Maximum Entropy Modeling

Jean Mark Gawron

Linguistics
San Diego State University
gawron@mail.sdsu.edu
http://www.rohan.sdsu.edu/∼gawron

Definitions

Entropy

Define the **information** (magnitude) or *surprisal* of an event e to be:

$$\mathsf{I}(e) = -\log p(e)$$

Suppose $p(e_1) = .25$ and $p(e_2) = .125$, and suppose e_1 and e_2 are independent:

$$\begin{array}{rlcl} & \mathsf{I}(e_1) & = & -\log .25 = -\log \frac{1}{4} = --\log 4 = --2 = 2 \\ & \mathsf{I}(e_2) & = & -\log .125 = -\log \frac{1}{8} = --\log 8 = --3 = 3 \\ & \mathsf{I}(e_1 \wedge e_2) & = & -\log .03125 = -\log \frac{1}{32} = --\log 32 = --5 = 5 \\ \hline \therefore & \mathsf{I}(e_1 \wedge e_2) & = & \mathsf{I}(e_1) + \mathsf{I}(e_2) \end{array}$$

Entropy of a distribution p (H(p)) is the **average** surprisal, or the **surprisal value**

$$\mathsf{H}(p) = \sum_e -p(e) \cdot \log p(e)$$

Maximum entropy

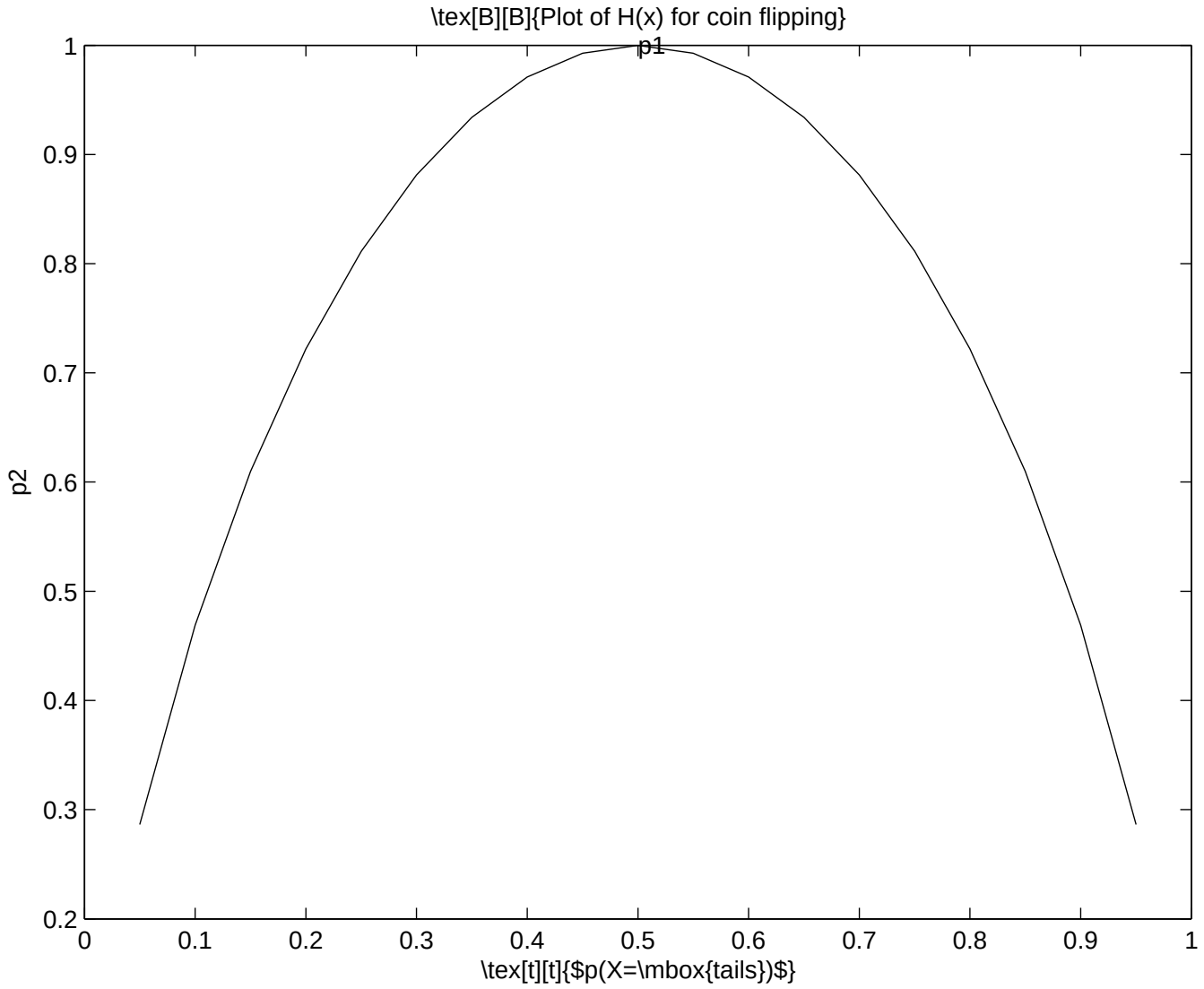
Suppose a coin isn't fair and the prob of heads is 1/4:

$$\begin{aligned} H(p) &= -.25 \cdot \log .25 + -.75 \log .75 && (1)\\ &= .5 + .311 && (2)\\ &= .811 && (3) \end{aligned}$$

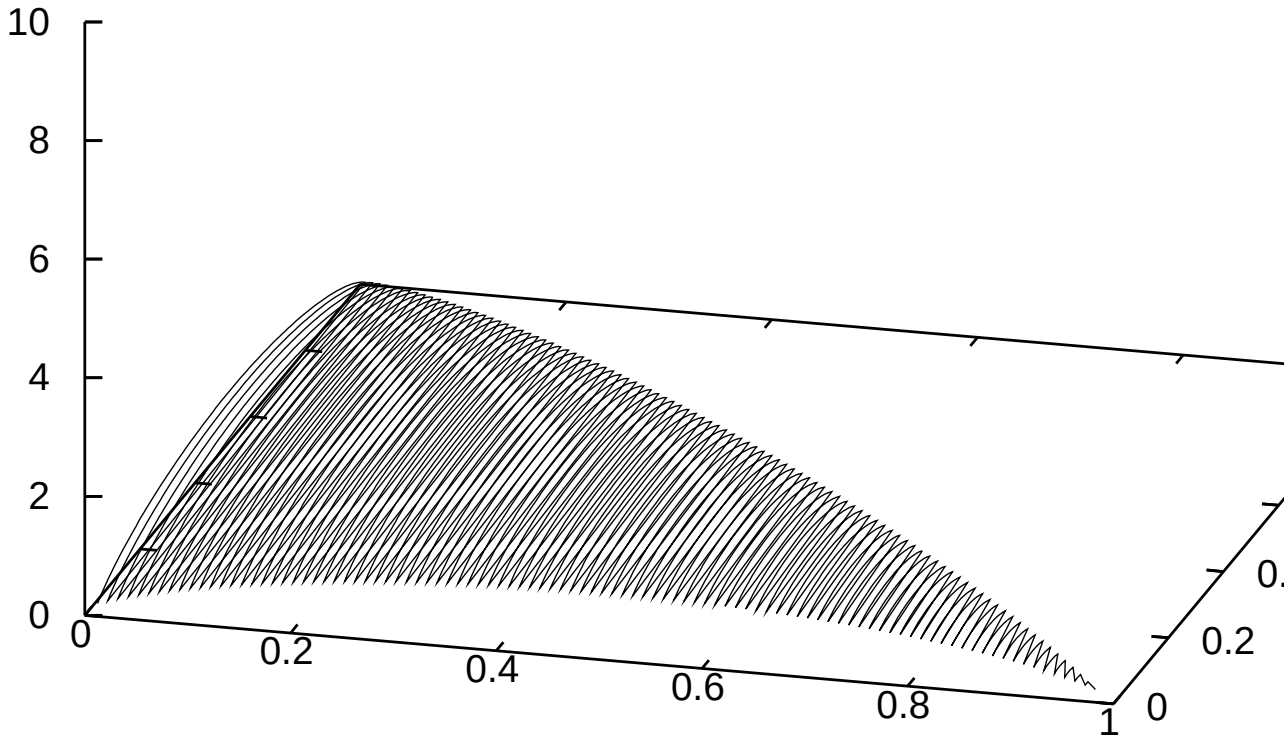
Compared to the fair coin:

$$\begin{aligned} H(p) &= -.5 \cdot \log .5 + -.5 \log .5 && (4)\\ &= .5 + .5 && (5)\\ &= 1 && (6) \end{aligned}$$

The coin's entropy



Entropy of 3-sided coins



Maximum Entropy Distribution

The uniform distribution is the distribution with the **maximum entropy**. This means it is the distribution in which events — on average — carry the most information.

The max entropy distribution captures the situation in which we have the least amount of information about the course of events. Because of this, each event on average carries the most information, delivers the most surprise. Because of this no events are predictable (carry less information).

Berger et al.

In their paper "A Maximum entropy approach to natural language processing", Berger et al. (1990) give the following example, building a statistical model for translating the English word *in* into French:

1. Translators always choose among *dans, en, a, au cours de*, and *pendant*

$$P(dans) + P(en) + P(a) + P(au\ cours\ de) + P(pendant) = 1$$

   With only this information what model shall we pick? Not

$$P(dans) = 1; P(en) = P(a) = P(au\ cours\ de) = P(pendant) = 0$$

2. An **additional** fact:

$$P(dans) + P(en) = .3$$

   Now what model?

3. And still a third fact:

$$P(dans) + P(a) = .5$$

Intuitive answers

$$P(dans) + P(en) + P(a) + P(au\ cours\ de) + P(pendant) = 1$$

$$\downarrow$$

$$P(dans) + P(en) = .3$$

$$\downarrow$$

$$P(dans) + P(a) = .5$$

$$\downarrow$$

;

Intuitive answers

$$P(dans) + P(en) + P(a) + P(au\ cours\ de) + P(pendant) = 1$$

$$\downarrow$$

$$P(dans) = P(en) = P(a) = P(au\ cours\ de) = P(pendant) = .2$$

$$P(dans) + P(en) = .3$$

$$\downarrow$$

$$P(dans) + P(a) = .5$$

$$\downarrow$$

;

Intuitive answers

$$P(dans) + P(en) + P(a) + P(au\ cours\ de) + P(pendant) = 1$$

$$\downarrow$$

$$P(dans) = P(en) = P(a) = P(au\ cours\ de) = P(pendant) = .2$$

$$P(dans) + P(en) = .3$$

$$\downarrow$$

$$P(dans) = P(en) = .15;\ P(a) = P(au\ cours\ de) = P(pendant) = .85/3 = .283$$

$$P(dans) + P(a) = .5$$

$$\downarrow$$

$$;$$

Intuitive answers

$$P(dans) + P(en) + P(a) + P(au\ cours\ de) + P(pendant) = 1$$
$$\downarrow$$
$$P(dans) = P(en) = P(a) = P(au\ cours\ de) = P(pendant) = .2$$

$$P(dans) + P(en) = .3$$
$$\downarrow$$
$$P(dans) = P(en) = .15;\ P(a) = P(au\ cours\ de) = P(pendant) = .85/3 = .283$$

$$P(dans) + P(a) = .5$$
$$\downarrow$$
$$P(dans) = .186; P(en) = .115;\ P(a) = .313;\ P(au\ cours\ de) = P(pendant) = .193$$

$$P(dans) + P(en) + P(a) + P(au\ cours\ de) + P(pendant) = 1.0$$
$$P(dans) + P(en) = .301;\ P(dans) + P(a) = .499$$

Intuitive answers

$$P(dans) + P(en) + P(a) + P(au\ cours\ de) + P(pendant) = 1$$
$$\downarrow$$
$$P(dans) = P(en) = P(a) = P(au\ cours\ de) = P(pendant) = .2$$

$$P(dans) + P(en) = .3$$
$$\downarrow$$
$$P(dans) = P(en) = .15;\ P(a) = P(au\ cours\ de) = P(pendant) = .85/3 = .283$$

$$P(dans) + P(a) = .5$$
$$\downarrow$$
$$P(dans) = .186; P(en) = .115;\ P(a) = .313;\ P(au\ cours\ de) = P(pendant) = .193$$

$$P(dans) + P(en) + P(a) + P(au\ cours\ de) + P(pendant) = 1.0$$
$$P(dans) + P(en) = .301;\ P(dans) + P(a) = .499$$

Why not either of?

$$P(dans) = .15; P(en) = .15;\ P(a) = .35;\ P(au\ cours\ de) = P(pendant) = .35/2 = .175$$
$$P(dans) = .25; P(en) = .05;\ P(a) = .25;\ P(au\ cours\ de) = P(pendant) = .55/2 = .275$$

Max Entropy Principle

Given a collection of facts, choose a model consistent with all the facts, but otherwise as uniform as possible. Berger,et al. p. 42

... the fact that a certain probability distribution maximizes entropy subject to certain constraints representing our incomplete information is the fundamental property which justifies use of that distribution for inference; it agrees with everything that is known but carefully avoids assuming anything that is not known. It is a transcription into mathematics of an ancient principle of wisdom... E.T. Jaynes (Jaynes 1990)

Occam's Razor

Nunquam ponenda est pluralitas sine necesitate. (Assume no more than is necessary.) William of Occam

Document Classification

⟨TITLE⟩COBANCO INC ⟨;CBCO⟩ YEAR NET⟨/TITLE⟩
⟨DATELINE⟩ SANTA CRUZ, Calif., Feb 26 -
⟨/DATELINE⟩⟨BODY⟩Shr 34 cts vs 1.19 dlrs
Net 807,000 vs 2,858,000
Assets 510.2 mln vs 479.7 mln
Deposits 472.3 mln vs 440.3 mln
Loans 299.2 mln vs 327.2 mln
Note: 4th qtr not available. Year includes 1985
extraordinary gain from tax carry forward of 132,000 dlrs, or
five cts per shr.
Reuter
⟨/BODY⟩⟨/TEXT⟩
⟨/REUTERS⟩
⟨REUTERS TOPICS="YES" LEWISSPLIT="TRAIN"
CGISPLIT="TRAINING-SET" OLDID="5555" NE
WID="12"⟩
⟨DATE⟩26-FEB-1987 15:19:15.45⟨/DATE⟩
⟨TOPICS⟩⟨D⟩earn⟨/D⟩⟨D⟩acq⟨/D⟩⟨/TOPICS⟩

Reuters

Category Set	Number of Categories	Number of Categories w/ 1+ Occurrences	Number of Categories w/ 20+ Occurrences
EXCHANGES	39	32	7
ORGS	56	32	9
PEOPLE	267	114	15
PLACES	175	147	60
TOPICS	135	120	57

The TOPICS categories are economic subject categories. Examples include "coconut", "gold", "inventories", and "money-supply".

The EXCHANGES, ORGS, PEOPLE, and PLACES categories correspond to named entities of the specified type. Examples include "nasdaq" (EXCHANGES), "gatt" (ORGS), "perez-de-cuellar" (PEOPLE), and "australia" (PLACES). Typically a document from one of these sets explicitly includes some form of the category name.

Classification

We are interesting in **classifying** things and events in the world.

Is it a good day to play tennis?

Is that word a noun or a determiner?

Is that a Volvo or a Bloody Mary? We shall assume classification is done by way of **attributes** that take numerical values. Is it a sunny day (1 or 0)? Is it the word *platypus*? Does it contain tomato juice?

vectors: There may be many attributes, so that we will assume a **vector** (ntuple) of numerical agttributes representing all the features we need for our classification task.

Context vectors: Such a vector is sometimes called a **context vector** because it summarizes all the attributes in context that matter.

Documents: The context vector for a document will have a 1 or 0 in each position, representing the occurrence (or non-occurrence) of a particular non-stop word.

Profit predicts 'earnings'

I follow the example in

> Manning, Christopher and Schütze, Hinrich. Foundations of Statistical Natural Language Processing, MIT Press. Cambridge, MA: May 1999. Chap 16.2 on Maximum Entropy Modeling (viz.Table 16.6.)

Suppose we are trying to determine for a set of documents whether or not each is in the 'earnings' classification of the Reuters data (an article announcing a company's earning for a year or a quarter).

We shall begin by having one attribute in a context vector, telling us whether the word 'profit' occurs in the article or not.

Features

Features pair classes and attributes, more generally, classes and sets of attributes. Features are defined in terms of context vectors $\vec{x}$ and classes c. Table 16.6 defines 2 features. F1 and F2.

1. F1 = 1 iff an article has the word 'profit' [x=1] AND is classified in earnings [c=1]. Else = 0.
2. F2 = 1 whenever F1 is 0

This satisfies the requirement of a particular learning algorithm called **Generalized Iterative Scaling** [GIS] that the sum of the features values be constant.

Feature summary

$$f_i(\vec{x}^j, c) = \begin{cases} 1 & \text{if } s_{ij} > 0 \text{ and } c = 1 \\ 0 & \text{otherwise} \end{cases}$$

s_{ij} is the term weight for word i in Reuters article j. It is positive when word i occurs in article j.

Table 16.6 [subpart]:

x	c	F1	F2
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

Examples

Two distributions

Maxe

x/c	oth	earn
oth	0.2	0.2
profit	0.2	0.4

Mide

x/c	oth	earn
oth	0.1	0.4
profit	0.1	0.4

Empirical expectation of a feature

The empirical expectation of feature f_i in a corpus of size K:

$$\mathsf{E}(f_i) = \frac{\sum_{j=1}^{\mathrm{K}} f_i(\vec{x}_j, c_j)}{\mathsf{K}} \quad (7)$$

$$= p(s_i > 0, c) \quad (8)$$

Expectation of a feature

The expectation of feature f_i according to distribution p over context vectors and classes is:

$$\mathsf{E}(f_i) \quad = \quad \sum_{\vec{x},c} p(\vec{x},c) \cdot f_i(\vec{x},c) \tag{9}$$

We seek distributions such that the expected value they assign to our features **equals** the empirical expected value of the corpus.

Example: expected value

The two distributions maxe and mide have the same expected values for feats f1 and f2.

$$\text{maxe}(\text{profit}, \text{earn}) = .4 \quad \text{E}(f1).4, E(f2) = .6$$
$$\text{mide}(\text{profit}, \text{earn}) = .4 \quad \text{E}(f1).4, E(f2) = .6$$

Two corpora

Now consider two corpora::

A ((other,other),(other,earnings),(profit,other),(profit,earnings), (profit,earnings))

B ((other,other),(other,earnings),(profit,other),(profit,earnings),(profit,earnings), (other,earnings),(other,earnings),(other,earnings),(profit,earnings),(profit,earnings))

A and B give the same empirical expected values for features F1 and F2

[E(F1)=.4,E(F2)=.6]

The maximum entropy choice

Leaving out the probability mass for the event (1,1), Maxe is a uniform distribution. Mide is lumpier. The more uniform a distribution, the higher its entropy:

Maxe

x/c	oth	earn
oth	0.2	0.2
profit	0.2	0.4

H(Maxe)= 1.92

Mide

x/c	oth	earn
oth	0.1	0.4
profit	0.1	0.4

H(Mide)= 1.72

An even higher entropy choice

There is of course an even more uniform joint distribution around:
Unie

x/c	oth	earn
oth	0.25	0.25
profit	0.25	0.25

H(Unie) = 2.0

Expected value again

But unie is not a contender for the max ent crown if the data set is either A or B. The constraint on all the distributions we are considering is that they preserve the expected values for the features that we observe in the corpus. But for Unie:

$$E(f1) = 0.25$$

Which is not the value we observe in either data set A or B (E(F1)=0.4). maxe and and mide get this right, so they are in (at least contending) and unie is out.

The right choice

Maxe

x/c	oth	earn
oth	0.2	0.2
profit	0.2	0.4

H(Maxe)= 1.92

Notice that despite its uniform value for most events, maxe has one 'probability lump' at (1,1). Now the virtue of the one probability lump is clear: That 0.4 probability serves to get the expected value of F1 right. F1 cares only about the event (1,1). It is indifferent to how the rest of the probability mass is distributed. maxe is the distribution which gets the F1 contraint right and mashes the rest of the probability out as uniformly as possible.

Max entropy form

The max ent equation (16.4):

$$
\begin{aligned}
p(\vec{x}, c) &= 1/Z * \alpha_1^{f_1(x,c)} * \alpha_2^{f_2(x,c)} \\
&= 1/Z * e^{w_1 * f_1(x,c) + w_2 * f_2(x,c)} \\
&= 1/Z \prod_{i=1}^{\mathrm{K}} e^{w_i * f_i(\vec{x},c)} \\
\log p(\vec{x}, c) &= -\log Z + \sum_{i=1}^{\mathrm{K}} w_i * f_i(x, c) \\
&= -\log Z + w_1 * f_1(x, c) + w_2 * f_2(x, c) \quad \text{two feats}
\end{aligned}
$$

You need to know:

- Various optimization algorithms (e.g., Generalized Iterative Scaling = GIS) will converge on the maximum likelihood model of this form (more later)
- **Can be shown:** The maximum likelihood model of this form is also the maximum entropy model that respects the empirical distribution of the features.

Comparing Models I

Compare probabilities assigned to the corpus B (maximum likelihood criterion):

Mide 6.55×10^{-6} $.1*.4*.1*.4*.4*$

$.4*.4*.4*.4*.4$

Maxe 1.64×10^{-6} $.2*.2*.2*.4*.4*$

$.2*.2*.2*.4*.4$

Maxe

x/c	oth	earn
oth	0.2	0.2
profit	0.2	0.4

Mide

x/c	oth	earn
oth	0.1	0.4
profit	0.1	0.4

(other,other),(other,earnings),(profit,other),(profit,earnings),(profit,earnings),

(other,earnings),(other,earnings),(other,earnings),(profit,earnings),(profit,earnings)

Comparing Models II

	Corpus A	Corpus B
Maximum Entropy	maxe	maxe
Maximum Likelihood	maxe	mide

A ((other,other),(other,earnings),(profit,other),(profit,earnings), (profit,earnings))

B ((other,other),(other,earnings),(profit,other),(profit,earnings),(profit,earnings), (other,earnings),(other,earnings),(other,earnings),(profit,earnings),(profit,earnings))

Maxe

x/c	oth	earn
oth	0.2	0.2
profit	0.2	0.4

H(Maxe)= 1.92

Mide

x/c	oth	earn
oth	0.1	0.4
profit	0.1	0.4

H(Mide)= 1.72

A question

The fact that Maxe assigns a lower probability to corpus B than Mide means it is does not capture some important information about Corpus B (for example, that (other,earnings) is 4 times as frequent as (other,other)).

Nevertheless the GIS algorithm that picks Maxe over Mide is just doing its job, which is to pick the maximum entropy model that captures the expected values of its features.

Investigating Mide

Suppose we tried to have a model for dist Mide (omitting Z...)

$$p(other, other) = 0.1 = e^{\alpha \cdot F1 + \beta \cdot F2} \quad (10)$$

$$p(other, earnings) = 0.4 = e^{\alpha \cdot F1 + \beta \cdot F2} \quad (11)$$

$$p(profit, other) = 0.1 = e^{\alpha \cdot F1 + \beta \cdot F2} \quad (12)$$

$$p(profit, earnings) = 0.4 = e^{\alpha \cdot F1 + \beta \cdot F2} \quad (13)$$

But neither F1 nor F2 distinguishes (other,other) from (other,earnings)

x	c	F1	F2
other	other	0	1
other	earnings	0	1
profit	other	0	1
profit	earnings	1	0

Therefore there is no model of this form with these features for distribution Mide.

The only model

Valid models for this feature set must assign the same probability to the 3 cases that can't be distinguished by the features:

$$P(other,\, other) = P(other,\, earnings) = P(profit,\, other)$$

$$\begin{array}{lll} \text{Also,} & P(profit, earnings) & = .4 \\ \text{And,} & P(other,\, other) + P(other,\, earnings) + & = 1 \\ & P(profit,\, other) + P(profit,\, earnings) & \end{array}$$

$$\text{Therefore, } .6 = P(other,\, other) + P(other,\, earnings) + P(profit,\, other) \quad (14)$$

$$.2 = P(other,\, other) = P(other,\, earnings) = P(profit,\, other) \quad (15)$$

Thus, the only distribution that satisfies these constraints is Maxe. It is, therefore, trivially the max entropy dist that satisfies these constraints.

Models: Feature sets

The point to take home is that the same corpus will yield different ME models with different feature sets. Indeed it is trivial to add a feature that will cause distribution Mide to be favored. Let's try. Consider features F1, F2, as follows:

x	c	F1	F2
other	other	0	1
other	earnings	0	1
profit	other	0	1
profit	earnings	1	0

GIS $\rightarrow$

x	c	F1	F2	F3	F4
other	other	0	1	1	0
other	earnings	0	1	0	1
profit	other	0	1	0	1
profit	earnings	1	0	0	1

To use GIS, in order to add F3 we must also add F4, because the sum of the feature values must always be constant (1). So if the corpus is B, distribution Maxe is no longer even a contender, because it does not respect the constraints introduced by F2.

Note there are still two rows that can't be distinguished.

Fully discriminating feature set

The point to take home is that the same corpus will yield different ME models with different feature sets. Indeed it is trivial to add a feature that will cause distribution Mide to be favored. Let's try. Consider features F1, F2, as follows:

x	c	F1	F2	F3	F4	F5	F6
other	other	0	1	1	0	0	1
other	earnings	0	1	0	1	0	1
profit	other	0	1	0	1	1	0
profit	earnings	1	0	0	1	0	1

Note all rows can now be distinguished.

The moral

Null hypothesis: a fully discriminative feature set

Using a fully discriminative feature set will capture all the constraints with respect to the attributes and classes you encode (profit, earning).

Not all the features of a fully discriminative model may be necessary for the task at hand.

Berget et al. (1992), Section 4, Feature selection algorithm

Context, Feature, Event I

Two important variations on the discussion here:

- ME models are often used to predict the Conditional distribution ($P(c \mid \vec{x})$), rather than the joint distribution ($P(c, \vec{x})$).
- More importantly, ME models can be used for more than just classification, they can be used to predict all kinds of hidden information.
- The hidden information view is really a generalization of the classification view;
- On the hidden information view, we use the term **event** for the pairing of the hidden information with the context. Each event determines its context uniquely.
  - In tagging, the contexts are words, and the events are word, tag pairs.
  - In parsing, the contexts are words, and the events are full trees containing the words
- Features
  - are functions from events to real numbers (since an event determines a context uniquely);
  - We sometimes speak of one feature function f from an event to a vector of d values; so f returns the values for d features.
- A model Θ is a d-dimensional vector containing the weights for the d features.

Context, Feature, Event II

Tagging (Ratnaparkhi 1998)

- Set of events E: sets of word+tag paired with context. For example, we might use the event e_1

$$e_1 = \langle \text{VB, run, to/TO \_\_\_ quickly/RB} \rangle$$

  to represent the fact that in the string *to run quickly*, the word *run* was tagged as VB in the corpus (with *to* and *quickly* tagged as TO and RB, respectively).

- Set of contexts W: W is a set of words paired with surrounding words for some model dependent window (say 1 word in either direction)

$$\langle \text{run, to/TO \_\_\_ quickly/RB} \rangle$$

Context, Feature, Event III

Tagging (Ratnaparkhi 1998), ctd.

- A feature function f

$$f : \mathsf{E} \rightarrow \mathcal{R}^d$$

That is, f is a function from events like e_1 to their characterization as a vector of features. For example, If

$$e_1 = \langle \text{VB, run, to/TO \_\_\_ quickly/RB} \rangle$$

$f(e_1)$ might be:

$$\begin{bmatrix} \text{PREV-TAG-IS-TO} & 1 \\ \text{WORD-IS-RUN} & 1 \\ \text{WORD-IS-FLY} & 0 \\ \ldots & \\ \text{NEXT-TAG-IS-RB} & 1 \end{bmatrix}$$

Some tagging feats

$$f_1(\vec{x}^i, c) = \begin{cases} 1 & \text{if word}_i = \text{“race”} \& c = \text{NN} \\ 0 & \text{otherwise} \end{cases}$$

$$f_5(\vec{x}^i, c) = \begin{cases} 1 & \text{if word}_i = \text{“race”} \& c = \text{VB} \\ 0 & \text{otherwise} \end{cases}$$

$$f_2(\vec{x}^i, c) = \begin{cases} 1 & \text{if } t_{i-1} = \text{TO} \& c = \text{VB} \\ 0 & \text{otherwise} \end{cases}$$

$$f_3(\vec{x}^i, c) = \begin{cases} 1 & \text{if suffix(word}_i) = \text{“ing”} \& c = \text{VBG} \\ 0 & \text{otherwise} \end{cases}$$

$$f_4(\vec{x}^i, c) = \begin{cases} 1 & \text{if is\_lower\_case(word}_i) \& c = \text{VB} \\ 0 & \text{otherwise} \end{cases}$$

Form of the model again

Using w for contexts (avoiding c, which suggests classes)

$$q_\Theta(e \mid w) = \frac{1}{Z} e^{\Theta \cdot f(e)}$$

- Θ: the vector of feature weights.
- $\Theta \cdot f(e)$: the dot product of the feature weights and feature vector of e

To be explicit about Z, the normalization factor, it's useful to have Y, a function from contexts W to subsets of E. So in the tagging example, if w is a sequence of words,Y(w) is a set of events like e_1, basically the set of all possible taggings and tagging contexts of w.

$$q_\Theta(e \mid w) = \frac{e^{\Theta \cdot f(e)}}{\sum_{y \in Y(w)} e^{\Theta \cdot f(y)}}$$

The denominator is the sum of all the weighted scores of all the ways of tagging w

ME models and others

Predicting w/Features

Regression

Map some input features into some output value: output is real-valued.

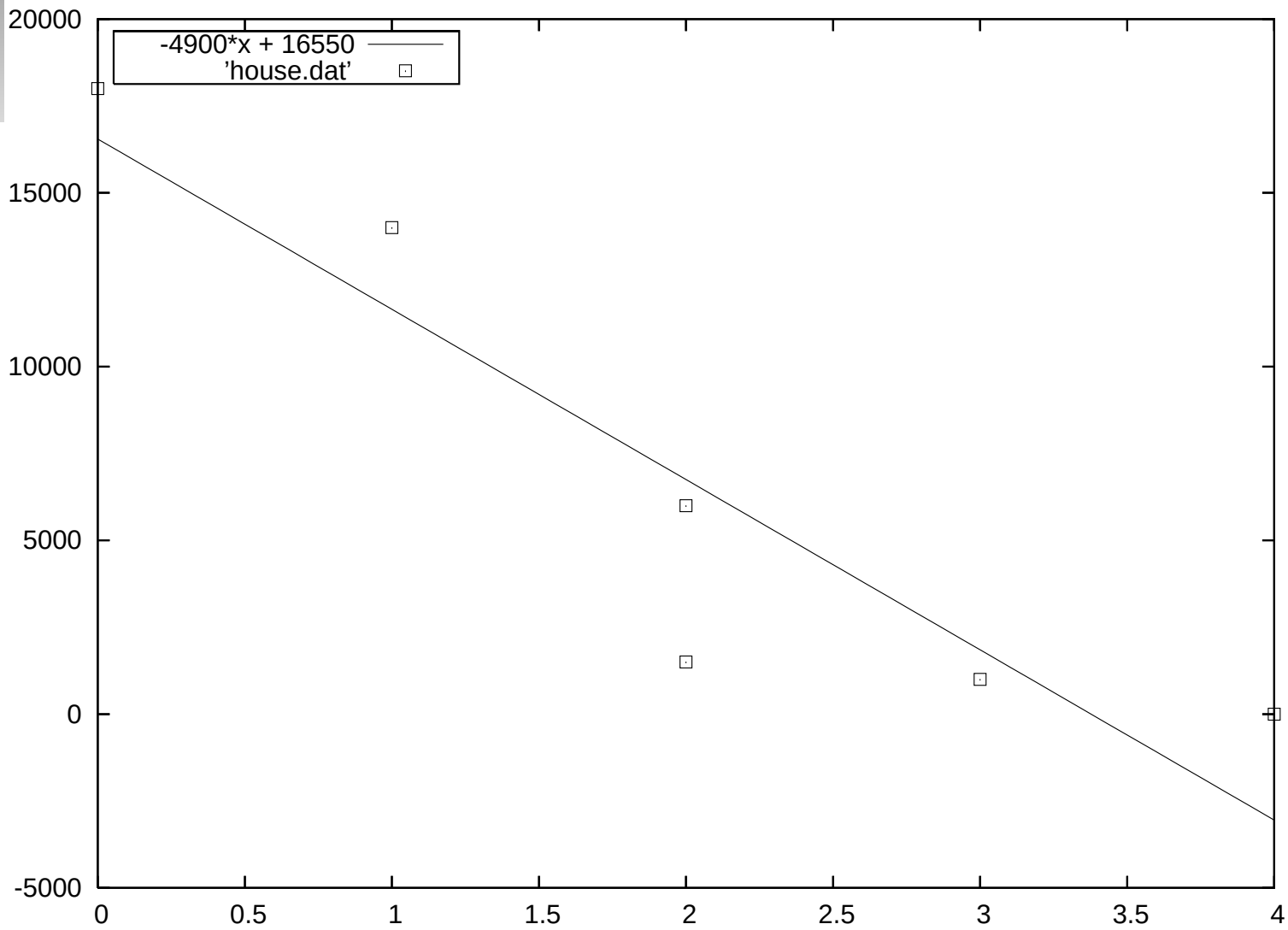
Levitt and Dubner predict house prices based on number of vague adjectives (*fantastic, cute, charming*, . . .)

# of vague adjectives	amount house sold over asking price
4	0
3	1000
2	1500
2	6000
1	14000
0	18000

Q: Why amount over asking and not just amount?

A: Predict a function f s/t: $-\infty < f(x) < \infty$

Least squares fit



Weights

Equation for our best fit line:

$$y = 16550 + -4900x$$

Think of the numbers as weights for features:

$$\begin{array}{cccccc} & & w_0 & & w_1 \\ y & = & 16550 * 1 & + & -4900 * x \end{array}$$

The features:

	Feature	Weight	Values
f_0	Intercept	16550	Always 1
f_1	NumAdjectives	-4900	0-20

Many Features

Many **predictive factors**:

	Feature	Weight	Values
f_0	InterceptFeat	16550	Always has value 1
f_1	NumAdjectives	-4900	0-20
f_2	Mortgage Rate	$??$	0-25
f_3	Num Unsold Houses	$??$	0-200,000

A house-sale event is a vector of features

Intercept, NumAdjectives, MortgageRate, Num Unsold Houses

(1 , 1 , 6.5 , 10,000)

Linear regression

The Model: price $= \sum_{i=0}^{\text{N}} w_i \times f_i = w \cdot f$

More generally, predict quantity y for each of M x's in a corpus.

For the jth instance, $x^{(j)}$:

$$y_{pred}^{(j)} = \sum_{i=0}^{\text{N}} w_i \times f_i^{(j)}$$

Error for the jth instance:

$$y_{pred}^{(j)} - y_{obs}^{(j)}$$

Cost function for the set of weights (**sum-squared error**)

$$\text{Cost(W)} = \sum_{j=0}^{M} (y_{pred}^{(j)} - y_{obs}^{(j)})^2$$

Function for computing W that minimizes sum-squared error available in R.

Logistic Regression

Linear Models again

We continue learning a linear function (a linear separator), though that function may be a function of many variables, which turns our line into a hyperplane.

- 2 features: Points “above” the line are classified one way; points below the other
- 3 features: Points “above” the plane are classified one way; points below the other
- 4 features: Dont try to visualize it!

Classification by probability

Instead of a real-valued function that returns costs above or below a certain line, the real-valued function we'd like to learn is a probability function:

$$p(c \mid \vec{x}) : \Sigma^n \rightarrow [0, 1]$$

Extending the range

A linear separator should be a function whose range is positive/negative infinity, in other words, a range with no bounds:

$$f : \Sigma^n \rightarrow [-\infty, \infty]$$

Step one: switch to an odds function:

$$f_{\text{ODDS}} : \Sigma^n \rightarrow \frac{p(c \mid \vec{x})}{1 - p(c \mid \vec{x})}$$

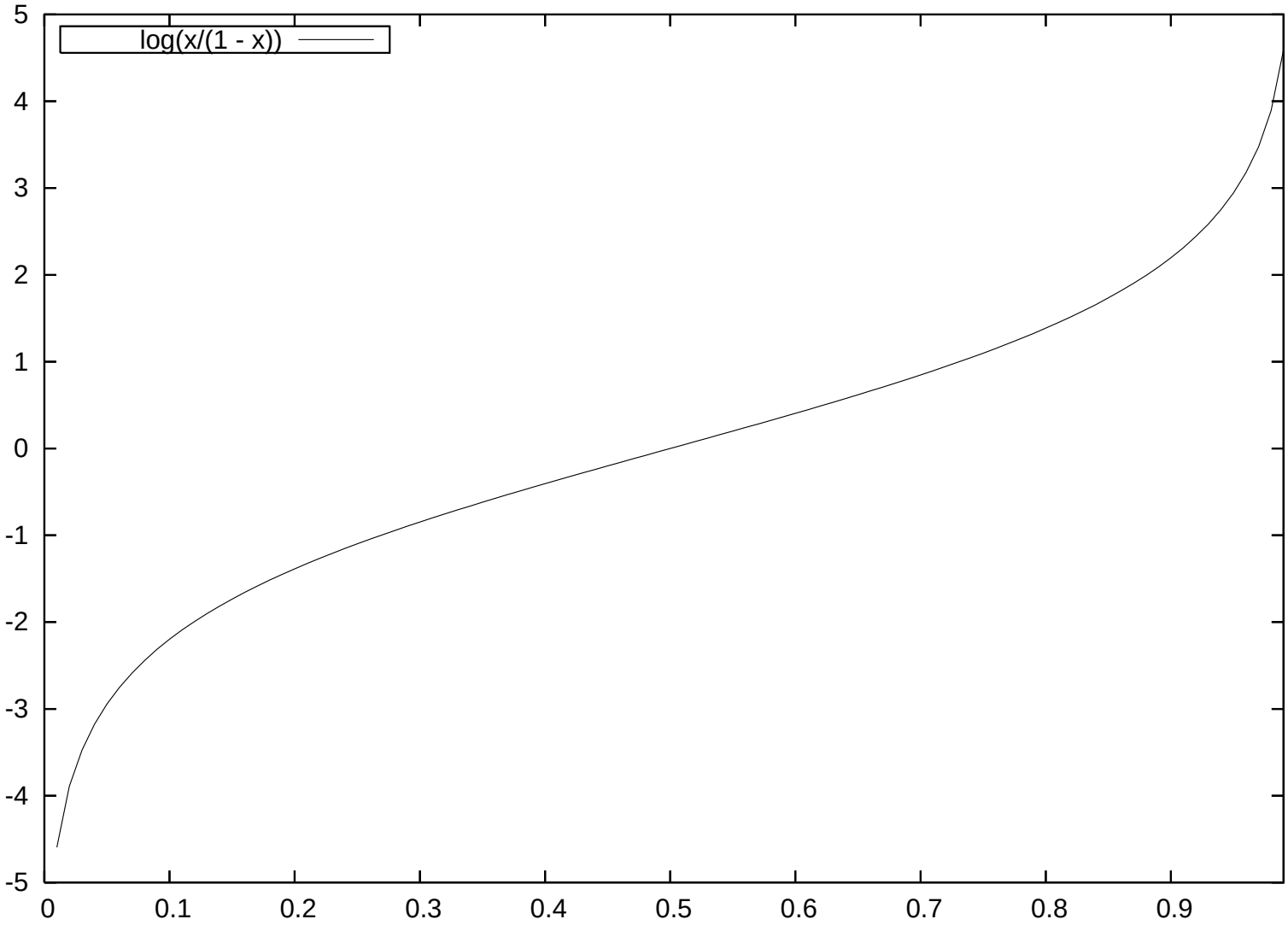
But the range is still $[0, \infty]$. Mapping to log probs takes us the rest of the way.

Step two: switch to log probs.

$$\text{LOGIT} : \Sigma^n \rightarrow \log \frac{p(c \mid \vec{x})}{1 - p(c \mid \vec{x})}$$

Positive means the odds are greater than 1, negative means less than 1; 0 means 1 ($p(c \mid \vec{x}) = .5$)

The logit function



Map $[0,\ 1]$ to $[-\infty,\ \infty]$, wheeling around the value .5. logit(.5)=0. All values to the right are positive, all values to the left negative.

Logistic function

Plug p into the logit function and solve for p to derive what's called the **logistic function**:

$$
\begin{aligned}
\ln \frac{p}{1-p} &= \text{logit(p)} \\
\frac{p}{1-p} &= e^{\text{logit(p)}} \\
p &= e^{\text{logit(p)}}(1-P) \\
p &= e^{\text{logit(p)}} - p \cdot e^{\text{logit(p)}} \\
p + p \cdot e^{\text{logit(p)}} &= e^{\text{logit(p)}} \\
p(1 + e^{\text{logit(p)}}) &= e^{\text{logit(p)}} \\
p &= \frac{e^{\text{logit(p)}}}{1 + e^{\text{logit(p)}}} \times \frac{e^{-\text{logit}(p)}}{e^{-\text{logit}(p)}} \\
p &= \frac{1}{1 + e^{-\text{logit(p)}}}
\end{aligned}
$$

The logistic model

Summarizing:

We learn weights $\vec{w}$ and a linear equation to predict logit(p) and use the logistic equation to find a probability:

$$\begin{aligned}
\text{logit}(p) &= \vec{w} \cdot \vec{x} \\
p(c \mid \vec{x}) &= \frac{e^{\text{logit(p)}}}{1 + e^{\text{logit}(p)}} \\
p(c \mid \vec{x}) &= \frac{e^{\vec{w} \cdot \vec{x}}}{1 + e^{\vec{w} \cdot \vec{x}}} \\
&= \frac{1}{1 + e^{-\vec{w} \cdot \vec{x}}}
\end{aligned}$$

Advantage: our classifier produces a probability distribution over all classes, useful in larger NLP tasks.

Learning

So now we learn logit of p using least squares, right?

Wrong! This would work well if we had corpora with events of the form

$$\langle \vec{x},\ P(c \mid \vec{x}) \rangle$$

The way we have houses paired with prices. But events rarely come annotated with their probabilities!

And if we had such corpora, it wouldn't be such a big deal learning p(x).

We simulate having such a corpus by choosing the set of weights that maximizes the overall probability of the corpus (more precisely, of the feature representation of it)

Maximum Likelihood

We want the set of weights that maximizes the log probability of the features we observe in all corpus events:

$$\widehat{w} = \underset{w}{\operatorname{argmax}} \sum_i \log \begin{cases} P(c^j \mid \vec{x}^j) & \text{if } c^j = 1 \\ 1 - P(c^j \mid \vec{x}^j) & \text{if } c^j = 0 \end{cases}$$

Since c is always either 1 or 0, this is neater:

$$\widehat{w} = \underset{w}{\operatorname{argmax}} \sum_i c^j \log P(c^j \mid \vec{x}^j) + (1 - c^j) \log(1 - P(c^j \mid \vec{x}^j))$$

Conditional Likelihood of Model Θ

$$\mathsf{L}(\Theta) = \sum_{c,\vec{x}} p(c, \vec{x}) \cdot q_\Theta(c \mid \vec{x})$$

The max ent result

The Maximum Likely Estimate (MLE) of a multinomial logistic regression model is also the Maximum Entropy (ME) model of that form.

Maximum Entropy models are a generalization of multinomial logistic regression models.

It can be shown that any multinomial logistic regression model can be put into the maximum entropy form (Ratnaparkhi 1998)

Learning

Malouf (2002)

Shallow Parsing Dataset (Bouma et al. 2001, a large set of fine-grained lexical features)

classes	features
8,625,782	264,142

Model	Accuracy	**Iterations**	**Time** (secs)
Generalized Iterative Scaling (GIS)	14.2	3494	21,223
Improved Iterative Scaling (IIS)	5.4	3264	66,855
Steepest Ascent	26.7	3677	85,062
Conjugate Gradient (fr)	24.7	1157	39,038
Conjugate Gradient (prp)	24.7	536	16251
limited memory variable metric	23.8	403	2420