A DISCRIMINATIVE APPROACH TO PHRASE BREAK MODELLING

Stephen Cox

School of Computing Sciences, University of East Anglia, Norwich NR4 7TJ, U.K.
sjc@cmp.uea.ac.uk

Abstract

We address the problem of predicting pauses between the words in a sentence, which is of considerable interest for text to speech systems. In doing so, we show that the performance of both a generative classifier (naive Bayes, NB) and a discriminative classifier (maximum entropy, ME) can be significantly enhanced by application of the generalised probabilistic descent (GPD) algorithm. The features used for prediction of pauses in sentences are both local (derived from the neighbourhood of a word juncture) and global (derived from a parse tree of the sentence). We first compare the results of using the NB and ME classifiers on these features, and then develop the theory required for applying GPD to these classifiers. We show that GPD is particularly suitable for application within the maximum entropy framework and increases very significantly the discriminative power of both the NB and ME classifiers. The F-score of 81.2% obtained after application of GPD to an ME classifier is believed to be the best performance obtained on the Boston Radio Corpus.

1. Introduction

Our goal in this work is to predict the location of pauses (breaks) within an utterance to be spoken by a text-to-speech (TTS) system. For present purposes, we have focused on classifying a juncture between two words in a sentence as being either a break or a non-break—a good solution to this problem is important for TTS systems. Although it is clear that some sentences require semantic and pragmatic analysis to determine pause placement, previous work (e.g. [2]) has shown that good accuracy is possible using a pattern classification approach. The arguments for and against using discriminative rather than generative classifiers are well-known. The principal attraction of using the former is that there is rarely enough data to estimate an accurate generative model, and in this situation, it has been shown that discriminative classifiers will outperform generative classifiers [11]. Generalised probabilistic descent (GPD) is a discriminative algorithm that has been extensively studied and used, especially in the field of speech recognition and related areas [7]. GPD is an adaptive, gradient-based procedure that aims to minimize the classification error on the training-set. It provides a general framework for iteratively adjusting the parameters of a model in such a way that performance on the training-data increases. GPD has been applied to applications such as speaker recognition [10], call-routing [8] and phoneme recognition [5]. The performance obtainable from GPD depends on the underlying model used for classification. In many practical tasks, there is little choice available for the underlying model used, but in cases where choice is available, the interaction of the model with the GPD procedure is an interesting question. In this paper, we examine the effect of adapting the parameters of a generative classifier (Naive Bayes) and a discriminative classifier (maximum entropy) using GPD.

It seems natural to adapt the parameters of a generative model, such as NB, using a discriminative approach (see, for instance, [9], where logistic regression was used to adapt the parameters of an NB classifier). However, it may seem surprising to adapt the parameters of an ME model, since ME is already a discriminative model, as it models the class posteriors directly. However, different criteria can be used to achieve discriminative modelling. The ME model is based on the premise that the best assumption for the probability distribution of the data is the distribution that maximises the entropy subject to the constraints specified in the data, and features in ME are estimated under these twin constraints. This may be viewed as a "softer" approach to discrimination than GPD, which estimates features that directly minimise the error on the training-set data (an approach known as minimum classification error, MCE [7]). Our contention is that the features estimated using the "soft" discriminative approach of ME may be a better starting-point for the application of GPD than features derived from a generative model.

2. Maximum entropy framework and features

In common with other applications of maximum entropy (ME) techniques for language processing, we define a *context* to be a set of attributes derived from words around a juncture—in our case, the context extends from two words before the juncture to the word after the juncture. We use "attribute" rather than "feature" to describe the information derived from the neighbourhood of the juncture to avoid confusion with the definition of an ME feature given in equation 1. A *contextual predicate (cp)* has a value of either *true* or *false* for a given context, and encapsulates some information that is believed to be useful for the classification task. For instance, suppose the set of attributes for a given juncture are part-of-speech (PoS) tags of the two words before the juncture and the PoS tag for the word after the juncture, and that for this juncture, the set is {DT, JJ, NN}. Then for this context, the *cp* "PoS_1='DT' AND PoS_2='JJ"' has the value *true*, whereas the *cp* "PoS_3='NNP"' has the value *false*. The contextual predicates are used for classification by defining *features*. Let the set of classes be denoted by $y = \{y_1, y_2, \ldots, y_{Ncl}\}$ and the set of observed contexts in the training data be $x = \{x_1, x_2, \ldots, x_{Ncon}\}$. Then a feature is defined as

$$f_{cp,y'}(x,y) = \begin{cases} 1 & \text{if } y = y' \text{ and } cp(x) = \textit{true} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

(Note that this use of "feature" differs from the use usually found in pattern recognition, where features are independent of class.) Under the maximum entropy framework, the conditional probability of a class y given an observed context x is modelled

as

$$\Pr(y|x) = \frac{1}{Z(x)} \exp\left(\sum_{j=1}^{N_f} \lambda_j f_j(x,y)\right), \quad (2)$$

where λ_j is a weight applied to each feature and N_f is the number of features. The term

$$Z(x) = \sum_y \exp\left(\sum_{j=1}^{N_f} \lambda_j f_j(x,y)\right), \quad (3)$$

is a normalising term to ensure that $\sum_y \Pr(y|x) = 1$. Training consists of estimating values of λ_j that are both consistent with the observed training-data and that maximise the conditional entropy $H(p) = \sum_{x,y} \Pr(x)\Pr(y|x)\log\Pr(y|x)$. There are several ways in which this can be done, and in these experiments, we used the Generalised Iterative Scaling (GIS) algorithm—see [1] for details. Classification of an unlabelled juncture is done by assigning it to class y^* where (in this two-class case) $y^* = argmax\{\Pr(y_1|x), \Pr(y_2|x)\}$. For a more comprehensive introduction to the ideas of maximum entropy in language processing, see [1, 13]

3. Naive Bayes framework

The naive Bayes (NB) classifier uses the same attributes as are used in the ME classifier. The probabilities $\Pr(\text{context} = x_i|\text{class} = y_c)$ are estimated as products of the conditional probabilities of the attributes. A typical conditional probability is estimated using Laplace smoothing as

$$\Pr(\text{attribute} = a_j|\text{class} = y_c) = \frac{\#(a_j, y_c) + 1}{\#(y_c) + 2}, \quad (4)$$

where $\#(a_j, y_c)$ is the number of joint occurrences of attribute a_j with class y_c and $\#(y_c)$ is the number of occurrences of class y. Hence the probability of class y_c given a set of attributes within a context c is

$$\Pr(y_c|x) = \Pr(y_c)\prod_{j=1}^{N_j} \Pr(a = a_j|y = y_c) \qquad c \in 1,2 \quad (5)$$

For classification, juncture i of unknown classification is assigned to class c^* where $y^* = argmax\{\Pr(y_1|x), \Pr(y_2|x)\}$.

4. Generalised Probabilistic Descent Applied to ME and NB

We assume that the GIS algorithm has been applied to the training-data to produce a set of N_f features and their corresponding weights $\lambda_1, \lambda_2, \ldots, \lambda_{N_f}$. Suppose that the i'th juncture in the training-data has class $y_{c(i)}, c(i) \in \{1,2\}$. Let r_i be the ratio

$$r_i = \frac{\Pr(y_{\bar{c}(i)}|x)}{\Pr(y_{c(i)}|x)} \quad (6)$$

where $y_{\bar{c}(i)}$ is the incorrect class. Let

$$l_i = \log(r_i) = \log(\Pr(y_{\bar{c}(i)}|x)) - \log(\Pr(y_{c(i)}|x)). \quad (7)$$

Then $l_i < 0$ if the juncture is correctly classified and $l_i > 0$ if the juncture is mis-classified. We might seek to estimate values of the λ_i that minimise l_i, but in practice, this tends only to further decrease l_i's that are already negative and leads to little improvement in classification. Instead, we use the logistic

$$z_i = \frac{1}{1 + e^{-\gamma l_i}} \qquad \gamma > 0 \quad (8)$$

to transform the l_i's, which also has the important property of being differentiable. In equation 8, $z_i \to 1$ for positive l_i, $z_i \to 0$ for negative l_i, and the higher the value of γ, the sharper the transition around 0.5.

The weights are adjusted iteratively according to

$$\lambda_i^{t+1} = \lambda_i^t + \delta\lambda_i \quad (9)$$

where

$$\delta\lambda_j = -\epsilon\frac{\partial z_i}{\partial \lambda_j} \qquad \epsilon > 0 \quad (10)$$

i.e. the weights are moved in the direction of the negative gradient according to an empirically chosen learning rate ϵ. Now

$$\frac{\partial z_i}{\partial \lambda_j} = \frac{\partial z_i}{\partial l_i}\frac{\partial l_i}{\partial \lambda_j}, \quad (11)$$

and it can be shown that

$$\frac{\partial z_i}{\partial l_i} = \gamma z_i(1 - z_i). \quad (12)$$

Using equations 2 and 7, we see that

$$l_i = \sum_{j, y \neq y_{c(i)}} \lambda_j f_j(x,y) - \sum_{j, y = y_{c(i)}} \lambda_j f_j(x,y) \quad (13)$$

Hence

$$\frac{\partial l_i}{\partial \lambda_j} = \begin{cases} f_j(x,y) & \text{if } y \neq y_{c(i)} \\ -f_j(x,y) & \text{if } y = y_{c(i)}. \end{cases} \quad (14)$$

Since $f_j(x,y) = 1$ if a feature is "active", the update equation is

$$\frac{\partial z_i}{\partial \lambda_j} = \begin{cases} \gamma z_i(1 - z_i) & \text{if } y \neq y_{c(i)} \\ -\gamma z_i(1 - z_i) & \text{if } y = y_{c(i)}. \end{cases} \quad (15)$$

We see from equations 15 and 10 that equivocal classifications, for which $l_i \approx 0$, give the maximum adjustment to the λ_j's in the appropriate direction. Note also that because of the exponential form of the dependence of the conditional probabilities on the weights (equation 2), the derivative of equation 14 is particularly simple to derive and compute.

To adapt the NB weights using GPD, we again form the log of the likelihood ratio of the incorrect class to the correct class for the i'th juncture, equation 7. For the NB classifier,

$$\Pr(y_c|x)) = \Pr(y_c)\prod_{j=1}^{N_j} \Pr(a = a_j|y = y_c) \qquad c \in 1,2 \quad (16)$$

so that

$$\begin{aligned} l_i \quad = \quad & \log(\Pr(\bar{c}(i))) - \log(\Pr(c(i))) + \\ & \sum_{j=1}^{N_j} \log(\Pr(a = a_j|y = y_{\bar{c}(i)})) - \\ & \sum_{j=1}^{N_j} \log(\Pr(a = a_j|y = y_{c(i)})). \end{aligned} \quad (17)$$

Denoting $\Pr(a = a_j|y = y_{c(i)}))$ as P_j,

$$\frac{\partial l_i}{\partial P_j} = \begin{cases} 1/P_j & \text{if } y \neq y_{c(i)} \\ -1/P_j & \text{if } y = y_{c(i)}. \end{cases} \quad (18)$$

which leads to the update equations

$$\frac{\partial z_i}{\partial P_j} = \begin{cases} \frac{\gamma z_i(1-z_i)}{P_j} & \text{if } y \neq y_{c(i)} \\ -\frac{\gamma z_i(1-z_i)}{P_j} & \text{if } y = y_{c(i)}. \end{cases} \quad (19)$$

5. Experiments and Results

The Boston Radio News Corpus [12] annotated to the full ToBI specification [15] was used for these experiments. This was divided into a training set of 13,754 words (3,437 breaks) and a testing-set of 15,333 words (3,894 breaks). Since the intention was to classify breaks/non-breaks, word junctures labelled as level 3 or above were considered to be breaks, and junctures with a lower level of labelling were considered to be non-breaks.

The principal attributes used for determination of the class of a juncture are the part-of-speech (PoS) tags of the words around the juncture. Most PoS taggers use over 40 tags, many of which are likely to introduce noise rather than being useful for the task of prosodic phrase break prediction. In [14], we describe a technique that both reduces the number of tags (typically to about 7–8) and increases the accuracy of break prediction. This reduced tag-set was used throughout these experiments. The other attributes are derived from a parse of the sentence using the Collins parser [4]. The attributes used to characterise a juncture are as follows:

1. A_1: Parse depth (positive integer)
2. A_2: Size of the largest phrase ending with the current symbol (positive integer)
3. A_3: Whether the juncture is in a major phrase or is not in a major phrase (Boolean)
4. A_4: PoS tag of word 3 words before juncture (positive integer)
5. A_5: PoS tag of word 2 words before juncture (positive integer)
6. A_6: PoS tag of word before juncture (positive integer)
7. A_7: PoS tag of word after juncture (positive integer)

If each attribute is considered independently and is either used or not used in the classification process, there are $2^7 = 128$ possible attribute combinations. However, attributes can also be combined to in the formation of a feature: for instance, if attributes A_1 and A_2 are used, we can form contextual predicates of the form: cp_1: "$A_1 = x$"; cp_2: "$A_2 = y$"; cp_3: "$A_1 = x$ AND $A_2 = y$" (where x and y are positive integers). All possible combinations of attributes were used in these experiments.

In forming the ME weights, the GIS algorithm was iterated until the difference between the ME model of the conditional probabilities and the conditional probabilities as estimated from the data was below some threshold (typically 20–30 iterations). For discriminative training, a range of values of ϵ and γ were examined: best results were obtained with $\epsilon = 0.1$ and $\gamma = 8$. Training was iterated until no improvement in classification performance was observed, typically 5–10 iterations.

We use F-score [3] to measure performance. The F-score effectively balances insertion errors (non-breaks identified as breaks) with deletion errors (breaks identified as non-breaks). It is calculated as $F = 2PR/(P + R)$, where $P = precision$ and $R = recall$ are defined as

$$P = \frac{\text{\# breaks correct}}{\text{\# breaks predicted}} \quad R = \frac{\text{\# breaks correct}}{\text{\# breaks in testing-set}}. \quad (20)$$

As a baseline, we measured performance using punctuation to demarcate breaks. Any occurrence within a sentence of a comma, colon, semicolon, question mark, exclamation mark, bracket (open or closed) quotation mark (open or closed) was marked as a pause. The result was $P = 95.83\%$, $R = 35.41\%$, $F = 51.72\%$, reflecting the fact that nearly all punctuation does demarcate a pause, but not all pauses are demarcated by punctuation. much Table 1 compares the performance using (a) NB probabilities, (b) ME weights, (c) NB probabilities after GPD adaptation and (d) ME weights after GPD adaptation, using ten different sets of attributes that were selected because they gave good performance using either the ME weights, or the NB probabilities, or both. Figures in bold indicate the best performance for a particular technique. Comments on the results:

Attributes used	*F-score* NB probs	ME weights	NB probs. +GPD	ME weights +GPD
6	**45.73**	62.68	54.07	62.68
6,7	36.31	**70.70**	56.83	78.91
3,6,7	37.86	67.46	71.91	79.36
3,5,6,7	26.24	69.43	69.31	80.17
3,4,6,7	18.20	68.36	**74.67**	79.29
2,6	24.51	68.68	64.00	80.20
2,6,7	15.73	69.31	67.60	80.48
2,3,6	21.72	68.24	65.45	80.04
2,3,6,7	15.63	68.78	55.70	**81.20**
1,3,6,7	24.99	68.76	68.48	80.10
MEAN	26.69	68.24	64.80	78.24

Table 1: Performance of NB probabilities and ME weights on different attribute combinations before and after application of GPD

1. No untransformed NB result is above the baseline given by using punctuation ($F = 51.72\%$), whereas all other results are above it.
2. Using untransformed naive Bayes probabilities gives much worse performance in all cases than using untransformed ME weights. In general, as the number of attributes used increases, the NB performance decreases, indicating that the NB assumption of independence of features is not justified. However, the ME results are much more consistent over the attribute combinations, suggesting that ME learns appropriate weightings for the attributes used.
3. For all attribute sets, performance after applying GPD to both the NB probabilities and the ME weights is considerably higher than on the untransformed probabilities or weights. Using "juncture error" as a metric rather than F-score, if any result chosen from column two of Table 1 is compared with any result chosen from column four, application of McNemar's Test [6] shows that the difference in performance is statistically hugely significant. The same applies to any pair of results chosen from columns three and five respectively, except for the first entry in these columns.
4. Best performance was obtained after applying GPD to the ME model that used attributes 2,3,6 and 7. Using these features, 4.93% of non-breaks were mis-classified and 23.33% of breaks.

It was instructive to examine some of the junctures on which the algorithm made errors. Shown below are six examples of errors made by the algorithm: the symbol ∇ denotes an actual break that was mis-recognised as a non-break and $\triangle$ a non-break that was mis-recognised as a break.

1. It may be the most important appointment ∇ Governor Michael

Dukakis makes during the remainder △ of his administration and one of toughest.

2. As WBUR's ▽ Margo Melnicove reports.....

3. Democratic Governor Michael Dukakis fulfilled a campaign promise to de-politicize ▽ judicial appointments.

4. That year ▽ Thomas Maffy, now president of the Massachusetts Bar Association, was...

5. Hennessy is the S.J.C.'s ▽ thirty-second chief justice.

6. Kassler says, unlike the Federal ▽ Supreme Court, there's no litmus test on particular issues that Massachusetts high court nominees must pass.

The above suggests that it is surprising that many of the breaks that were mis-recognized as non-breaks (B→NB, about 2/3 of the errors) are actually breaks: for instance, the B→NB misrecognitions in examples 1, 2, 3 and 5. They reflect the nature of the data, which was news reports broadcast over the radio. Because their scripts are "semantically dense", newsreaders tend to read slowly with pauses before or after semantically important words to emphasise them. Although the classifier was trained on this material, it may be that the resulting phrases have less correlation with parse constituents than naturally spoken phrases and so they are harder to spot. However, the B→NB errors in examples 4 and 6 are examples of breaks that require deeper processing. In example 4, there is a break before "Thomas Naffy" because the phrase "that year" functions in the same way as "yesterday" or "today", but the parser failed to separate it from the following name, and no examples of "that year" occur in the training data. Example 6 shows how semantic and long-range considerations are used by humans when planning the sentence prosody. The break after "Federal" is to emphasise the word in order to contrast it with "Massachusetts", as the sentence is comparing the two court systems. This clearly requires an understanding of the long-range sentence structure and the semantics of contrasting concepts that is beyond what is implemented in this classifier. Analysis of the non-breaks classified as breaks (NB→B, about 1/3 of the errors) reveals various kinds of error, the most common being a tendency to over-segment longer constituents (e.g. "And he says △ there's ten million dollars from bond sales") and to be unaware of collocations (e.g. "drink △ and drive", "cider △ and beer"). But in general, the NB→B errors mostly appear to be capable of being fixed using syntax, in contrast to many of the B→NB errors, which require semantic processing.

6. Discussion

In this paper, we have applied the generalised probabilistic descent (GPD) algorithm to adapt the parameters of both a naive Bayes (NB) and a maximum entropy (ME) model on the task of classifying the junctures between words in an utterance to be spoken by a text-to-speech (TTS) system as either breaks or non-breaks. In both cases, application of GPD gave a substantial increase in performance. The best result ($F = 81.2\%$, juncture-error = 10.24%), obtained using GPD applied to ME weights, is considerably higher than the best result reported in [14] of 13.10% juncture error using a reduced PoS tag-set and trigrams, which at that time claimed to be state-of-the-art. We hope that this work will motivate more work in understanding the relationship between ME and GPD discrimination and in applying the algorithm to language processing tasks.

Acknowledgment

Thanks to Ian Read for generating the features used in this work.

7. References

[1] A.L. Berger, S.A. Della Pietra, and V.J. Della Pietra. A maximum entropy approach to language processing. *Computational Linguistics*, 22(1):39–71, 1996.

[2] A.W. Black and P. Taylor. Assigning phrase breaks from part-of-speech sequences. *Computer Speech and Language*, 12:99–117, 1998.

[3] B. Busser, W. Daelemans, and A. van den Bosch. Predicting phrase breaks with memory-based learning. In *Proc. of the Fourth ISCA Tutorial and Research Workshop on Speech Synthesis*, 2001.

[4] M. Collins. *Head-Driven Statistical Models for Natural Language Parsing*. PhD thesis, University of Pennsylvania, 1999.

[5] S.J. Cox. Using context to correct phone recognition errors. In *Proc. Int. Conf. on Spoken Language Processing*, October 2004.

[6] L Gillick and S.J. Cox. Some statistical issues in the comparison of speech recognition algorithms. In *Proc. IEEE Conf. on Acoustics, Speech and Signal-processing*, pages 532–535, April 1989.

[7] S. Katagiri, B.H. Juang, and C.H. Lee. Pattern recognition using a family of design algorithms based upon the generalized probabilistic descent method. *Proceedings of the IEEE*, 86(11):2345–2373, November 1998.

[8] H.K.J. Kuo and C Lee. Discriminative training in natural language call-routing. In *Proceedings of the International Conference on Spoken Language Processing*, Beijing, October 2000.

[9] David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. In W. Bruce Croft and Cornelis J. van Rijsbergen, editors, *Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval*, pages 3–12, Dublin, IE, 1994. Springer Verlag, Heidelberg, DE.

[10] C.S. Liu et al. A study on minimum error discriminative training for speaker recognition. *Journal of the Acoustical Society of America*, 97:637–648, January 1995.

[11] A. Ng and M. Jordan. On discriminative vs. generative classifers: A comparison of logistic regression and naive Bayes. In *Advances in Neural Information Processing Systems*. The MIT Press, 2002.

[12] M. Ostendorf, P.J. Price, and S. Shattuck-Hufnagel. The Boston University radio news corpus. Technical Report ECS-95-001, Boston University, 1995.

[13] A. Ratnaparkhi. *Maximum Entropy Models For Natural Language Ambiguity Resolution*. PhD thesis, University of Pennsylvania, 1998.

[14] I.H. Read and S.J. Cox. Using part-of-speech for predicting phrase breaks. In *Proc. Int. Conf. on Spoken Language Processing*, October 2004.

[15] K. Silverman et al. TOBI: a standard for labelling english prosody. In *Proc. Int. Conf. on Spoken Language Processing*, pages 867–870, Banff, Canada, September 1992.