Maximum Entropy Models for Natural Language Processing

James Curran
The University of Sydney
james@it.usyd.edu.au

6th December, 2004

Overview

- a brief probability and statistics refresher
  - statistical modelling
  - Naïve Bayes
- Information Theory concepts
  - uniformity and entropy
- Maximum Entropy principle
  - choosing the most uniform model

Overview

- Maximum Entropy models
  - Features and Constraints
  - Maximising Entropy
  - Alternative formulation
- Estimating Maximum Entropy models
  - GIS, IIS, conjugate gradient, quasi-Newtonian methods
  - smoothing techniques
- Applications

  . . .

Statistical Modelling

- given a set of **observations** (i.e. *measurements*):
  $\Longrightarrow$ extract a mathematical description of observations
  $\Longrightarrow$ **statistical model**
  $\Longrightarrow$ use this for **predicting** future observations
- a statistical model should:
  - **represent faithfully** the original set of measurements
  - **generalise sensibly** beyond existing measurements

Faithful Representation

- trivial **if no generalisation** is required
  just look up the relative frequency directly
- trust the training data exclusively
- but unseen observations are impossible
  since relative frequency is zero
- and **most observations** are unseen

  $\Longrightarrow$ **practically useless!!**

Sensible Generalisation

- want to find correct distribution given seen cases
  i.e. to minimise error in prediction
- **sensible** is very hard to pin down
- may be based on some hypothesis about the problem space
- might be based on attempts to account for unseen cases

  $\Longrightarrow$ **generalisation reduces faithfulness**

Example: Modelling a Dice Roll

- consider a single roll of a 6-sided dice
- **without any extra information** (any measurements)
- what is the probability of each outcome?
- **why do you make that decision?**

Example: Modelling a Biased Dice Roll

- now consider observing lots (e.g. millions) of dice rolls
- **imagine the relative frequency of sixes is unexpectedly high**

$$P(6) = 1/3$$

- now what is the probability of each outcome?
- **why do you make that decision?**

Uniform Distribution

- generalisation without any other information?
- **most sensible choice is uniform distribution of mass**
- when all mass is accounted for by observations
  we must redistribute mass to allow for unseen events
- i.e. take mass from seen events to give to unseen events

Example: Modelling a Complex Dice Roll

- we can make this much more complicated
- $P(6) = 1/3$, $P(4) = 1/4$, $P(2 \textit{ or } 3) = 1/6, \ldots$
- impossible to visualise uniformity
- impossible to **analytically distribute mass uniformly**

Entropy

$$-\sum_{\mathbf{x}} \mathbf{p}(\mathbf{x}) \log_2 \mathbf{p}(\mathbf{x})$$

- **Entropy is a measure of uncertainty of a distribution**
- *higher the entropy the more uncertain a distribution is*
- entropy matches out intuitions regarding uniformity
  i.e. it measures uniformity of a distribution

  but applies to distributions in general
- also a measure of the number of alternatives

Maximum Entropy principle

- *Maximum Entropy modelling*:
  - predicts observations from training data
    (faithful representation)
  - this **does not uniquely identify the model**
- *chooses the model which has the most uniform distribution*
  - i.e. the **model with the maximum entropy**
    (sensible generalisation)

Features

- **features encode observations from the training data**
- include the class for classification tasks

(title caps, **NNP**)	Citibank, Mr.
(suffix `-ing`, **VBG**)	running, cooking
(POS tag **DT**, **I-NP**)	the bank, a thief
(current word `from`, **I-PP**)	from the bank
(next word `Inc.`, **I-ORG**)	Lotus Inc.
(previous word `said`, **I-PER**)	said Mr. Vinken

Complex Features

- features can be arbitrarily complex
  - e.g. document level features
    (document = `cricket` & current word = `Lancashire`, **I-ORG**)
    $\Longrightarrow$ hopefully tag `Lancashire` as **I-ORG** not **I-LOC**
- features can be combinations of atomic features
  - (current word = `Miss` & next word = `Selfridges`, **I-ORG**)
    $\Longrightarrow$ hopefully tag `Miss` as **I-ORG** not **I-PER**

Features in Maximum Entropy Models

- Features encode elements of the context C useful for predicting klass t

- Features are binary valued functions (**not true**), e.g.

$$f_i(C,t) = \begin{cases} 1 & \text{if } \texttt{word}(C) = \texttt{Moody} \;\; \& \;\; t = \textbf{I-ORG} \\ 0 & \text{otherwise} \end{cases}$$

- $\texttt{word}(C) = \texttt{Moody}$ is a *contextual predicate*

- identify (`contextual_predicate, tag`) pairs in classification tasks

The Model

$$p(t|C) = \frac{1}{Z(C)} \exp\left(\sum_{i=1}^{n} \lambda_i f_i(C,t)\right)$$

- f_i is a feature

- λ_i is a weight (large value implies informative feature)

- $Z(C)$ is a normalisation constant ensuring a proper probability distribution

- Also known as a *log-linear* model

- Makes no independence assumptions about the features

Model Estimation

$$p(t|C) = \frac{1}{Z(C)} \exp\left(\sum_{i=1}^{n} \lambda_i f_i(C,t)\right)$$

- Model estimation involves setting the weight values λ_i
- The model should reflect the data
  $\Longrightarrow$ **use the data to *constrain* the model**
- What form should the constraints take?
  $\Longrightarrow$ **constrain the *expected value* of each feature** f_i

The Constraints

$$E_p f_i = \sum_{C,t} p(C,t) f_i(C,t) = K_i$$

- Expected value of each feature must satisfy some constraint K_i
- **A natural choice for** K_i **is the average empirical count:**

$$K_i = \quad E_{\tilde{p}} f_i = \frac{1}{N} \sum_{j=1}^{N} f_i(C_j, t_j)$$

derived from the training data $(C_1, t_1), \ldots, (C_N, t_N)$

Choosing the Maximum Entropy Model

- **The constraints do not *uniquely* identify a model**
- From those models satisfying the constraints:
  choose the Maximum Entropy model
- The maximum entropy model is the *most uniform model*
  $\Longrightarrow$ makes no assumptions in addition to what we know from the data
- Set the weights to give the MaxEnt model satisfying the constraints

The Other Derivation

- start with a log-linear model:

$$p(t|C) = \frac{1}{Z(C)} \exp\left(\sum_{i=1}^{n} \lambda_i f_i(C, t)\right)$$

- the *Maximum Likelihood Estimate* for these forms of models ...
- also happens to be the Maximum Entropy Model

two completely independent justifications!

Finding Maximum Entropy Model

Three approaches to solving the constrained optimisation problem:

- Generalised Iterative Scaling (GIS)
- Improved Iterative Scaling
- direct constrained optimisation, e.g.:
  - conjugate gradient
  - limited memory BFGS

progressively improving speed of convergence

GIS in Practice

Stephen Clark and I have:

- proved that there is no need for correction feature
- showed with clever implementation GIS is fast
- showed that GIS converges fast enough for many NLP tasks

Smoothing

- Models which satisfy the constraints exactly tend to *overfit* the data
- In particular, empirical counts for low frequency features can be unreliable
  - often leads to very large weight values
- Common smoothing technique is to ignore low frequency features
  - **but low frequency features may be important**
- Use a *prior* distribution on the parameters
  - **encodes our knowledge that weight values should not be too large**

Gaussian Smoothing

- We use a *Gaussian prior* over the parameters
  - penalises models with extreme feature weights
- This is a form of *maximum a posteriori* (MAP) estimation
- Can be thought of as relaxing the model constraints
- Requires a modification to the update rule

Tagging with Maximum Entropy Models

- The conditional probability of a tag sequence $t_1 \ldots t_n$ is

$$p(t_1 \ldots t_n | w_1 \ldots w_n) \approx \prod_{i=1}^{n} p(t_i | C_i)$$

given a sentence $w_1 \ldots w_n$ and contexts $C_1 \ldots C_n$

- The context includes previously assigned tags (for a fixed history)
- Beam search is used to find the most probable sequence in practice‘

Part of Speech (POS) Tagging

Mr.	Vinken	is	chairman	of	Elsevier	N.V.	,
NNP	**NNP**	**VBZ**	**NN**	**IN**	**NNP**	**NNP**	**,**

the	Dutch	publishing	group	.
DT	**NNP**	**VBG**	**NN**	**.**

- 45 POS tags
- 1 million words Penn Treebank WSJ text
- 97% state of the art accuracy

Chunk Tagging

Mr.	Vinken	is	chairman	of	Elsevier	N.V.	,
I-NP	**I-NP**	**I-VP**	**I-NP**	**I-PP**	**I-NP**	**I-NP**	**O**

the	Dutch	publishing	group	.
I-NP	**I-NP**	**I-NP**	**I-NP**	**O**

- 18 phrase tags
- B-XX separates adjacent phrases of same type
- 1 million words Penn Treebank WSJ text
- 94% state of the art accuracy

Named Entity Tagging

Mr.	Vinken	is	chairman	of	Elsevier	N.V.	,
I-PER	**I-PER**	**O**	**O**	**O**	**I-ORG**	**I-ORG**	**O**

the	Dutch	publishing	group	.
O	**O**	**O**	**O**	**O**

- 9 named entity tags
- B-XX separates adjacent phrases of same type
- 160,000 words Message Understanding Conference (MUC-7) data
- 92-94% state of the art accuracy

Contextual Predicates

Condition	Contextual predicate
freq$(w_i) < 5$	X is prefix/suffix of w_i, $\|X\| \leq 4$ w_i contains a digit w_i contains uppercase character w_i contains a hyphen
$\forall w_i$	$w_i = X$ $w_{i-1} = X$, $w_{i-2} = X$ $w_{i+1} = X$, $w_{i+2} = X$
$\forall w_i$	$\text{POS}_i = X$ $\text{POS}_{i-1} = X$, $\text{POS}_{i-2} = X$ $\text{POS}_{i+1} = X$, $\text{POS}_{i+2} = X$
$\forall w_i$	$\text{KLASS}_{i-1} = X$ $\text{KLASS}_{i-2}\text{KLASS}_{i-1} = XY$

Additional Contextual Predicates

Condition	Contextual predicate
freq$(w_i) < 5$	w_i contains period w_i contains punctuation w_i is only digits w_i is a number w_i is {upper,lower,title,mixed} case w_i is alphanumeric length of w_i w_i has only Roman numerals w_i is an initial (X.) w_i is an acronym (ABC, A.B.C.)

Additonal Contextual Predicates

Condition	Contextual predicate
$\forall w_i$	memory NE tag for w_i unigram tag of w_{i+1} unigram tag of w_{i+2}
$\forall w_i$	w_i in a gazetteer w_{i-1} in a gazetteer w_{i+1} in a gazetteer
$\forall w_i$	w_i not lowercase and $f_{\text{lc}} > f_{\text{uc}}$
$\forall w_i$	unigrams of word type bigrams of word types trigrams of word types

Example Word Types

- `Moody` $\Longrightarrow$ `Aa`
- `A.B.C.` $\Longrightarrow$ `A.A.A.`
- `1,345.00` $\Longrightarrow$ `0,0.0`
- `Mr. Smith` $\Longrightarrow$ `Aa. Aa`

Combinatory Categorial Grammar (CCG)

- CCG is a **lexicalised grammar formalism**

The	WSJ	is	a	publication	that	I	read
NP/N	N	$(S[dcl]\backslash NP)/NP$	NP/N	N	$(NP\backslash NP)/(S[dcl]/NP)$	NP	$(S[dcl]\backslash NP)/NP$

- grammatical information **encoded in the lexical categories**
- a small number of combinatory rules combine the categories
- designed for recovery of long-range dependencies
  e.g. relativisation, coordination

Supertagging

- **assigning one or more lexical categories to each word**
- **increases parser efficiency by reducing number of structures**
- parsing as assigning categories and then combining using rules
- introduced for Lexicalised Tree Adjoining Grammar (LTAG)
  Bangalore and Joshi (1999)
- previously each word was assigned *every* category it was seen with

Supertagging for CCG

- initially adapted to CCG to improve parsing efficiency
  Clark (2002)

- allows for rapid porting to new domains, e.g. questions
  Clark et al. (2004)

- **makes discriminative training feasible**
  $\Longrightarrow$ sophisticated log-linear statistical model

- **makes parsing extremely efficient**
  $\Longrightarrow$ fastest parser for a *linguistically-motivated* formalism

Supertagging for CCG

$$\frac{He}{NP} \quad \frac{goes}{(S[dcl]\backslash NP)/PP} \quad \frac{on}{PP/NP} \quad \frac{the}{NP/N} \quad \frac{road}{N} \quad \frac{with}{((S\backslash NP)\backslash(S\backslash NP))/NP} \quad \frac{his}{NP/N} \quad \frac{piano}{N}$$

$$\frac{A}{NP/N} \quad \frac{bitter}{N/N} \quad \frac{conflict}{N} \quad \frac{with}{(NP\backslash NP)/NP} \quad \frac{global}{N/N} \quad \frac{implications}{N}$$

- $\approx$ 400 lexical category types (from a complete set of $\approx$ 1,200)

- Baseline tagging accuracy is $\approx$ 72%

- **significantly harder than POS tagging**

CCG Unitagging

- assign **one** category per word
- train on sections 2-21 of CCGbank
- use GIS with a Gaussian prior for smoothing
  Curran and Clark (2003)
- **91.7% per-word accuracy on Section 23**
- **accuracy is not high enough for integration into a parser**
  Clark (2002)

CCG Multitagging

- assign **potentially more than one** category per word
- use $P(y_i|X)$ directly to assign categories to i-th word:
  assign any category with probability within β of the most probable category
- $P(y_i|X) \approx P(y_i|x_i)$ (ignoring history features)
- no beam required – extremely fast
- **a better solution is to use the forward-backward algorithm**
  but this simple solution works very well

Multitagging Accuracy

β	CATS/ WORD	GOLD POS WORD	GOLD POS SENT	AUTO POS WORD	AUTO POS SENT
0.1	1.4	97.0	62.6	96.4	57.4
0.075	1.5	97.4	65.9	96.8	60.6
0.05	1.7	97.8	70.2	97.3	64.4
0.01	2.9	98.5	78.4	98.2	74.2
$0.01_{k=100}$	3.5	98.9	83.6	98.6	78.9
0	21.9	99.1	84.8	99.0	83.0

The Parser

- takes POS tagged text as input
- uses a packed chart to represent **every** possible analysis consistent with supertags
- uses CKY chart parsing algorithm described in Steedman (2000)
- uses **conditional log-linear** parsing model
- uses Viterbi algorithm to find the most probable derivation

Log-Linear Parsing Models

- many parsing models evaluated in Clark and Curran (2004)
  - all-derivations model
  - **normal-form model**

- **recovers dependencies at around 85% F-score**

Log-Linear Parsing Models

- many parsing models evaluated in Clark and Curran (2004)
  - all-derivations model
  - **normal-form model**

- **recovers dependencies at around 84% F-score**

- all use a **discriminative** estimation method
  $\Longrightarrow$ **requires all of the derivation space**

- wide-coverage CCG charts are often huge (trillions of possible derivations)

Practical Estimation

- **40 000 sentences × up to several trillion parses each**
- packed chart representation is extremely compact
- still requires **over 31 GB of RAM** !
- use a 64-node Beowulf cluster and MPI programming

Training Data

- CCGbank data consists of one normal-form derivation
- supertagger assigns **additional plausible but incorrect categories**
- categories + CCG rules determines the search space
- parser **learns to select correct derivation** from this space
- minimise search space w/o loss of parser accuracy
  $\Longrightarrow$ **can reduce space with supertagging and constraints**

Constraints

normal-form only uses type-raising and composition when necessary

CCGbank constraints only allow seen category combinations
e.g. although NP/NP NP/NP can forward compose
doesn't appear in CCGbank Sections 2-21

Eisner normal-form constraints limits use of composed categories
very useful for restricting search space

Reducing the Space for Training

SUPERTAGGING/PARSING CONSTRAINTS	USAGE DISK	USAGE MEMORY
original $\beta = 0.01 \rightarrow 0.05 \rightarrow 0.1$	17 GB	31 GB
new constraints	9 GB	16 GB
new $\beta = 0.05 \rightarrow 0.1$	2 GB	4 GB

$\beta = 0.01$ is the **least restrictive** supertagger setting
packed charts limited to 300,000 nodes

Reducing the Space for Training

- constraints **reduce space by about 48%**
- constraints + tighter supertagging **reduce space by 87%**
- **gives state-of-the-art performance of 84.6 F-score**
- **now feasible to perform estimation on a single machine**

Running the Parser

old strategy give the parser maximum freedom to find best parse

- assign **as many categories as possible** initially
- reduce the number of categories if the chart gets too big

new strategy give the parser limited freedom to find the best parse

- assign **as few categories as possible** initially
- increase the number of categories if we don't get an analysis

$\Longrightarrow$ **parser decides if the categories provided are acceptable**

Parse Times for Section 23

SUPERTAGGING/PARSING CONSTRAINTS	TIME SEC	SENTS /SEC	WORDS /SEC
original $\beta = 0.01 \rightarrow \ldots \rightarrow 0.1$	3 523	0.7	16
new constraints	995	2.4	55
new $\beta = 0.1 \rightarrow \ldots 0.01_{k=100}$	608	3.9	90
new constraints	100	24.0	546
new beam	67	35.8	814
new beam and $\beta = 0.1 \rightarrow 0.075$	46	52.2	1 186
oracle	18	133.4	3 031

Parser is using the correct supertags
Coverage is 93%

I canna break the laws of physics . . .

- **speed increased by a factor of 77**
- F-score also **increased by 0.5%** using new strategy
- faster than other wide-coverage linguistically-motivated parsers by an **order of magnitude** (and approaching two) e.g. Collins (1998) and Kaplan et al. (2004)
- still room for **further speed gains** with better supertagging

Further Tagging Developments

Conditional Random Fields (a.k.a. Markov Random Fields)

- assign probability to entire sequence as a single classification
- uses cliques of pairs of tags and Forward-Backward algorithm
- overcome the *label bias* problem
- but in practice this doesn't seem to be a major difficulty

Work in Progress

- forward-backward multitagging
- real-valued features for tagging tasks
- question classification

Forward-Backward Multitagging

- how can we incorporate the history into multitagging?
- one solution: **sum over all sequences involving a given tag**
- i.e. all of the probability mass which use a tag
- use the forward-backward algorithm
- gives much lower ambiguity for the same level of accuracy

Real-valued features (David Vadas)

- **features can have any non-negative real-value**
  i.e. features are not *required* to be binary-valued
- can encode corpus derived information about unknown words

  e.g. *John ate the* **blag** *.*

- gives $\approx 1.4\%$ improvement on POS tagging unseen words

Question Classification (Krystle Kocik)

- questions can be classified by their answer type
  e.g. *What is the capital of Australia* → **`LOC:city`**
- 6 course grained and 50 fine grained categories
- state of the art is SNoW (Li and Roth, 1999) at 84.2% accuracy (fine grained)
- Maximum Entropy model gives accuracy 85.4% with CCG parser features

Future Work

- using multitags as features in cascaded tools
- i.e. keeping the ambiguity in the model for longer
- automatic discovery of useful complex features
- other smoothing functions (L_1 normalisation)

Conclusions

Maximum Entropy modelling is a
very powerful,
flexible and
theoretically well motivated
Machine Learning approach.

It has been applied successfully to many NLP tasks

Use it!

Acknowledgements

Stephen Clark is my co-conspirator in almost all of these MaxEnt experiments.

Thanks to **Krystle Kocik** and **David Vadas** for their great honours project work.

I would like to thank Julia Hockenmaier for her work in developing CCGbank and Mark Steedman for advice and guidance.

This work was supported by EPSRC grant GR/M96889 (Wide-Coverage CCG parsing) and a Commonwealth scholarship and a Sydney University Travelling scholarship.