A Hierarchical Probabilistic Model for Novelty Detection in Text

L. Douglas Baker
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
ldbapp@cs.cmu.edu

Thomas Hofmann
ICSI, UC Berkeley
1947 Center St., Ste. 600
Berkeley, CA 94704
hofmann@icsi.berkeley.edu

Andrew K. McCallum
Just Research
4616 Henry St.
Pittsburgh, PA 15213
mccallum@justresearch.com

Yiming Yang
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
yiming@cs.cmu.edu

Abstract

Topic Detection and Tracking (TDT) is a variant of classification in which the set of classes grows over time. This paper describes a new approach to TDT based on probabilistic, generative models. Strong statistical techniques address the many challenges: hierarchical shrinkage for sparse data, statistical "garbage collection" for new event detection, clustering in time to separate the different events of a common topic, and deterministic annealing to create the hierarchy. Preliminary experimental results show promise.

1 Introduction

Topic Detection and Tracking (TDT), or novelty detection, is a variant of the traditional classification problem to allow for the identification of new classes. For example, news articles often report on new events that are different from any that have been seen before. Situations such as this require that a classifier be able to expand its set of classes. This new and interesting problem presents strong challenges for a learning algorithm: identifying and learning about new classes from very few documents, identifying and effectively using the temporal relationships in the data, and describing the new classes to a user.

When a new class is identified there will be only a few documents from which to learn about the new class. Hierarchical shrinkage [5] is used to improve the parameter estimates of these sparse classes by "borrowing" data from their siblings in the hierarchy. The hierarchy can be given in advance, or it can be learned using deterministic annealing [3, 9].

In many domains the temporal relationships among the documents are a useful feature. For example, news articles about the same event are usually published at

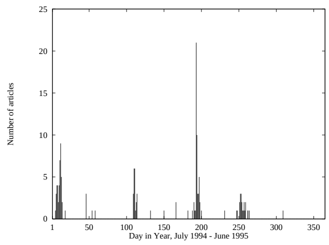

Figure 1: Histogram of the number of articles per day from a set of 150 documents that discuss floods. The documents discussing each of the four separate floods in these data are distinctly clustered in time.

about the same time, and different events related to the same topic generally occur at different times and places. For example, 'floods' may be a topic, but a specific flood at a specific time is an event (Figure 1)[1]. Clustering in time is used to separate the different events of a common topic. These events are added as the final layer of leaves in the hierarchy.

From these events 'vertical word movement' can then identify and separate topic-specific from event-specific words. The more general words are associated with nodes closer to the root of the hierarchy, and the more specific words with nodes closer to the leaves.

Deterministic annealing and EM are used to cluster documents by topic and to create a hierarchy of topics. The separation of topic- and event-specific words helps improve the quality of the hierarchy by letting the algorithm ignore the event-specific words when it is creating the topic hierarchy.

Statistical "garbage collection" is used for new event detection. New nodes are added to the hierarchy that capture those documents that do not belong to any of the existing events but may still be related to existing topics. New events can be described in relation to their position in the hierarchy.

2 Probabilistic Framework

We start with a multinomial naïve Bayes classifier and then show how this classifier can be used for clustering unlabelled documents (Section 2.1), incorporating a hierarchy of classes and separating general from specific words (Section 2.2), and finally learning the hierarchy itself (Section 2.3). Each of these changes results in a straightforward change to the way the parameters of the model are estimated.

The data are a set of documents $d_i \in \mathcal{D}$, each labelled as belonging to one of a set of classes $c_j \in \mathcal{C}$. Documents contain words $w_t \in \mathcal{V}$ from a vocabulary. We assume the data is generated by a parametric mixture model with one mixture component per class. We use a multinomial naïve Bayes model, parameterized by θ.

It is straightforward to use Bayes' rule to derive the probability of a class given a document (cf. for example [4]). Classifying a new document d is done by selecting the most likely class given the document according to

$$\mathrm{P}(c_j|d;\theta) = \frac{\mathrm{P}(c_j)\prod_{t=1}^{|\mathcal{V}|}\mathrm{P}(w_t|c_j)^{N(w_t,d)}}{\sum_{k=1}^{|\mathcal{C}|}\mathrm{P}(c_k)\prod_{t=1}^{|\mathcal{V}|}\mathrm{P}(w_t|c_k)^{N(w_t,d)}}. \tag{1}$$

[1]Refer to Section 4 for details on how these results were obtained.

$N(w,d)$ is the number of times word w occurs in document d.

The parameters of this model are the class priors $\hat{\theta}_{j0} = \mathrm{P}(c_j)$ and the class conditional word probabilities $\hat{\theta}_{jt} = \mathrm{P}(w_t|c_j)$. $\hat{\theta}$ indicates an estimate of θ. In what follows all probabilities are conditioned on $\hat{\theta}$, so we omit $\hat{\theta}$ from our notation. These parameters are estimated by counting the occurrences of documents and words in the training data:

$$\hat{\theta}_{j0} \quad = \quad \frac{\sum_{i=1}^{|\mathcal{D}|} \mathrm{P}(c_j|d_i)}{|\mathcal{D}|}, \quad \hat{\theta}_{jt} = \frac{\sum_{i=1}^{|\mathcal{D}|} N(w_t,d_i)\mathrm{P}(c_j|d_i)}{\sum_{s=1}^{|\mathcal{V}|}\sum_{i=1}^{|\mathcal{D}|} N(w_s,d_i)\mathrm{P}(c_j|d_i)}. \tag{2}$$

Here $\mathrm{P}(c_j|d_i) \in \{0,1\}$ according to the given document labels.

2.1 Clustering Unlabelled Documents

If our training data are a set of unlabelled instead of labelled documents, as they are in the novelty detection task, then we use our $\mathrm{P}(c_j|d_i)$ as probabilistic estimates of the labels for each document. However, since $\mathrm{P}(c_j|d_i)$ depends on $\hat{\theta}$ (via (1)), and $\hat{\theta}$ depends on $\mathrm{P}(c_j|d_i)$ (via (2)), we use an EM (Expectation Maximization) algorithm to find the final estimates of these parameters. First we choose some arbitrary initial values for $\hat{\theta}$. E-Step: We use $\hat{\theta}$ to estimate $\mathrm{P}(c_j|d_i)$. M-Step: then we use the new $\mathrm{P}(c_j|d_i)$ to re-estimate $\hat{\theta}$. These two E and M steps are iterated. Dempster, *et al.* [2] have shown that EM will converge to parameter values that locally maximize the likelihood of the model parameters for the given data.

2.2 A Hierarchical Model

In a novelty detection task, the parameter estimates for newly identified classes will be unreliable due to a lack of data. If the classes are agglomerated into hierarchy (with each node formed recursively by pooling the data from its children), we can use shrinkage to improve these estimates [5]. Shrinkage linearly interpolates the parameter estimates of each leaf in the hierarchy with the parameter estimates of the leaf's ancestors.

We first *locally* estimate the parameters for each node in the hierarchy using (2). Let $\{\hat{\theta}_j^1, \hat{\theta}_j^2, \ldots, \hat{\theta}_j^K\}$ be K such estimates for the nodes n_j^k along the path from a leaf n_j^1 to the root n_j^{K-1} (and beyond it, to the uniform estimate at the fictitious node n_j^K). $\check{\theta}$ (instead of $\hat{\theta}$) gives the new estimate of the model parameters:

$$\check{\theta}_{jt} = \mathrm{P}(w_t|c_j;\check{\theta}) = \lambda_j^1\hat{\theta}_{jt}^1 + \lambda_j^2\hat{\theta}_{jt}^2 + \cdots + \lambda_j^k\hat{\theta}_{jt}^k. \tag{3}$$

The EM algorithm used to estimate $\hat{\theta}$ is augmented to also estimate empirically optimal mixing weights λ_j^i for each leaf with its ancestors. Define $\mathrm{P}(n_j^k|w_t,c_j)$ to be the probability that n_j^k was used to generate a word w_t in class c_j. We start EM with an arbitrary setting of the λ's. E-Step: We calculate the degree, β_j^k, to which each estimate predicts the words w_t in the training data from class c_j. M-Step: then, we derive new (and guaranteed improved) weights λ_j^k by normalizing the β's.

$$\mathrm{P}(n_j^k|w_t,c_j) = \frac{\lambda_j^k\hat{\theta}_{jt}^k}{\sum_m \lambda_j^m\hat{\theta}_{jt}^m}, \quad \beta_j^k \quad = \sum_{w_t\in\mathcal{H}_j} \mathrm{P}(n_j^k|w_t,c_j), \quad \lambda_j^k = \frac{\beta_j^k}{\sum_m \beta_j^m} \tag{4}$$

Here $\mathcal{H}$ denotes a held-out set of data that is not used otherwise during training or testing. This is necessary to avoid overfitting and to avoid setting $\lambda_j^1 = 1.0$ for all leaves. In practice, to make efficient use of our training data, we use all the data in a leave-one-out fashion to estimate the mixing parameters.

We use 'vertical word movement' to separate specific and general words into different levels of the hierarchy, with more general words associated with nodes higher in the hierarchy. To do this we use a further extension of our EM algorithm to estimate in the M-step the multinomial probabilities $\hat{\theta}_{jt}^{k}$ based on $\mathrm{P}(n_j^k|w_t, c_j)$ from (4):

$$\hat{\theta}_{jt}^{k} = \frac{\sum_{c_l \downarrow n_j^k} P(n_j^k|w_t, c_l) \sum_{i=1}^{|\mathcal{D}|} \mathrm{P}(c_l|d_i) N(w_t, d_i)}{\sum_{s=1}^{|\mathcal{V}|} \sum_{c_l \downarrow n_j^k} P(n_j^k|w_s, c_l) \sum_{i=1}^{|\mathcal{D}|} \mathrm{P}(c_l|d_i) N(w_s, d_i)}. \tag{5}$$

$c_j \downarrow n_j^k$ denotes all class nodes (leaves) in the subtree with root node n_j^k.

2.3 Deterministic Annealing

We have shown how we can take advantage of a topic hierarchy when it is given. Deterministic Annealing (DA) is an approach to clustering based on probabilistic and information-theoretic principles that can be used to build a hierarchy and also helps avoid many poor locally optimal solutions that can trap the EM algorithm [9].

Usually, when using $\mathrm{P}(c_j|d_i)$ for classification, we simply choose the class with the highest probability. Alternatively, we could view classification as a random process where the class is chosen randomly according to $\mathrm{P}(c_j|d_i)$. In this view, it would be best for the probability to be 1 for the correct class and 0 otherwise. The key idea in DA is to view cluster formation as this kind of random process and then optimize the parameters while constraining the entropy of the distributions $\mathrm{P}(c_j|d_i)$ to prevent hard 0 and 1 probabilities. Intuitively, by constraining the entropy, we are implicitly smoothing the surface on which EM is hill-climbing, thus reducing the number of local optima in which EM might get stuck.

See [9] for a detailed derivation of how this entropy constraint affects parameter estimation in our EM algorithm. The result of that derivation is that, instead of (1), we use (6) to compute generalized posterior probabilities, in which the document probability has been raised to the power $1/T$.

$$\mathrm{P}(c_j|d_i;\hat{\theta}) \propto \mathrm{P}(c_j|\hat{\theta}) \left(\prod_{t=1}^{|\mathcal{V}|} \mathrm{P}(w_t|c_j;\hat{\theta})^{N(w_t,d_i)} \right)^{1/T} \tag{6}$$

To build the hierarchy the algorithm starts with an (effectively) infinite temperature so that no matter how many classes EM might assume, the parameter estimates for all of them will converge to the same values during EM. Thus, there is effectively only one class, and it becomes the root of the hierarchy. As the temperature is lowered, the system undergoes phase transitions at which the effective number of clusters grows by splitting one node in the hierarchy into two children [9]. Shrinkage gives meaning to the hierarchy by enforcing (through $\check{\theta}$) some degree of similarity between leaves and their ancestors.

2.4 Adding Event Nodes to the Hierarchy

As shown in Figure 1, the temporal relationships in the data can often provide useful information about the classes to which a document belongs. Once we have a topic hierarchy, we can add another, deeper layer to the hierarchy by splitting the topics into separate events. This is done by clustering the documents by time *separately* within each topic as a mixture of Gaussians. Recall that documents are assigned only probabilistically to each topic, with probability $\mathrm{P}(c_j|d_i)$. During event clustering this probability is held constant and we denote it here by ρ_{ji}. We use EM and DA to cluster in time according to

$$\mathrm{P}(e_{jk}|d_i, c_j) \propto \mathrm{P}(e_{jk}|c_j)\mathrm{P}(\tau_{d_i}|e_{jk})^{1/T}, \tag{7}$$

$$\mathrm{P}(\tau_{d_i}|e_{jk}) = \left[e^{-(\tau_{d_i}-\mu_{jk})^2/(2\sigma_{jk}^2)} / \sqrt{2\pi\sigma_{jk}^2} \right]. \tag{8}$$

Here τ_{d_i} is the time associated with document d_i, and e_{jk} is the k^{th} event cluster for class c_j.

To estimate the parameters of each mixture component, we start from arbitrarily chosen initial values for the parameters $\hat{\pi}$, $\hat{\mu}$, and $\hat{\sigma}$. E-Step: Estimate $\mathrm{P}(e_{jk}|d_i, c_j)$ from $\hat{\pi}$, $\hat{\mu}$ and $\hat{\sigma}$ (7). M-Step: Estimate $\hat{\pi} = \mathrm{P}(e_{jk}|c_j) \propto \sum_{i=1}^{|\mathcal{D}|} \rho_{ji}\mathrm{P}(e_{jk}|d_i, c_j)$; $\hat{\mu}$ and $\hat{\sigma}$ are estimated by

$$\hat{\mu}_{jk} \propto \sum_{i=1}^{|\mathcal{D}|} \rho_{ji}\mathrm{P}(e_{jk}|d_i, c_j)\tau_{d_i}, \quad \hat{\sigma}_{jk} \propto \sum_{i=1}^{|\mathcal{D}|} \rho_{ji}\mathrm{P}(e_{jk}|d_i, c_j)(\tau_{d_i} - \mu_j)^2. \tag{9}$$

Although EM and DA are used to cluster the documents, we assume that there is no hierarchical structure to the events in a topic; thus, we do not build a hierarchy. Instead, when there is a phase transition, the number of clusters is simply increased.

After the documents have been clustered by time into events, we can compute the overall probability that the document belongs to a certain event within a topic as

$$\mathrm{P}(e_{jk}|d_i) \equiv \mathrm{P}(c_j \uparrow e_{jk}|d_i)\mathrm{P}(e_{jk}|d_i, c_j). \tag{10}$$

Here $c_j \uparrow e_{jk}$ denotes the topic that is the parent of event e_{jk}.

Now that we have identified the events within each topic, we can compute the local model parameters $\mathrm{P}(w_t|e_{jk})$ by substituting $\mathrm{P}(e_{jk}|d_i)$ for $\mathrm{P}(c_j|d_i)$ in equations 5.

Then we use EM to do vertical word movement and determine the shrinkage weights λ for each event node. Vertical word movement will have the effect of moving the event specific words out of the topic nodes and into the event node, and vice-versa. Thus the topic nodes will capture more cleanly the topic specific words.

2.5 On-line Event Detection

The procedure we have described is a batch learning procedure. New events can be detected retrospectively using this clustering algorithm. This would be useful, for example, for processing a collection of news stories that a user received while away on vacation. The unread stories would be clustered along with the old stories the user had already read, and the user would be presented only with those event clusters that do not contain any old stories. Often, we may wish to be notified immediately when a document about a new event appears. Thus we desire an *on-line* procedure.

In a sense, without any prior knowledge all documents are new. Thus for on-line novelty detection we will assume that we start with a hierarchy obtained either from existing data or prior knowledge. The task now is to determine for each subsequent document seen whether that document is the first one that reports on a new event.

We do this by augmenting the given hierarchy with extra "garbage collection" event nodes. Each topic node in the hierarchy is given a new child. This child has no

	CMU	UMass	Dragon	New
Recall (%)	.62	.34	.61	.64
Precision (%)	.82	.53	.69	.60
F_1	.79	.60	.75	.62

Table 1: Average recall, precision, and F_1 results for the TDT-1 corpus of 15,863 documents with 1132 documents labelled as belonging to one of 25 events.

documents associated with it, so there are no empirical parameter estimates for the child. To determine the word distributions $Pr(w_t|e_{jk})$ we give each event node a uniform local distribution (in which each word has equal probability $1/|\mathcal{V}|$) and a set of shrinkage weights which are the average of the node's siblings' weights.

When we see a new document, we do not use (10) for classification; instead we classify a document into a leaf using only the word distributions. If the document is most likely in one of the new nodes, then the document is labelled as new. If the document is most likely in an existing event node, then we further threshold the temporal probability $\mathrm{P}(\tau_d|e_{jk})$ for the most likely event. If that probability is too low, then the new document is still labelled as new. The event node a new document is assigned to gives an indication of what kind of new event the document describes via the node's position in the hierarchy.

3 Related Work

A number of different approaches to novelty detection in text have been explored in the context of an on-going government-sponsored initiative called Topic Detection and Tracking (TDT) [1]. Results from the first year of this growing project are discussed in Section 4. The approaches to novelty detection they tried include incremental clustering and hierarchical agglomerative clustering based on TFIDF vector-space models, clustering by constructing information retrieval queries, and clustering using a version of k-means. Novelty (or outlier) detection has been explored in other contexts as well: for example, Tax and Duin [10] use classifier instability to identify outliers, and Moya *et al.* [7] use bounded regions which contain almost all the data. In our approach we use an explicit generative model of the task domain that explicitly accounts for the important temporal aspects of the data. It has been shown [6, 8] that using multiple mixture components per class can sometimes result in better classification performance. Here we have multiple "garbage collection" nodes to capture different kinds of new events.

4 Results

We have done a preliminary evaluation of our system using a data set from the first year of the TDT initiative, which consists of 15,863 news reports covering one year from July 1994 to June 1995, about half of which are articles from the Reuters newswire service and half of which are closed-caption transcripts of CNN broadcast news. From all the events in these documents, 25 were chosen at random to be used for the official evaluation. The number of documents in each of these events ranges from 2 to 295. The total number of labelled documents is 1132.

We used a preliminary version of our algorithm to create a topic and event hierarchy of all 15,863 documents in the TDT data. We used the TDT project's evaluation procedure: for each event we found the best-performing leaf in the hierarchy and used that to compute the precision (P), recall (R), and F_1 ($= 2PR/(P+R)$). Table 1 compares our results, column 'New', with the published results of the members of the TDT project. The best performing system used an order of magnitude more clusters than the 128 clusters we created. We believe that when we can scale our implementation up to generate more clusters, our precision results, and hence our overall F_1, will improve.

5 Conclusion

The main contribution of this paper is the decription of a probabilistic framework in which powerful statistical techniques can be combined in order to address the new and challenging problems of topic detection and tracking. Our approach brings together Expectation-Maximization, shrinkage, vertical word movement and deterministic annealing in a novel way, creating an approach that is more than the sum of its parts. For example, the topic/event word separation described in Section 2.4 is a new and important process that naturally emerges from the combination.

Preliminary experiments indicate that this approach should perform well, and we are currently working to scale up our approach for large-scale comparative evaluations. We are also investigating ways to use event/topic specific words and the temporal clustering phase to determine how large to build the hierarchy.

References

[1] Allan, J., Carbonell, J., Doddington, G., Yamron, J., and Yang, Y. Topic detection and tracking pilot study: Final report. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop* (1998).

[2] Dempster, A. P., Laird, N. M., and Rubin, D. B. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B 39* (1977), 1–38.

[3] Hofmann, T., Puzicha, J., and Jordan, M. I. Learning from dyadic data. In *Advances in Neural Information Processing Systems 11* (1999), M. S. Kearns, S. A. Solla, and D. A. Cohn, Eds., MIT Press.

[4] McCallum, A., and Nigam, K. A comparison of event models for naive Bayes text classification. In *AAAI-98 Workshop on Learning for Text Categorization* (1998).

[5] McCallum, A., Rosenfeld, R., Mitchell, T., and Ng, A. Improving text classification by shrinkage in a hierarchy of classes. In *ICML-98* (1998), pp. 359–367.

[6] McCallum, A. K. Multi-label text classification with a mixture model trained by em. Submitted to NIPS99.

[7] Moya, M. R., Koch, M. W., and Hostetler, L. D. One-class classifier networks for target recognition applications. In *Proceedings World Congress on Neural Networks, International Neural Network Society (INNS)* (1993), pp. 797–801.

[8] Nigam, K., McCallum, A., Thrun, S., and Mitchell, T. Learning to classify text from labeled and unlabeled documents. In *Proceedings of Fifteenth National Conference on Artificial Intelligence (AAAI-98)* (1998), pp. 792–799.

[9] Rose, K. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *Proceedings of the IEEE 86:11* (1998), 2210–2239.

[10] Tax, D. M. J., and Duin, R. P. W. Outlier detection using classifier instability. In *Advances in Pattern Recognition, the Joint IAPR International Workshops* (1998), pp. 593–601.