One-Class Clustering in the Text Domain

Ron Bekkerman
HP Laboratories
Palo Alto, CA 94304, USA
ron.bekkerman@hp.com

Koby Crammer
University of Pennsylvania
Philadelphia, PA 19104, USA
crammer@cis.upenn.edu

Abstract

Having seen a news title "Alba denies wedding reports", how do we infer that it is primarily about Jessica Alba, rather than about weddings or reports? We probably realize that, in a randomly driven sentence, the word "Alba" is less anticipated than "wedding" or "reports", which adds value to the word "Alba" if used. Such anticipation can be modeled as a ratio between an empirical probability of the word (in a given corpus) and its estimated probability in general English. Aggregated over all words in a document, this ratio may be used as a measure of the document's topicality. Assuming that the corpus consists of on-topic and off-topic documents (we call them *the core* and *the noise*), our goal is to determine which documents belong to the core. We propose two unsupervised methods for doing this. First, we assume that words are sampled i.i.d., and propose an information-theoretic framework for determining the core. Second, we relax the independence assumption and use a simple graphical model to rank documents according to their likelihood of belonging to the core. We discuss theoretical guarantees of the proposed methods and show their usefulness for Web Mining and Topic Detection and Tracking (TDT).

1 Introduction

Many intelligent applications in the text domain aim at determining whether a document (a sentence, a snippet etc.) is on-topic or off-topic. In some applications, topics are explicitly given. In binary text classification, for example, the topic is described in terms of positively and negatively labeled documents. In information retrieval, the topic is imposed by a query. In many other applications, the topic is unspecified, however, its existence is assumed. Examples of such applications are within text summarization (extract the most topical sentences), text clustering (group documents that are close topically), novelty detection (reason whether or not test documents are on the same topic as training documents), spam filtering (reject incoming email messages that are too far topically from the content of a personal email repository), etc.

Under the (standard) Bag-Of-Words (BOW) representation of a document, *words* are the functional units that bear the document's topic. Since some words are topical and some are not, the problem of detecting on-topic documents has a dual formulation of detecting topical words. This paper deals with the following questions: (a) Which words can be considered topical? (b) How can topical words be detected? (c) How can on-topic documents be detected given a set of topical words?

The BOW formalism is usually translated into the generative modeling terms by representing documents as multinomial word distributions. For the on-topic/off-topic case, we assume that words in a document are sampled from a mixture of two multinomials: one over topical words and another one over general English (i.e. the background). Obviously enough, the support of the "topic" multinomial is significantly smaller than the support of the background. A document's topicality is then determined by aggregating the topicality of its words (see below for details). Note that by introducing the background distribution we refrain from explicitly modeling the class of off-topic documents—a document is supposed to be off-topic if it is "not topical enough".

Such a formulation of topicality prescribes using the *one-class* modeling paradigm, as opposed to sticking to the *binary* case. Besides being much

Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, pages 41–50,
Honolulu, October 2008.

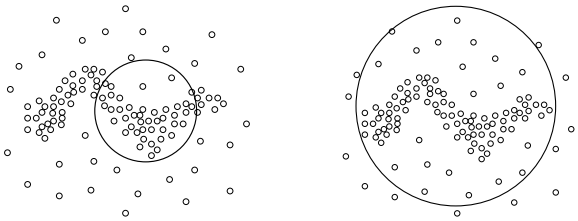


Figure 1: The problem of hyperspherical decision boundaries in one-class models for text, as projected on 2D: (left) a too small portion of the core is captured; (right) too much space around the core is captured.

less widely studied and therefore much more attractive from the scientific point of view, one-class models appear to be more adequate for many real-world tasks, where negative examples are not straightforwardly observable. One-class models separate the desired class of data instances (*the core*) from other data instances (*the noise*). Structure of noise is either unknown, or too complex to be explicitly modeled.

One-class problems are traditionally approached using vector-space methods, where a convex decision boundary is built around the data instances of the desired class, separating it from the rest of the universe. In the text domain, however, those vector-space models are questionably applicable—unlike effective *binary* vector-space models. In binary models, decision boundaries are linear[1], whereas in (vector-space) one-class models, the boundaries are usually *hyperspherical*. Intuitively, since core documents tend to lie on a lower-dimensional manifold (Lebanon, 2005), inducing hyperspherical boundaries may be sub-optimal as they tend to either capture just a portion of the core, or capture too much space around it (see illustration in Figure 1). Here we propose alternative ways for detecting the core, which work well in text.

One-class learning problems have been studied as either outlier detection or identifying a small coherent subset. In one-class outlier detection (Tax and Duin, 2001; Schölkopf et al., 2001), the goal is to identify *a few* outliers from the given set of examples, where the vast majority of the examples are considered relevant. Alternatively, a complementary goal is to distill a subset of relevant examples, in the space with many outliers (Crammer and Chechik, 2004; Gupta and Ghosh, 2005; Crammer et al., 2008). Most of the one-class approaches employ geometrical concepts to capture the notion of relevancy (or irrelevancy) using either hyperplanes (Schölkopf et al., 2001) or hyperspheres (Tax and Duin, 2001; Crammer and Chechik, 2004; Gupta and Ghosh, 2005). In this paper we adopt the latter approach: we formulate one-class clustering in text as an optimization task of identifying the *most coherent* subset (*the core*) of k documents drawn from a given pool of $n > k$ documents.[2]

Given a collection $\mathcal{D}$ of on-topic and off-topic documents, we assume that on-topic documents share a portion of their vocabulary that consists of "relatively rare" words, i.e. words that are used in $\mathcal{D}$ more often than they are used in general English. We call them *topical* words. For example, if some documents in $\mathcal{D}$ share words such as "Bayesian", "classifier", "reinforcement" and other machine learning terms (infrequent in general English), whereas other documents do not seem to share any subset of words (besides stopwords), then we conclude that the machine learning documents compose the core of $\mathcal{D}$, while non-machine learning documents are noise.

We express the level of topicality of a word w in terms of the ratio $\rho(w) = \frac{p(w)}{q(w)}$, where $p(w)$ is w's empirical probability (in $\mathcal{D}$), and $q(w)$ is its estimated probability in general English. We discuss an interesting characteristic of $\rho(w)$: if $\mathcal{D}$ is large enough, then, with high probability, $\rho(w)$ values are greater for topical words than for non-topical words. Therefore, $\rho(w)$ can be used as a mean to measure the topicality of w.

Obviously, the quality of this measure depends on the quality of estimating $q(w)$, i.e. the general English word distribution, which is usually estimated over a large text collection. The larger the collection is, the better would be the estimation. Recently, Google has released the `Web 1T` dataset[3] that provides $q(w)$ estimated on a text collection of one trillion tokens. We use it in our experimentation.

We propose two methods that use the ρ ratio to

[1] As such, or after applying the *kernel trick* (Cristianini and Shawe-Taylor, 2000)

[2] The parameter k is analogous to the number of clusters in (multi-class) clustering, as well as to the number of outliers (Tax and Duin, 2001) or the radius of Bregmanian ball (Crammer and Chechik, 2004)—in other formulations of one-class clustering.

[3] `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2006T13`

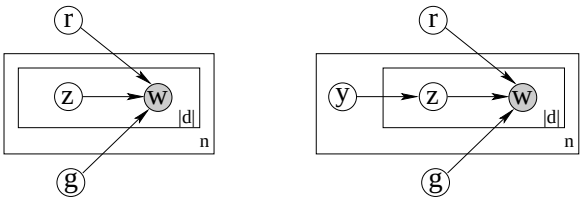


Figure 2: (left) A simple generative model; (right) Latent Topic/Background model (Section 4).

solve the one-class clustering problem. First, we express documents' topicality in terms of aggregating their words' ρ ratios into an information-theoretic "topicality measure". The core is then composed of k documents with the highest topicality measure. We show that the proposed measure is optimal for constructing the core cluster among documents of equal length. However, our method is not useful in a setup where some long documents have a topical *portion*: such documents should be considered on-topic, but their heavy tail of background words overcomes the topical words' influence. We generalize our method to non-equally-long documents by first extracting words that are supposed to be topical and then projecting documents over those words. Such projection preserves the optimality characteristic and results in constructing a more accurate core cluster in practice. We call such a method of choosing both topical words and core documents *One-Class Co-Clustering (OCCC)*.

It turns out that our OCCC method's performance depends heavily on choosing the number of topical words. We propose a heuristic for setting this number. As another alternative, we propose a method that does not require tuning this parameter: we use words' ρ ratios to initialize an EM algorithm that computes the likelihood of documents to belong to the core—we then choose k documents of maximal likelihood. We call this model the *Latent Topic/Background (LTB)* model. LTB outperforms OCCC in most of our test cases.

Our one-class clustering models have interesting cross-links with models applied to other Information Retrieval tasks. For example, a model that resembles our OCCC, is proposed by Zhou and Croft (2007) for *query performance prediction*. Tao and Zhai (2004) describe a *pseudo-relevance feedback* model that is similar to our LTB. These types of cross-links are common for the models that are

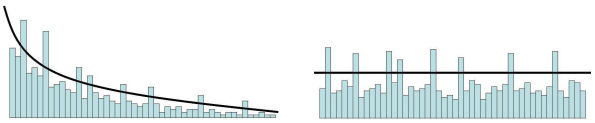


Figure 3: (left) Words' $p(w)$ values when sorted by their $q(w)$ values; (right) words' $\rho(w)$ values.

general enough and relatively simple. In this paper we put particular emphasis on the simplicity of our models, such that they are feasible for theoretical analysis as well as for efficient implementation.

2 Motivation for using ρ ratios

Recall that we use the $\rho(w) = \frac{p(w)}{q(w)}$ ratios to express the level of our "surprise" of seeing the word w. A high value of $\rho(w)$ means that w is used in the corpus more frequently than in general English, which, we assume, implies that w is topical. The more topical words a document contains, the more "topical" it is—k most topical documents compose the core $\mathcal{D}^k \subset \mathcal{D}$.

An important question is whether or not the ρ ratios are sufficient to detecting the *actually* topical words. To address this question, let us model the corpus $\mathcal{D}$ using a simple graphical model (Figure 2 left). In this model, the word distribution $p(w)$ is represented as a mixture of two multinomial distributions: p_r over a set $\mathcal{R}$ of topical words, and p_g over all the words $\mathcal{G} \supset \mathcal{R}$ in $\mathcal{D}$. For each word w_{ij} in a document d_i, we toss a coin Z_{ij}, such that, if $Z_{ij} = 1$, then w_{ij} is sampled from p_r, otherwise it is sampled from p_g. Define $\pi \triangleq p(Z_{ij} = 1)$.

If $|\mathcal{G}| \gg |\mathcal{R}| \gg 0$, and if $\pi \gg 0$, then topical words would tend to appear more often than non-topical words. However, we cannot simply base our conclusions on word counts, as some words are naturally more frequent than others (in general English). Figure 3 (left) illustrates this observation: it shows words' $p(w)$ values sorted by their $q(w)$ values. It is hard to fit a curve that would separate between $\mathcal{R}$ and $\mathcal{G} \setminus \mathcal{R}$. We notice however, that we can "flatten" this graph by drawing $\rho(w)$ values instead (see Figure 3 right). Here, naturally frequent words are penalized by the q factor, so we can assume that, when re-normalized, $\rho(w)$ behaves as a mixture of two discrete *uniform* distributions. A simple threshold can then separate between $\mathcal{R}$ and $\mathcal{G} \setminus \mathcal{R}$.

Proposition 1 *Under the uniformity assumption, it is sufficient to have a* log-linear *size sample (in $|\mathcal{G}|$) in order to determine the set $\mathcal{R}$ with high probability.*

See Bekkerman (2008) for the proof. The proposition states that in corpora of practical size[4] the set of topical words can be almost perfectly detected, simply by taking words with the highest ρ ratios. Consequently, the core $\mathcal{D}^k$ will consist of k documents, each of which contains more topical words than any document from $\mathcal{D} \setminus \mathcal{D}^k$.

To illustrate this theoretical result, we followed the generative process as described above, and constructed an artificial dataset with characteristics similar to those of our WAD dataset (see Section 5.1). In particular, we fixed the size of the artificial dataset to be equal to the size of the WAD dataset ($N = 330,000$). We set the ratio of topical words to 0.2 and assumed uniformity of the ρ values. In this setup, we were able to detect the set of topical words with a 98.5% accuracy.

2.1 Max-KL Algorithm

In this section, we propose a simple information-theoretic algorithm for identifying the core $\mathcal{D}^k$, and show that it is optimal under the uniformity assumption. Given the ρ ratios of words, the *aggregated topicality* of the corpus $\mathcal{D}$ can be expressed in terms of the KL-divergence:

$$\begin{aligned} KL(p||q) &= \sum_{w\in\mathcal{G}} p(w)\log\frac{p(w)}{q(w)} \\ &= \sum_{d\in\mathcal{D},w\in\mathcal{G}} p(d,w)\log\frac{p(w)}{q(w)}. \end{aligned}$$

A document d's contribution to the aggregated topicality measure will assess the topicality of d:

$$KL_d(p||q) = \sum_{w\in\mathcal{G}} p(d,w)\log\frac{p(w)}{q(w)}. \qquad (1)$$

The core $\mathcal{D}^k$ will be composed of documents with the highest topicality scores. A simple, greedy algorithm for detecting $\mathcal{D}^k$ is then:

1. Sort documents according to their topicality value (1), in decreasing order.
2. Select the first k documents.

[4] $N = O(m\log m)$, where N is the number of word tokens in $\mathcal{D}$, and $m = |\mathcal{G}|$ is the size of the vocabulary.

Since the algorithm chooses documents with high values of the KL divergence we call it the *Max-KL* algorithm. We now argue that it is optimal under the uniformity assumption. Indeed, if the corpus $\mathcal{D}$ is large enough, then according to Proposition 1 (with high probability) any topical word w has a lower ρ ratio than any non-topical word. Assume that all documents are of the same length ($|d|$ is constant). The Max-KL algorithm chooses documents that contain more topical words than any other document in the corpus—which is exactly the definition of the core, as presented in Section 1. We summarize this observation in the following proposition:

Proposition 2 *If the corpus $\mathcal{D}$ is large enough, and all the documents are of the same length, then the Max-KL algorithm is optimal for the one-class clustering problem under the uniformity assumption.*

In contrast to the (quite natural) uniformity assumption, the all-the-same-length assumption is quite restrictive. Let us now propose an algorithm that overcomes this issue.

3 One-Class Co-Clustering (OCCC)

As accepted in Information Retrieval, we decide that a document is on-topic if it has a topical *portion*, no matter how long its non-topical portion is. Therefore, we decide about documents' topicality based on topical words only—non-topical words can be completely disregarded. This observation leads us to proposing a one-class *co-clustering* (OCCC) algorithm: we first detect the set $\mathcal{R}$ of topical words, represent documents over $\mathcal{R}$, and then detect $\mathcal{D}^k$ based on the new representation.[5]

We reexamine the document's topicality score (1) and omit non-topical words. The new score is then:

$$KL^r_d(p||q) = \sum_{w\in\mathcal{R}} p'(d,w)\log\frac{p(w)}{q(w)}, \qquad (2)$$

where $p'(d,w) = p(d,w)/(\sum_{w\in\mathcal{R}} p(d,w))$ is a joint distribution of documents and (only) topical words. The OCCC algorithm first uses $\rho(w)$ to

[5] OCCC is the simplest, *sequential* co-clustering algorithm, where words are clustered prior to clustering documents (see, e.g., Slonim and Tishby (2000)). In OCCC, word clustering is analogous to *feature selection*. More complex algorithms can be considered, where this analogy is less obvious.

choose the most topical words, then it projects documents on these words and apply the *Max-KL* algorithm, as summarized below:

1. Sort words according to their ρ ratios, in decreasing order.
2. Select a subset $\mathcal{R}$ of the first m_r words.
3. Represent documents as bags-of-words over $\mathcal{R}$ (delete counts of words from $\mathcal{G} \setminus \mathcal{R}$).
4. Sort documents according to their topicality score (2), in decreasing order.
5. Select a subset $\mathcal{D}^k$ of the first k documents.

Considerations analogous to those presented in Section 2.1, lead us to the following result:

Proposition 3 *If the corpus $\mathcal{D}$ is large enough, the OCCC algorithm is optimal for one-class clustering of documents, under the uniformity assumption.*

Despite its simplicity, the OCCC algorithm shows excellent results on real-world data (see Section 5). OCCC's time complexity is particularly appealing: $O(N)$, where N is the number of word tokens in $\mathcal{D}$.

3.1 Choosing size m_r of the word cluster

The choice of $m_r = |\mathcal{R}|$ can be crucial. We propose a useful heuristic for choosing it. We assume that the distribution of ρ ratios for $w \in \mathcal{R}$ is a Gaussian with a mean $\mu_r \gg 1$ and a variance σ_r^2, and that the distribution of ρ ratios for $w \in \mathcal{G} \setminus \mathcal{R}$ is a Gaussian with a mean $\mu_{nr} = 1$ and a variance σ_{nr}^2. We also assume that *all* the words with $\rho(w) < 1$ are non-topical. Since Gaussians are symmetric, we further assume that the number of non-topical words with $\rho(w) < 1$ equals the number of non-topical words with $\rho(w) \geq 1$. Thus, our estimate of $|\mathcal{G} \setminus \mathcal{R}|$ is twice the number of words with $\rho(w) < 1$, and then the number of topical words can be estimated as $m_r = |\mathcal{G}| - 2 \cdot \#\{\text{words with } \rho(w) < 1\}$.

4 Latent Topic/Background (LTB) model

Instead of sharply thresholding topical and non-topical words, we can have them all, *weighted* with a probability of being topical. Also, we notice that our original generative model (Figure 2 left) assumes that words are i.i.d. sampled, which can be relaxed by deciding on the document topicality first. In our new generative model (Figure 2 right), for each document d_i, Y_i is a Bernoulli random variable where

Algorithm 1 EM algorithm for one-class clustering using the LTB model.

Input:
$\mathcal{D}$ – the dataset
$\rho(w_l) = \frac{p(w_l)}{q(w_l)}$ – ρ scores for each word $w_l|_{l=1}^m$
T – number of EM iterations
Output: Posteriors $p(Y_i = 1|d_i, \Theta^T)$ for each doc $d_i|_{i=1}^n$

Initialization:
for each document d_i **initialize** π_i^1
for each word w_l **initialize** $p_r^1(w_l) = \Omega_r \rho(w_l)$;
$p_g^1(w_l) = \frac{\Omega_g}{\rho(w_l)}$, s.t. Ω_r and Ω_g are normalization factors

Main loop:
for all $t = 1, \ldots, T$ **do**
E-step:
for each document d_i **compute** $\alpha_i^t = p(Y_i = 1|d_i, \Theta^t)$
for each word token w_{ij} **compute**

$$\beta_{ij}^t = p(Z_{ij} = 1|Y_i = 1, w_{ij}, \Theta^t)$$

M-step:
for each document d_i **update** $\pi^{t+1} = \frac{1}{|d_i|} \sum_j \beta_{ij}^t$
for each word w_l **update**

$$p_r^{t+1}(w_l) = \frac{\sum_i \alpha_i^t \sum_j \delta(w_{ij} = w_l)\, \beta_{ij}^t}{\sum_i \alpha_i^t \sum_j \beta_{ij}^t}$$

$$p_g^{t+1}(w_l) = \frac{N_w - \sum_i \alpha_i^t \sum_j \delta(w_{ij} = w_l)\, \beta_{ij}^t}{N - \sum_i \alpha_i^t \sum_j \beta_{ij}^t}$$

$Y_i = 1$ corresponds to d_i being on-topic. As before, Z_{ij} decides on the topicality of a word token w_{ij}, but now *given* Y_i. Since not all words in a core document are supposed to be topical, then for each word of a core document we make a separate decision (based on Z_{ij}) whether it is sampled from $p_r(W)$ or $p_g(W)$. However, if a document does not belong to the core ($Y_i = 0$), each its word is sampled from $p_g(W)$, i.e. $p(Z_{ij} = 0|Y_i = 0) = 1$.

Inspired by Huang and Mitchell (2006), we use the Expectation-Maximization (EM) algorithm to *exactly* estimate parameters of our model from the dataset. We now describe the model parameters Θ. First, the probability of any document to belong to the core is denoted by $p(Y_i = 1) = \frac{k}{n} = p_d$ (this parameter is fixed and will not be learnt from data). Second, for each document d_i, we maintain a probability of each its word to be topical given that the document is on-topic, $p(Z_{ij} = 1|Y_i = 1) = \pi_i$ for $i = 1, \ldots, n$. Third, for each word w_l (for $k = 1...m$), we let $p(w_l|Z_l = 1) = p_r(w_l)$ and $p(w_l|Z_l = 0) = p_g(w_l)$. The overall number of pa-

rameters is $n + 2m + 1$, one of which (p_d) is preset. The dataset likelihood is then:

$$\begin{aligned} p(\mathcal{D}) &= \prod_{i=1}^{n} [p_d \, p(d_i|Y_i = 1) + (1 - p_d) p(d_i|Y_i = 0)] \\ &= \prod_{i=1}^{n} \left[p_d \prod_{j=1}^{|d_i|} [\pi_i p_r(w_{ij}) + (1 - \pi_i) p_g(w_{ij})] \right. \\ &\quad \left. + (1 - p_d) \prod_{j=1}^{|d_i|} p_g(w_{ij}) \right]. \end{aligned}$$

At each iteration t of the EM algorithm, we first perform the E-step, where we compute the posterior distribution of hidden variables $\{Y_i\}$ and $\{Z_{ij}\}$ given the current parameter values Θ^t and the data $\mathcal{D}$. Then, at the M-step, we compute the new parameter values Θ^{t+1} that maximize the model log-likelihood given Θ^t, $\mathcal{D}$ and the posterior distribution.

The initialization step is crucial for the EM algorithm. Our pilot experimentation showed that if distributions $p_r(W)$ and $p_g(W)$ are initialized as uniform, the EM performance is close to random. Therefore, we decided to initialize word probabilities using normalized ρ scores. We do not propose the optimal way to initialize π_i parameters, however, as we show later in Section 5, our LTB model appears to be quite robust to the choice of π_i.

The EM procedure is presented in Algorithm 1. For details, see Bekkerman (2008). After T iterations, we sort the documents according to α_i in decreasing order and choose the first k documents to be the core. The complexity of Algorithm 1 is linear: $O(TN)$. To avoid overfitting, we set T to be a small number: in our experiments we fix $T = 5$.

5 Experimentation

We evaluate our OCCC and LTB models on two applications: a *Web Mining* task (Section 5.1), and a *Topic Detection and Tracking (TDT)* (Allan, 2002) task (Section 5.2).

To define our evaluation criteria, let C be the constructed cluster and let C_r be its portion consisting of documents that actually belong to the core. We define precision as Prec $= |C_r|/|C|$, recall as Rec $= |C_r|/k$ and F-measure as (2 Prec Rec)/(Prec+Rec). Unless stated otherwise, in our experiments we fix $|C| = k$, such that precision equals recall and is then called *one-class clustering accuracy*, or just *accuracy*.

We applied our one-class clustering methods in four setups:

- **OCCC with the heuristic to choose** m_r (from Section 3.1).
- **OCCC with optimal** m_r**.** We *unfairly* choose the number m_r of topical words such that the resulting accuracy is maximal. This setup can be considered as the upper limit of the OCCC's performance, which can be hypothetically achieved if a better heuristic for choosing m_r is proposed.
- **LTB initialized with** $\pi_i = 0.5$ **(for each** i**).** As we show in Section 5.1 below, the LTB model demonstrates good performance with this straightforward initialization.
- **LTB initialized with** $\pi_i = p_d$**.** Quite naturally, the number of topical words in a dataset depends on the number of core documents. For example, if the core is only 10% of a dataset, it is unrealistic to assume that 50% of all words are topical. In this setup, we condition the ratio of topical words on the ratio of core documents.

We compare our methods with two existing algorithms: (a) One-Class SVM clustering[6] (Tax and Duin, 2001); (b) One-Class Rate Distortion (OC-RD) (Crammer et al., 2008). The later is considered a state-of-the-art in one-class clustering. Also, to establish the lowest baseline, we show the result of a random assignment of documents to the core $\mathcal{D}^k$.

The OC-RD algorithm is based on rate-distortion theory and expresses the one-class problem as a lossy coding of each instance into a few possible instance-dependent codewords. Each document is represented as a distribution over words, and the KL-divergence is used as a distortion function (generally, it can be any Bregman function). The algorithm also uses an "inverse temperature" parameter (denoted by β) that represents the tradeoff between compression and distortion. An annealing process is employed, in which the algorithm is applied with a sequence of increasing values of β, when initialized with the result obtained at the previous itera-

[6] We used Chih-Jen Lin's `LibSVM` with the `-s 2` parameter. We provided the core size using the `-n` parameter.

Method	WAD	TW
Random assignment	38.7%	$34.9 \pm 3.1\%$
One-class SVM	46.3%	$45.2 \pm 3.2\%$
One-class rate distortion	48.8%	$63.6 \pm 3.5\%$
OCCC with the m_r heuristic	80.2%	$61.4 \pm 4.5\%$
OCCC with optimal m	82.4%	$68.3 \pm 3.6\%$
LTB initialized with $\pi_i = 0.5$	79.8%	$65.3 \pm 7.3\%$
LTB initialized with $\pi_i = p_d$	78.3%	$68.0 \pm 5.9\%$

Table 1: One-class clustering accuracy of our OCCC and LTB models on the WAD and the TW detection tasks, as compared to OC-SVM and OC-RD. For TW, the accuracies are macro-averaged over the 26 weekly chunks, with the standard error of the mean presented after the $\pm$ sign.

tion. The outcome is a sequence of cores with decreasing sizes. The annealing process is stopped once the largest core size is equal to k.

5.1 Web appearance disambiguation

Web appearance disambiguation (WAD) is proposed by Bekkerman and McCallum (2005) as the problem of reasoning whether a particular mention of a person name in the Web refers to the person of interest or to his or her unrelated namesake. The problem is solved given *a few* names of people from one social network, where the objective is to construct a cluster of Web pages that mention names of related people, while filtering out pages that mention their unrelated namesakes.

WAD is a classic one-class clustering task, that is tackled by Bekkerman and McCallum with *simulated* one-class clustering: they use a sophisticated agglomerative/conglomerative clustering method to construct multiple clusters, out of which one cluster is then selected. They also use a simple *link structure (LS)* analysis method that matches hyperlinks of the Web pages in order to compose a cloud of pages that are close to each other in the Web graph. The authors suggest that the best performance can be achieved by a hybrid of the two approaches.

We test our models on the WAD dataset,[7] which consists of 1085 Web pages that mention 12 people names of AI researchers, such as Tom Mitchell and Leslie Kaelbling. Out of the 1085 pages, 420 are on-topic, so we apply our algorithms with $k = 420$. At a preprocessing step, we binarize document vectors and remove low frequent words (both in terms of $p(w)$ and $q(w)$). The results are summarized in the middle column of Table 1. We can see that both OCCC and LTB dramatically outperform their competitors, while showing practically indistinguishable results compared to each other. Note that when the size of the word cluster in OCCC is *unfairly* set to its optimal value, $m_r = 2200$, the OCCC method is able to gain a 2% boost. However, for obvious reasons, the optimal value of m_r may not always be obtained in practice.

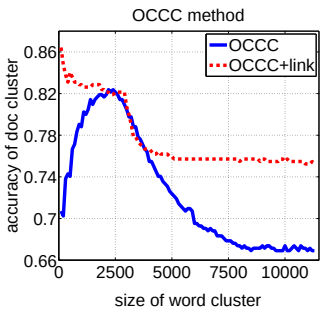
[7] http://www.cs.umass.edu/~ronb/name_disambiguation.html

#	OCCC	LTB
1	cheyer	artificial
2	kachites	learning
3	quickreview	cs
4	adddoc	intelligence
5	aaai98	machine
6	kaelbling	edu
7	mviews	algorithms
8	mlittman	proceedings
9	hardts	computational
10	meuleau	reinforcement
11	dipasquo	papers
12	shakshuki	cmu
13	xevil	aaai
14	sangkyu	workshop
15	gorfu	kaelbling

Table 2: Most highly ranked words by OCCC and LTB, on the WAD dataset.

Table 2 lists a few most topical words according to the OCCC and LTB models. The OCCC algorithm sorts words according to their ρ scores, such that words that often occur in the dataset but rarely in the Web, are on the top of the list. These are mostly last names or login names of researchers, venues etc. The EM algorithm of LTB is the given ρ scores as an input to initialize $p_r^1(w)$ and $p_g^1(w)$, which are then updated at each M-step. In the LTB columns, words are sorted by $p_r^5(w)$. High quality of the LTB list is due to conditional dependencies in our generative model (via the Y_i nodes).

Solid lines in Figure 4 demonstrate the robustness of our models to tuning their main parameters (m_r for OCCC, and the π_i initialization for LTB). As can be seen from the left panel, OCCC shows robust performance: the accuracy above 80% is obtained when the word cluster is of any size in the 1000–3000 range. The heuristic from Section 3.1 suggests a cluster size of 1000. The LTB is even more robust: practically any value of π_i (besides the very large ones, $\pi_i \approx 1$) can be chosen.



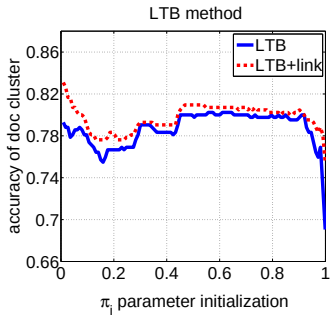


Figure 4: **Web appearance disambiguation:** (left) OCCC accuracy as a function of the word cluster size; (right) LTB accuracy over various initializations of π_i parameters. The red dotted lines show the accuracy of each method's results combined with the Link Structure model results. On the absolute scale, OCCC outperforms LTB, however LTB shows more robust behavior than OCCC.

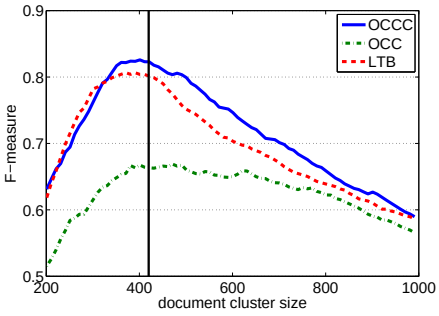


Figure 5: **Web appearance disambiguation:** F-measure as a function of document cluster size: a vertical line indicates the point where precision equals recall (and therefore equals accuracy). "OCC" refers to the OCCC model where all the words are taken as the word cluster (i.e. no word filtering is done).

To perform a fair comparison of our results with those obtained by Bekkerman and McCallum (2005), we construct hybrids of their link structure (LS) analysis model with our OCCC and LTB, as follows. First, we take their LS core cluster, which consists of 360 documents. Second, we pass over all the WAD documents in the order as they were ranked by either OCCC or LTB, and enlarge the LS core with 60 most highly ranked documents that did not occur in the LS core. In either case, we end up with a hybrid core of 420 documents.

Dotted lines in Figure 4 show accuracies of the resulting models. As the F-measure of the hybrid model proposed by Bekkerman and McCallum (2005) is 80.3%, we can see that it is significantly inferior to the results of either OCCC+LS or LTB+LS, when their parameters are set to a small value ($m_r < 3000$ for OCCC, $\pi_i < 0.06$ for LTB). Such a choice of parameter values can be explained by the fact that we need only 60 documents to expand the LS core cluster to the required size $k = 420$. When the values of m_r and π_i are small, both OCCC and LTB are able to build very small and very precise core clusters, which is exactly what we need here. The OCCC+LS hybrid is particularly successful, because it uses non-canonical words (see Table 2) to compose a clean core that almost does not overlap with the LS core. Remarkably, the OCCC+LS model obtains 86.4% accuracy with $m_r = 100$, which is the state-of-the-art result on the WAD dataset.

To answer the question how much our models are sensitive to the choice of the core size k, we computed the F-measure of both OCCC and LTB as a function of k (Figure 5). It turns out that our methods are quite robust to tuning k: choosing any value in the 300–500 range leads to good results.

5.2 Detecting the topic of the week

Real-world data rarely consists of a clean core and uniformly distributed noise. Usually, the noise has some structure, namely, it may contain coherent components. With this respect, one-class clustering can be used to detect the *largest* coherent component in a dataset, which is an integral part of many applications. In this section, we solve the problem of automatically detecting the *Topic of the Week (TW)* in a newswire stream, i.e. detecting all articles in a weekly news roundup that refer to the most broadly discussed event.

We evaluate the TW detection task on the benchmark TDT-5 dataset[8], which consists of 250 news events spread over a time period of half a year, and 9,812 documents in English, Arabic and Chinese (translated to English), annotated by their relationship to those events.[9] The largest event in TDT-5 dataset (#55106, titled *"Bombing in Riyadh, Saudi Arabia"*) has 1,144 documents, while 66 out of the 250 events have only one document each. We split the dataset to 26 weekly chunks (to have 26 full

[8] `http://projects.ldc.upenn.edu/TDT5/`

[9] We take into account only labeled documents, while ignoring unlabeled documents that can be found in the TDT-5 data.

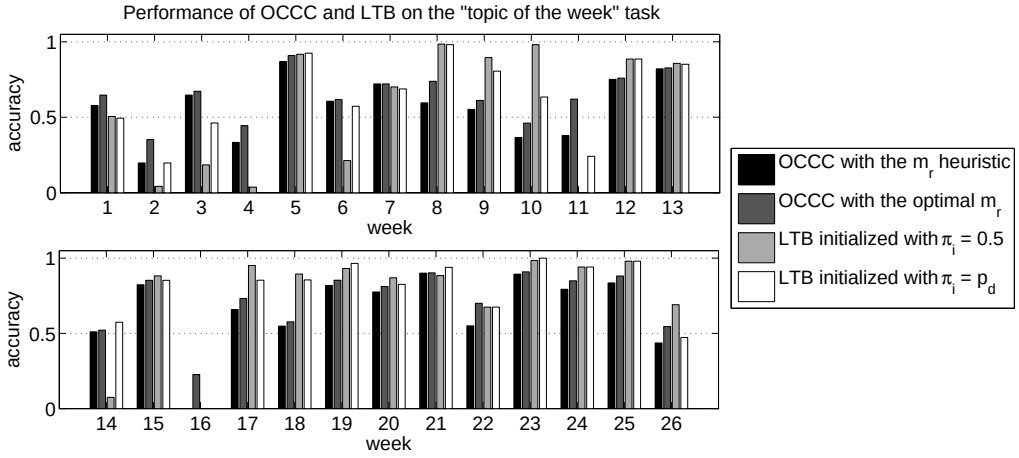


Figure 6: **"Topic of the week" detection task:** Accuracies of two OCCC methods and two LTB methods.

weeks, we delete all the documents dated with the last day in the dataset, which decreases the dataset's size to 9,781 documents). Each chunk contains from 138 to 1292 documents.

The one-class clustering accuracies, macro-averaged over the 26 weekly chunks, are presented in the right column of Table 1. As we can see, both LTB models, as well as OCCC with the optimal m_r, outperform our baselines. Interestingly, even the optimal choice of m_r does not lead OCCC to significantly superior results while compared with LTB. The dataset-dependent initialization of LTB's π_i parameters ($\pi_i = p_d$) appears to be preferable over the dataset-independent one ($\pi_i = 0.5$).

Accuracies *per week* are shown in Figure 6. These results reveal two interesting observations. First, OCCC tends to outperform LTB only on data chunks where the results are quite low in general (less than 60% accuracy). Specifically, on weeks 2, 4, 11, and 16 the LTB models show extremely poor performance. While investigating this phenomenon, we discovered that in two of the four cases LTB was able to construct very clean core clusters, however, those clusters corresponded to the *second* largest topic, while we evaluate our methods on the first largest topic.[10] Second, the (completely unsupervised) LTB model can obtain very good results on some of the data chunks. For example, on weeks 5, 8, 19, 21, 23, 24, and 25 the LTB's accuracy is above 90%, with a striking 100% on week-23.

6 Conclusion

We have developed the theory and proposed practical methods for one-class clustering in the text domain. The proposed algorithms are very simple, very efficient and still surprisingly effective. More sophisticated algorithms (e.g. an *iterative*[11] version of OCCC) are emerging.

7 Acknowledgements

We thank Erik Learned-Miller for the inspiration on this project. We also thank Gunjan Gupta, James Allan, and Fernando Diaz for fruitful discussions. This work was supported in part by the Center for Intelligent Information Retrieval and in part by the Defense Advanced Research Projects Agency (DARPA) under contract number HR0011-06-C-0023. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor.

References

J. Allan, editor. 2002. *Topic detection and tracking: event-based information organization*. Kluwer Academic Publishers.

[10] For example, on the week-4 data, topic #55077 (*"River ferry sinks on Bangladeshi river"*) was discovered by LTB as the largest and most coherent one. However, in that dataset, topic #55077 is represented by 20 documents, while topic #55063 (*"SARS Quarantined medics in Taiwan protest"*) is represented by 27 documents, such that topic #55077 is in fact the second largest one.

[11] See, e.g., El-Yaniv and Souroujon (2001)

R. Bekkerman and A. McCallum. 2005. Disambiguating web appearances of people in a social network. In *Proceedings of WWW-05, the 14th International World Wide Web Conference*.

R. Bekkerman. 2008. *Combinatorial Markov Random Fields and their Applications to Information Organization*. Ph.D. thesis, University of Massachusetts at Amherst.

K. Crammer and G. Chechik. 2004. A needle in a haystack: local one-class optimization. In *Proceedings of the 21st International Conference on Machine Learning*.

K. Crammer, P. Talukdar, and F. Pereira. 2008. A rate-distortion one-class model and its applications to clustering. In *Proceedings of the 25st International Conference on Machine Learning*.

N. Cristianini and J. Shawe-Taylor. 2000. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press.

R. El-Yaniv and O. Souroujon. 2001. Iterative double clustering for unsupervised and semi-supervised learning. In *Advances in Neural Information Processing Systems (NIPS-14)*.

G. Gupta and J. Ghosh. 2005. Robust one-class clustering using hybrid global and local search. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 273–280.

Y. Huang and T. Mitchell. 2006. Text clustering with extended user feedback. In *Proceedings of the 29th annual international ACM SIGIR conference*, pages 413–420.

G. Lebanon. 2005. *Riemannian Geometry and Statistical Machine Learning*. Ph.D. thesis, CMU.

B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson. 2001. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471.

N. Slonim and N. Tishby. 2000. Document clustering using word clusters via the information bottleneck method. In *Proceedings of the 23rd annual international ACM SIGIR conference*, pages 208–215.

T. Tao and C. Zhai. 2004. A two-stage mixture model for pseudo feedback. In *Proceedings of the 27th annual international ACM SIGIR conference*, pages 486–487.

D. M. J. Tax and R. P. W. Duin. 2001. Outliers and data descriptions. In *Proceedings of the 7th Annual Conference of the Advanced School for Computing and Imaging*, pages 234–241.

Y. Zhou and W. B. Croft. 2007. Query performance prediction in web search environments. In *Proceedings of the 30th Annual International ACM SIGIR Conference*.