Machine Learning for Language Processing
Lecture 3: Maximum Entropy Models

Stephen Clark

October 6, 2015

Discriminative Models With a discriminative model $P(w_j|\boldsymbol{x})$ there is no need to model the joint distribution $P(w_j, \boldsymbol{x})$. For many NLP tasks this makes sense, since the input $\boldsymbol{x}$ is given and constant across the set of possible outputs, so why model $\boldsymbol{x}$? Modelling the conditional distribution directly is also arguably closer to the classification process, which assumes a fixed input. Finally, there is no need to specify a generative process, with all its associated independence assumptions, which may not be correct.

NER as a Tagging Problem We'll use Named Entity Recognition (NER) as an example of a tagging problem, as well as POS tagging, since NER is a problem that benefits from rich, overlapping features. The main motivation for MaxEnt models is to enable the use of such feature sets. Here we're using a standard NER tagset: { LOC, PER, ORG, DAT, TIME } (location, person, organisation, date, time).[1] I-PER, for example, should be read as "inside person". Sequences of such tags pick out named entities, with B-PER being used if two separate person entities are contiguous.

Feature-based Models Features in a MaxEnt tagging model should be thought of as pairs: elements of the context paired with a particular tag. For example, if the word being POS tagged starts with a capital letter, then that's positive evidence for the NNP tag. The feature which encodes this evidence is $\langle$ title_caps, NNP $\rangle$; and the associated weight, which is learnt during the training process, encodes the strength of evidence.

Complex Features The key point about Maximum Entropy models is that the features can be arbitrarily complex, and they don't have to be statistically independent (unlike in the Naive Bayes model). For example, when doing NER tagging, it might be useful to know that the topic of the document is cricket, rather than, say, hill walking, in which case *Lancashire* might be tagged as an organisation (the cricket club) rather than location (the county). Incorporating

[1] It's possible to use much larger tagsets, depending on the application.

such information into an HMM tagger would be non-trivial; in a MaxEnt tagger it can easily be incorporated into the features.

Feature-based Tagging The downside of MaxEnt models, compared to the HMM, is that a more complex estimation method is required – although by current machine learning standards in NLP, the estimation is still relatively easy, since the objective function is convex (at least for MaxEnt models for supervised taggers).

Features in MaxEnt Models Mathematically a feature is a pair, or equivalently, a binary-valued indicator function which takes the value 1 when the pair is given as input to the function, and 0 otherwise.[2] The part of the feature definition which picks out the relevant part of the context is often called a *contextual predicate*.

The Model The form of the MaxEnt model is *log linear*, a general form of probabilistic model which is very popular in machine learning. Here we're using a conditional model: the probability of a tag, t, given a context, C. For the tagging problem, C is usually defined as a fixed-word window either side of the word being tagged, but in general C can contain any information deemed relevant to the tagging decision.

For a particular tag, context pair, we say that a feature "fires" or is "on" when it has the value 1. The total number of features n can be very large, even in the millions. However, only a small proportional of these features will fire for any particular tag, context pair.

Tagging with MaxEnt Models The model on the previous slide is a classification model. (In fact, Pang et. al used such a model for sentiment classification; see Lecture 1.) A simple way to use a classification model for sequence tagging is to chain the decisions together, multiplying the probabilities [2]. Crucially, the context for a particular decision will include the tags assigned to the previous word, much like the n-gram HMM tagger, and features will be defined in terms of those previous decisions. The search problem is very similar to that for an HMM tagger, and a variant of the Viterbi algorithm can be used to find the highest-scoring sequence of tags according to the probability of the tag sequence given the word sequence, $p(t_1 \dots t_n | w_1 \dots w_n)$.

There is a theoretically more pleasing approach to conditional sequence modelling, which is the Conditional Random Field (CRF) [1]. CRFs have the appealing property that the sequence probability is *globally* optimised, during training and testing. In contrast, the training for the MaxEnt model is *local*, in the sense that it effectively treats each tagging decision as a separate training instance (whilst still conditioning on the previous tags). There are a number of

[2] Entending MaxEnt models to handle count-based, or even real-valued, features is straightforward.

DP algorithms defined for CRFs, including the Viterbi algorithm for finding the highest-scoring tag sequence at test time.

Whether the theoretical niceties of the CRF lead to better performance over MaxEnt taggers is unclear. In practice, MaxEnt models can usually be made to perform as well as CRFs by using a large, rich feature set.

Model Estimation There are two ways of approaching the estimation problem for MaxEnt models. One is to simply assume the log-linear form given earlier, and use maximum likelihood estimation (i.e. write down the probability of the data assuming a log-linear model, and find the weight values which maximise that probability). Another, which derives the log-linear form, is to start with a natural set of constraints, and then choose the maximum entropy model from the set of models which satisfy those constraints.[3] Remarkably, both approaches lead to the same estimates for the weight values.

The Constraints A natural choice for the constraints is to say that the expected value of each feature according to the model should be equal to the empirical expected value. The empirical expected value is just the number of times the feature turns up in the data. The expected value according to the model is the prediction the model makes about how often it thinks the feature should turn up.

Choosing the MaxEnt Model These constraints do not pick out a single model, so we need a method of selecting from the set of models that do satisfy the constraints. A natural choice is the model with maximum entropy. The entropy H of a distribution p is a measure of how uniform the distribution is.[4] The expression given on the slide for $H(p)$ is the *conditional* entropy, since we're dealing with a set of conditional probability distributions.

The Maximum Entropy Model One intuitive motivation for selecting the maximum entropy model is that it's the most uniform model which satistfies the constraints provided by the data, so in some sense, when selecting it, we are making no assumptions outside of what is observed in the data.

One simple method for estimating the weights of a MaxEnt model is Generalised Iterative Scaling (GIS).

Generalised Iterative Scaling (GIS) We're much more sophisticated in NLP now regarding numerical optimisation than we were in the late 90s, when MaxEnt models were starting to become popular. Hence a standard method now for estimating the weights would be to just optimise for (log-)likelihood, using any standard gradient-based optimisation technique. The gradients are easy to calculate — in fact they're just the differences between the expected

[3] By model we just mean the set of conditional probability distributions we're trying to estimate.

[4] $H(p) = -\sum_x p(x) \log p(x)$

and empirical feature values — and, since the likelihood is convex, global optimisation is easy.

However, it is worth considering GIS since it has a clear intuition, is easy to implement, and, for the tagging problem at least, training is relatively quick. (Training on the Penn Treebank for the C&C POS tagger takes roughly 10 minutes for 100 iterations.)

POS Tagger Features The features on the slide, which are a fairly standard set for a POS tagger, look at the words in a 5-word window centred on the word being tagged. These also include the previous two tags, assuming a left-to-right tagging process.

POS Tagger Features for Rare Words The more interesting features, and the ones which display the flexibility of MaxEnt models, are those that only apply to the rare words (here defined as words that occur less then 5 times in the training data). The idea is that, if a word occurs enough times in the training data, we have good evidence for its tag by simply looking at the word itself and the words (and tags) in the local window. However, for rare words it's useful to look at the internals of the word itself, for example its prefixes and suffixes, whether it contains a digit, uppercase character or hyphen.

The range of features that can be defined in this way is only restricted by the imagination of the model developer. Whether such features help with accuracy is an empirical question, of course, and needs to be tested using the training and (development) test data.

Performance MaxEnt taggers still offer competitive performance, although taggers based on RNNs may turn out to be superior, especially if you are interested in maintaining accuracy across domains (since RNNs tend to generalise better).

Contextual Predicates for NER The final set of slides are designed to show the range of features that can be added to a tagger, in this case the C&C NER tagger.

Readings for Today's Lecture

- Chapter 6 (Hidden Markov and Maximum Entropy Models) of Jurafsky and Martin (2nd. Ed.)

References

[1] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, pages 282–289, Williams College, MA, 2001.

[2] Adwait Ratnaparkhi. A maximum entropy part-of-speech tagger. In *Proceedings of the EMNLP Conference*, pages 133–142, Philadelphia, PA, 1996.