi
theory
method
computation
information
Processing
published
implementation
Selection
resources
Optimization
Fields
model
Estimation
Physics
Berger
N-gram
Maxent
Modeling
Andrew
Iterative
work
Improved
paper
Entropy
many
another
modeling
fields
Rosenfeld
introduction
application
find
read
parameter
Introduction
Data
one
tutorial
maximum
Attribute-Value
must
Random
IPIN
Della
Pietra
Model
tagging
short
java
related
CRF
also
tasks
Gaussian
features
first
can
Theory
Python
called
several
written
IIS
estimation
Whole
Robert
will
want
statistical
great
Yet
technique
web
detail
longer
Scaling
Conditional
list
NLP
Prior
algorithm
source
tutorials
GIS
Abney
papers
Simple
entropy
given
Toolkit
Statistical
applies
L-BFGS
general
Models
page
text
Package
Jaynes
framework
Another
Maximum
Natural
binding
Feature
categorization
training
Language
maxent
software
feature
MaxEnt
influential
incomplete
package
conditional
describes
Information
recommended
presented
Parameter
applying
Learning
language
Sentence
parsing
problems
machine
McCallum
estimator
estimate

Bowl Maximum Entropy #4 By Ejay Weiss

Maxent Models: Foundations

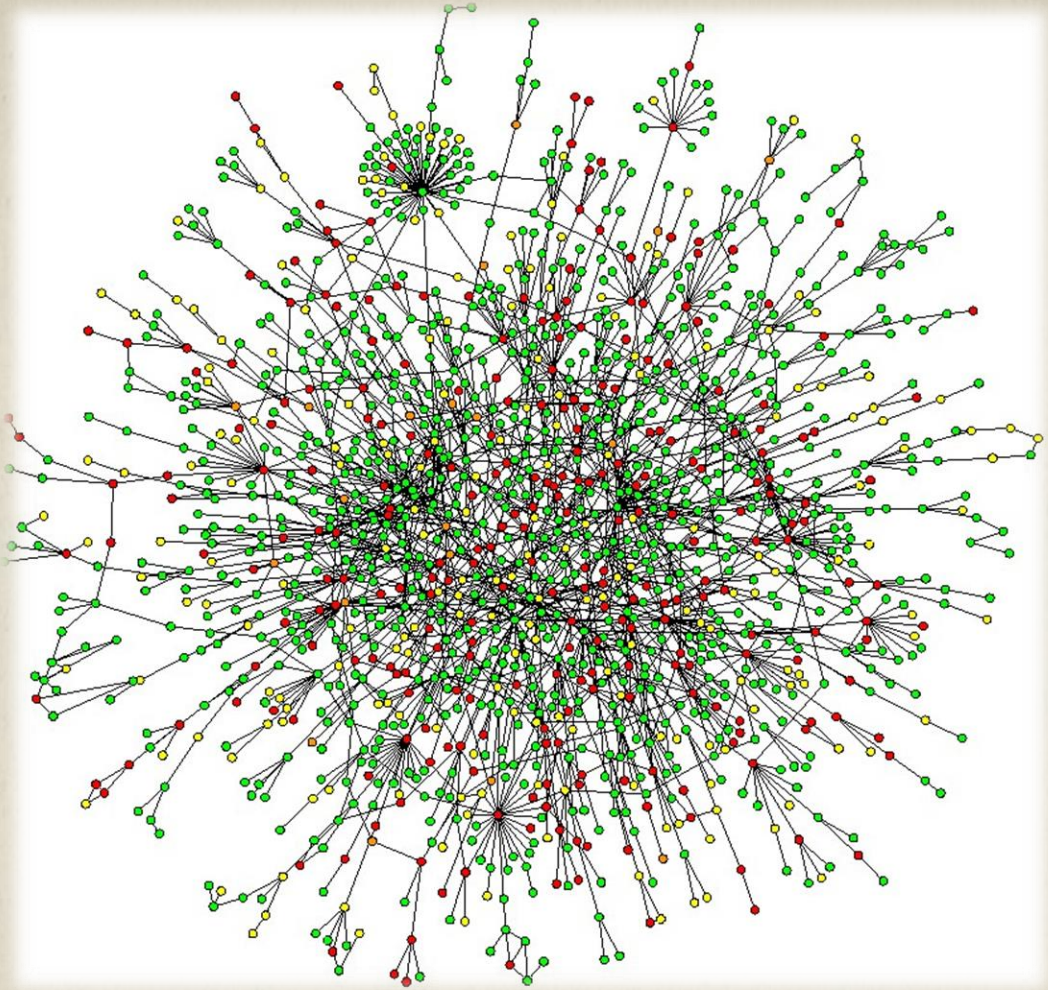
Maximum Entropy

A Basic Comprehension with Derivations

By Yanju Chen

Outlines

- Generative vs. Discriminative
- Feature-Based Models
- Softmax Function & Exponential Family
- Maximum Entropy Derivation
- Conclusions



A Large Graphical Model

Generative vs. Discriminative

iPIN

Generative Models

- We have some data $\{(d,c)\}$ of paired
  - observations d
  - hidden classes c
- A generative model place probabilities over both *observed data* and the *hidden stuff:*

$$P(c,d)$$

- Classic Generative Models:
  - n-gram Models, Naïve Bayes Classifiers, Hidden Markov Models, etc.

Discriminative Models

- We have some data $\{(d,c)\}$ of paired
  - observations d
  - hidden classes c
- A discriminative model take the *observed data* as given, and put a probability over *hidden structure* given the *observed data*.

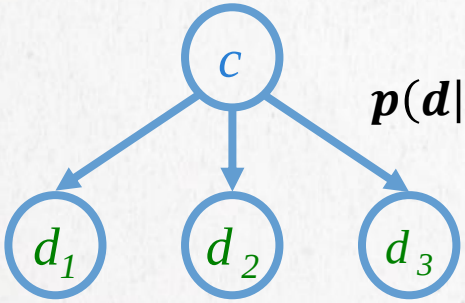
$$P(c|d)$$

- Classic Discriminative Models:
  - Logistic Regression, Conditional Random Fields, SVMs(not directly probabilistic), etc.

A Comparison

Generative Models		
$P(c,d)$	c_0	c_1
d_0	0.2	0.2
d_1	0.18	0.42

Discriminative Models		
$P(c\|d)$	c_0	c_1
d_0	0.5	0.5
d_1	0.3	0.7



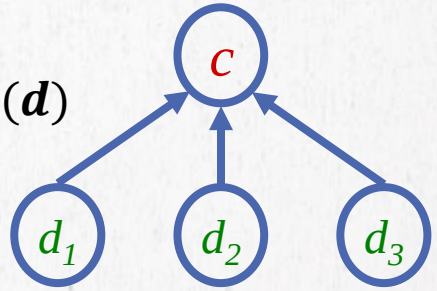
Naïve Bayes

Generative

$p(c,d) = p(d|c)p(c)$

Bayes' Theorem

$$\boldsymbol{p(d|c)p(c) = p(c|d)p(d)}$$



Logistic Regression

Discriminative

$p(c,d) = p(c|d)p(d)$?

Another Comparison

Training Set	
Objective	Accuracy
Generative	86.8
Discriminative	98.5

Test Set	
Objective	Accuracy
Generative	73.6
Discriminative	76.1

- Even with exactly the same features, changing from joint to conditional estimation increases performance
- That is, we use the same smoothing, and the same word-class features, we just change the numbers (parameters)

Klein and Manning 2002, using Senseval-1 Data

So, why Generative?

- Better Inference Algorithm
  - EM vs. GD
- Modular Learning, New Classes & Missing Data
- Better Accuracy on Future Data
- Make Explicit Claims about the Process that Underlies A Data
- Generate Synthetic Data Sets (Sampling)

Pinyin Recognition

- Given a Pinyin sequence D={kai, fang, ri}, find out the most possible Chinese characters.

$$c_1, c_2, c_3 = \underset{c_1,c_2,c_3}{\operatorname{argmax}}\, p(c_1, c_2, c_3 | d_1, d_2, d_3)$$

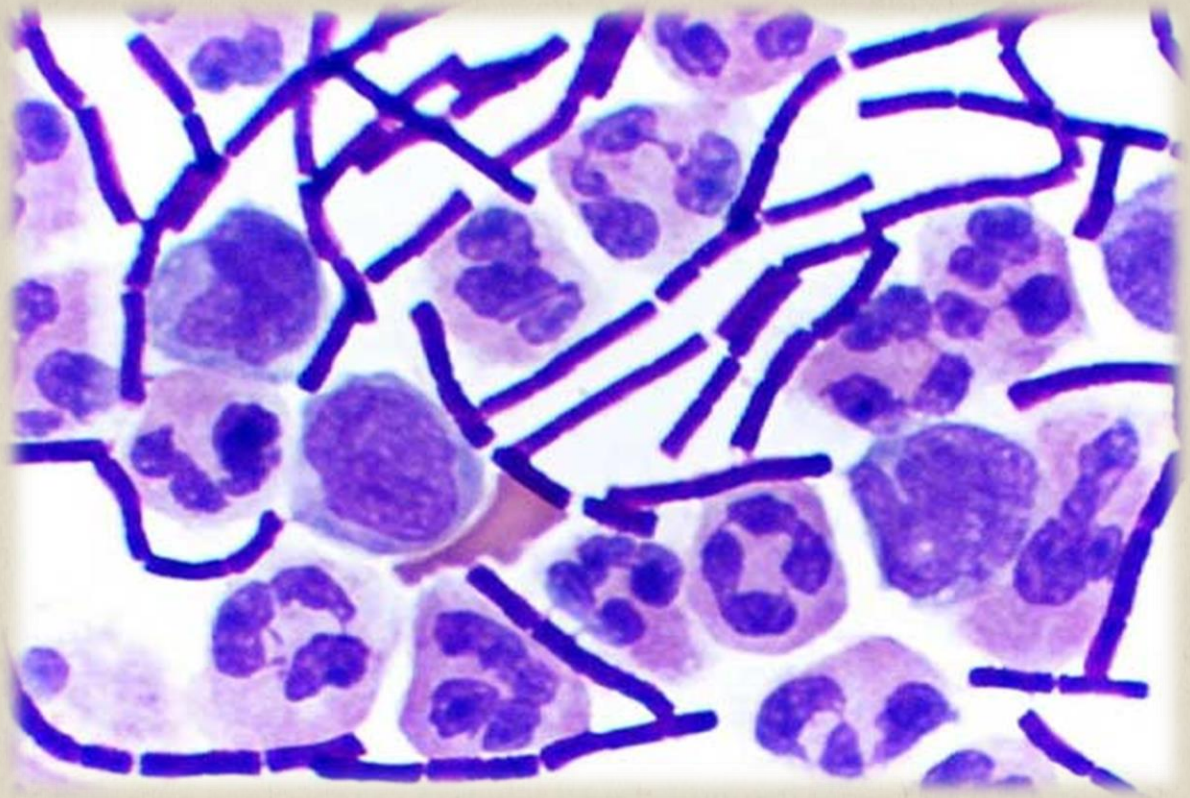
$$p(c_1, c_2, c_3 | d_1, d_2, d_3) = \frac{p(d_1, d_2, d_3 | c_1, c_2, c_3) p(c_1, c_2, c_3)}{p(d_1, d_2, d_3)}$$

$$p(d_1, d_2, d_3 | c_1, c_2, c_3) p(c_1, c_2, c_3)$$

$$p(d_1, d_2, d_3 | c_1, c_2, c_3) = p(d_1 | c_1) p(d_2 | c_2) p(d_3 | c_3)$$

$$p(c_1, c_2, c_3) = p(c_3 | c_2) p(c_2 | c_1) p(c_1)$$

$$c_1, c_2, c_3 = \underset{c_1,c_2,c_3}{\operatorname{argmax}}\, p(d_1 | c_1) p(d_2 | c_2) p(d_3 | c_3) \cdot p(c_3 | c_2) p(c_2 | c_1) p(c_1)$$



A Grain stain of Bacillus anthracis

Feature-Based Models

iPIN

Features

- Features f are
  - elementary pieces of evidence that link aspects of what we observe d with a category c that we want to predict
- A feature is a function with a bounded real value.
  - $f: C \times D \rightarrow \mathbb{R}$
- In NLP uses, usually a feature specifies an indicator function – a **yes/no** Boolean matching function.
  - Each feature picks out a data subset and suggests a label for it.

Minesweeper: An Example

- 8x8 Map of Minesweeper
- Size of Data Set D: 64
- Labels C={0:safe, 1:unsafe}
- $f_1 = [c = 0 \ \wedge d \ in \ corner]$
- $f_2 = [c = 1 \ \wedge suma. \geq 8]$

Minesweeper: An Example

- $f_1 = [c = 0 \ \wedge d \ in \ corner]$

0	0	1	1	1	1	1	1
0							1
1							1
1							1
1							1
0							1
1							1
1	1	1	1	1	1	1	1

Minesweeper: An Example

- $f_2 = [c = 1 \wedge suma. \geq 8]$

0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	0
0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
1	0	1	0	1	0	0	0
0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0

Feature Expectations

- Empirical Expectation of A Feature
  - $\tilde{E}(f_i) = \sum_{(c,d) \in observed(C,D)} f_i(c,d)$
- Model Expectation of A Feature
  - $E(f_i) = \sum_{(c,d) \in (C,D)} p(c,d) f_i(c,d)$

Example: Text Categorization

(Zhang and Oles 2001)

- Features are presence of each word in a document and the document class (they do feature selection to use reliable indicator words)
- Tests on classic Reuters data set (and others)
  - Naïve Bayes: 77.0% F1
  - Linear regression: 86.0%
  - Logistic regression: 86.4%
  - Support vector machine: 86.5%
- Paper emphasizes the importance of regularization (smoothing) for successful use of discriminative methods (not used in much early NLP/IR work)

Other Examples

- Sentence boundary detection (Mikheev 2000)
  - Is a period end of sentence or abbreviation?
- Sentiment analysis (Pang and Lee 2002)
  - Word unigrams, bigrams, POS counts, …
- PP attachment (Ratnaparkhi 1998)
  - Attach to verb or noun? Features of head noun, preposition, etc.
- Parsing decisions in general (Ratnaparkhi 1997; Johnson et al. 1999, etc.)

Feature-Based Linear Classifiers

- Linear classifiers at classification time:
  - Linear function from feature sets $\{f_i\}$ to classes $\{c\}$.
  - Assign a weight λ_i to each feature f_i.
  - We consider each class for an observed datum d
  - For a pair (c,d), features vote with their weights:

$$\text{vote}(c) = \Sigma \lambda_i f_i(c,d)$$

  - Choose the class c which maximizes $\Sigma \lambda_i f_i(c,d)$

Feature-Based Linear Classifiers

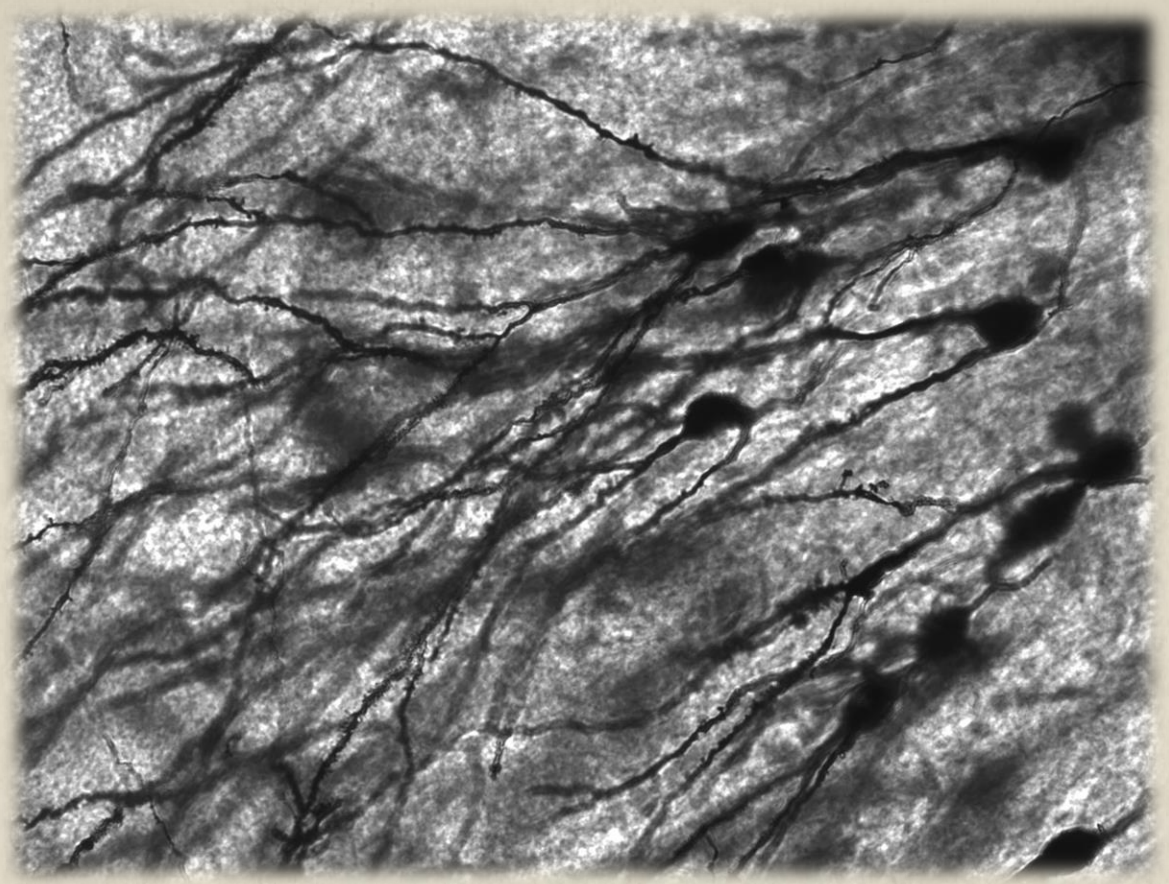
- Make a probabilistic model from the linear combination, we usually get:

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

← Makes votes positive

← Normalizes votes

- Why exponentiate it?
  - One of the reason is for better calculation when doing Maximum Likelihood Estimation.
  - And there are others…
- **Is this good enough?**



Golgi-stained Neurons in Human Hippocampal Tissue

Softmax Function & Exponential Family

iPIN

The Softmax Function

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

Makes votes positive

Normalizes votes

- The distribution is in fact a Softmax function.
  - a generalization of the logistic function
  - usually used in multiclass classification

$$\mu_k = \frac{\exp(\eta_k)}{1 + \sum_j \exp(\eta_j)}$$

- And it is also called *normalized exponential.*

The Exponential Family

- A Broad Class of Distributions
  - Including
    - Bernoulli Distribution
    - Binomial Distribution
    - Poisson Distribution
    - Exponential Distribution
    - Normal Distribution
    - etc.
  - Distributions share many properties in common.

The Exponential Family

- Distribution over x, given parameters η, is defined to be the set of distributions of the form

$$p(x|\eta) = h(x)g(\eta)\exp\{n^T u(x)\}$$

  - x: scalar of vector, discrete or continuous
  - η: natural parameters
  - $g(\eta)$: normalizer

- It can take another form:

$$p(x|\eta) = (1 + \sum_{k=1}^{M-1} \exp(\eta_k))^{-1}\exp(\eta^T x)$$

It's a Softmax!

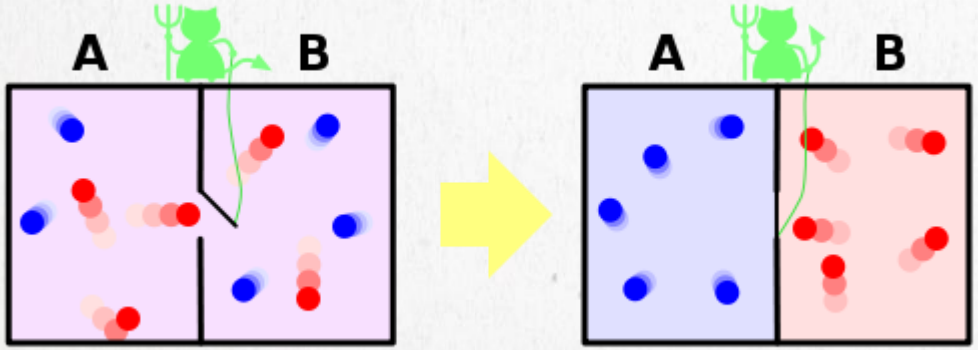
Maxwell's Demon

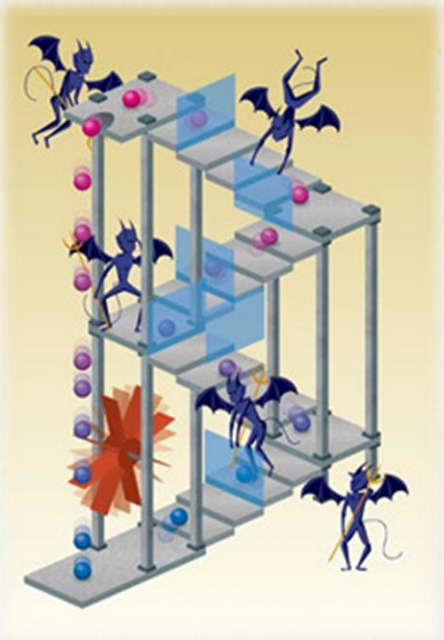
Maximum Entropy Derivation

iPIN

Maxwell's Demon

- A thought experiment which suggests how *Second Law of Thermodynamics* could hypothetically be violated.





Entropy: Thermodynamics

- commonly understood as a measure of ***disorder***.
- According to the second law of thermodynamics the entropy of an isolated system never decreases; such a system will ***spontaneously*** proceed towards thermodynamic equilibrium, the configuration with ***maximum entropy***.

Rudolf Clausius

Entropy: Information Theory

- Shannon Entropy
- the ***Expected Value*** of the Information Contained in Each Message
- Entropy of A Random Variable

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

- A Conditional Entropy

$$H(Y|X) = \sum_{x\in\chi} p(x)H(Y|X=x) = H(x,y) - H(x)$$

$$H(Y|X) = -\sum_{x,y} p(x,y)\log p(y|x)$$

Claude Shannon

Principle of Maximum Entropy

- The principle of maximum entropy states that, subject to precisely stated prior data (such as a proposition that expresses testable information), the probability distribution which best represents the current state of knowledge is the one with largest entropy.

E. T. Jaynes

Examples

- Rolling A Dice

1	2	3	4	5	6
1/6	1/6	1/6	1/6	1/6	1/6
2/15	2/15	1/3	2/15	2/15	2/15
3/20	3/20	1/3	1/15	3/20	3/20

- Minesweeping

Select A Model

- What is the best model under given observations?
  - the ones that fit the observations as many as possible
  - the ones that represent the current state of knowledge

- Why are those models good?
  - The selected distribution is the one that makes the least claim to being informed beyond the stated prior data, that is to say the one that admits the most ignorance beyond the stated prior data.
  - A Measure of Uninformativeness

Building A Maxent Model

- Feature Expectations
  - Empirical Expectation of A Feature
    - $\tilde{p}(f_i) = \tilde{E}(f_i) = \sum_{(c,d) \in observed(C,D)} f_i(c,d)$
  - Model Expectation of A Feature
    - $p(f_i) = E(f_i) = \sum_{(c,d) \in (C,D)} p(c,d) f_i(c,d)$

- A Maxent Model Requires:

$$\tilde{p}(f) = p(f), for\ all\ f$$

Building A Maxent Model

- Select the model p^* with largest entropy from the set C whose models meet the requirements.

$$p^* = \underset{p \in C}{\operatorname{argmax}}\, H(p) = \underset{p \in C}{\operatorname{argmax}} - \sum_{x,y} p(x,y) \log p(y|x)$$

$$= \underset{p \in C}{\operatorname{argmax}} - \sum_{x,y} p(x) p(y|x) \log p(y|x)$$

$$= \underset{p \in C}{\operatorname{argmax}} - \sum_{x,y} \tilde{p}(x) p(y|x) \log p(y|x)$$

$$s.t. \quad p(y|x) \geq 0$$

$$\sum_y p(y|x) = 1$$

$$\sum_{x,y} \tilde{p}(x) p(y|x) f(x,y) = \sum_{x,y} \tilde{p}(x,y) f(x,y)$$

Building A Maxent Model

- It turns out to be a *Nonlinear Programming Problem*
- Process of Solving *Optimization Problem*
  - *Generalized Lagrange Multiplier*
- *It does not necessarily have a optimal solution.*
  - The Karush–Kuhn–Tucker Conditions provide necessary conditions for a solution to be optimal

Generalized Lagrange Multiplier

$$\xi(x,\Lambda,\gamma) = -\sum_{x,y}\tilde{p}(x)p(y|x)\log p(y|x)$$
$$+\sum_{i}\lambda_i[\sum_{x,y}\tilde{p}(x)p(y|x)f_i(x,y) - \sum_{x,y}\tilde{p}(x,y)f_i(x,y)]$$
$$+\gamma[\sum_{y}p(y|x) - 1]$$

$$\frac{\partial\xi}{\partial p(y|x)} = -\tilde{p}(x)[\log p(y|x) + 1] + \sum_i \lambda_i\tilde{p}(x)f_i(x,y) + \gamma \quad ①$$

$$① = 0, \text{we have } p(y|x) = \exp\left[\sum_i \lambda_i f_i(x,y)\right]\exp[\frac{\gamma}{\tilde{p}(x)} - 1] \quad ②$$

$$summrize, we\ have\ \exp\left[\frac{\gamma}{\tilde{p}(x)} - 1\right] = \frac{1}{\sum_y \exp[\sum_i \lambda_i f_i(x,y)]} \quad ③$$

Generalized Lagrange Multiplier

substitute ③ into ②

$$p(y|x) = \exp[\sum_i \lambda_i f_i(x,y)] \frac{1}{\sum_y \exp[\sum_i \lambda_i f_i(x,y)]}$$

$$let\ Z(x) = \sum_y \exp[\sum_i \lambda_i f_i(x,y)]$$

$$p(y|x) = \exp[\sum_i \lambda_i f_i(x,y)] \frac{1}{Z(x)}$$

which means

$$p^*(y|x) = \exp[\sum_i \lambda_i^* f_i(x,y)] \frac{1}{Z^*_{(x)}}\mathrm{c}$$

The Maxent Models

- The best model under the Principle of Maximum Entropy takes the form:

$$p^*(y|x) = \exp[\sum_i \lambda_i^* f_i(x,y)] \frac{1}{Z^*_{(x)}} \mathrm{c}$$

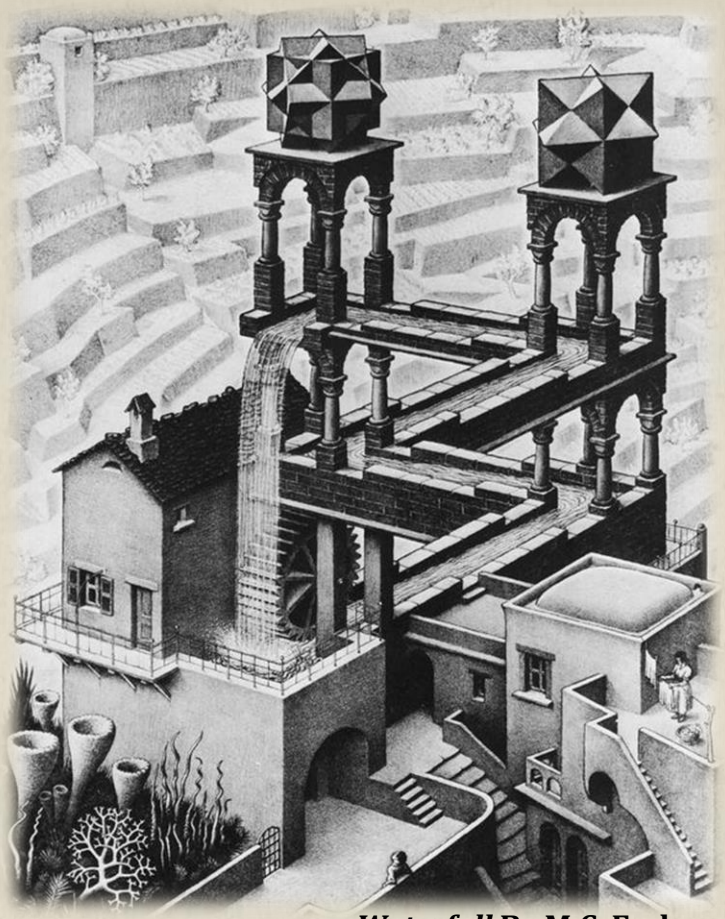
- The Distributions of Exponential Family have a normalized exponential form:

$$p(x|\eta) = (1 + \sum_{k=1}^{M-1} \exp(\eta_k))^{-1} \exp(\eta^T x)$$

- And the Softmax Function takes the form:

$$\mu_\mathrm{k} = \frac{\exp(\eta_k)}{1 + \sum_j \exp(\eta_j)}$$

They are actually the same!



***Waterfall* By M.C. Escher**

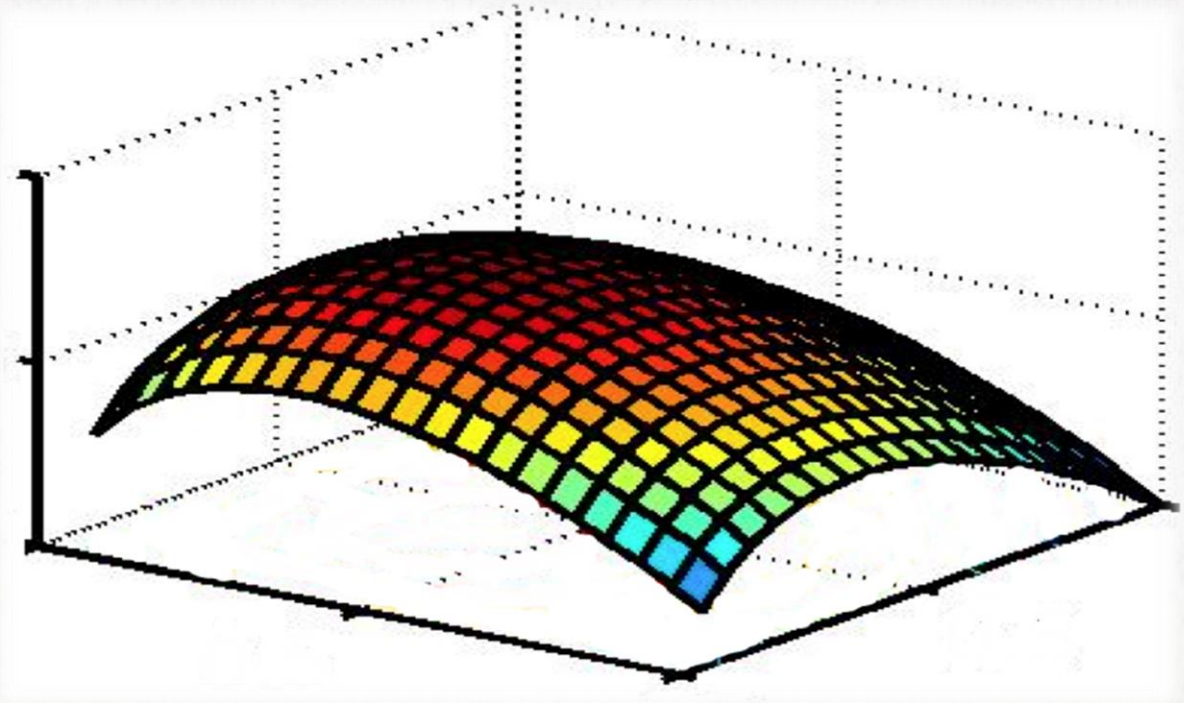
Conclusions

iPIN

Generative vs. Discriminative

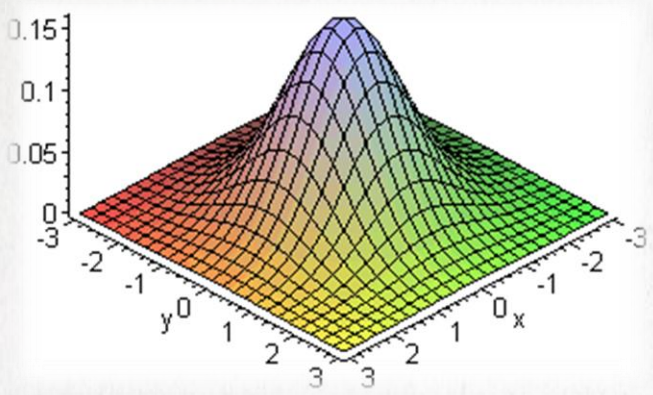
Feature-Based Models

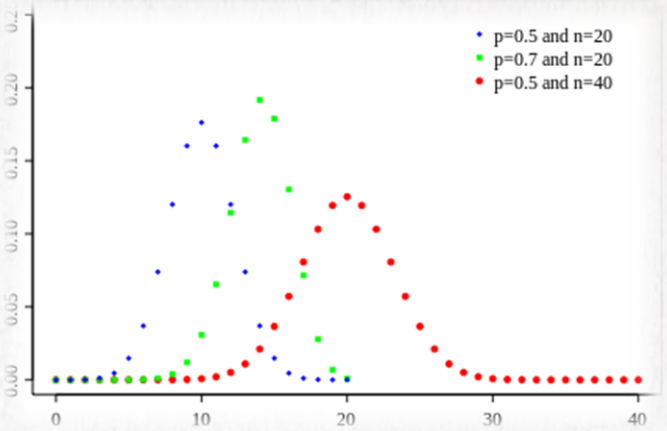
Softmax Function

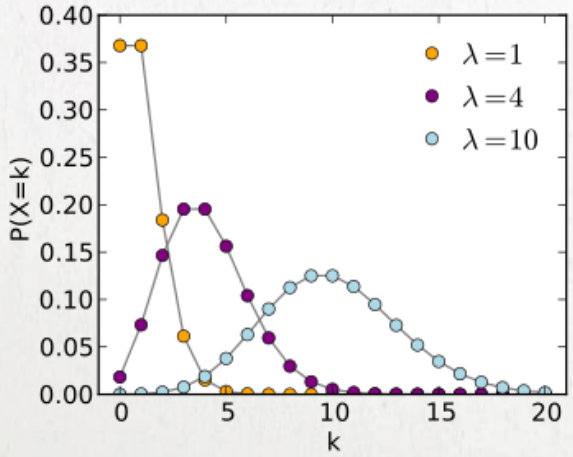


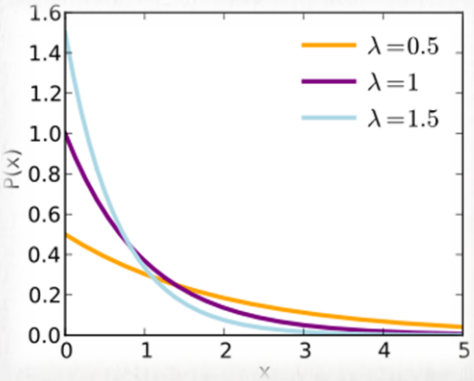
A Likelihood Surface

Exponential Family









Maximum Entropy Derivation

$$p^*(y|x) = \exp[\sum_i \lambda_i^* f_i(x,y)] \frac{1}{Z^*_{(x)}}\text{c}$$