Unsupervised Learning for Lexicon-Based Classification

Jacob Eisenstein
Georgia Institute of Technology
801 Atlantic Drive NW
Atlanta, Georgia 30318

Abstract

In lexicon-based classification, documents are assigned labels by comparing the number of words that appear from two opposed lexicons, such as positive and negative sentiment. Creating such words lists is often easier than labeling instances, and they can be debugged by non-experts if classification performance is unsatisfactory. However, there is little analysis or justification of this classification heuristic. This paper describes a set of assumptions that can be used to derive a probabilistic justification for lexicon-based classification, as well as an analysis of its expected accuracy. One key assumption behind lexicon-based classification is that all words in each lexicon are equally predictive. This is rarely true in practice, which is why lexicon-based approaches are usually outperformed by supervised classifiers that learn distinct weights on each word from labeled instances. This paper shows that it is possible to learn such weights without labeled data, by leveraging co-occurrence statistics across the lexicons. This offers the best of both worlds: light supervision in the form of lexicons, and data-driven classification with higher accuracy than traditional word-counting heuristics.

Introduction

Lexicon-based classification refers to a classification rule in which documents are assigned labels based on the count of words from lexicons associated with each label (Taboada et al. 2011). For example, suppose that we have opposed labels $Y \in \{0, 1\}$, and we have associated lexicons $\mathcal{W}_0$ and $\mathcal{W}_1$. Then for a document with a vector of word counts $\boldsymbol{x}$, the lexicon-based decision rule is,

$$\sum_{i \in \mathcal{W}_0} x_i \gtrless \sum_{j \in \mathcal{W}_1} x_j, \tag{1}$$

where the $\gtrless$ operator indicates a decision rule. Put simply, we select the label whose lexicons matches the most words.

Lexicon-based classification is widely used in industry and academia, with applications ranging from sentiment classification and opinion mining (Pang and Lee 2008; Liu 2015) to the psychological (Tausczik and Pennebaker 2010) and ideological (Laver and Garry 2000) analysis of texts. The popularity of this approach can be explained by its relative simplicity and ease of use: for domain experts, creating lexicons is intuitive, and, in comparison with labeling instances, may offer a faster path towards a reasonably accurate classifier (Settles 2011). Furthermore, classification errors can be iteratively debugged by refining the lexicons.

However, from a machine learning perspective, there are a number of drawbacks to lexicon-based classification. First, while intuitively reasonable, lexicon-based classification lacks theoretical justification: it is not clear what conditions are necessary for it to work. Second, the lexicons may be incomplete, even for designers with strong substantive intuitions. Third, lexicon-based classification assigns an equal weight to each word, but some words may be more strongly predictive than others.[1] Fourth, lexicon-based classification ignores multi-word phenomena, such as negation (e.g., *not so good*) and discourse (e.g., *the movie would be watchable if it had better acting*). Supervised classification systems, which are trained on labeled examples, tend to outperform lexicon-based classifiers, even without accounting for multi-word phenomena (Liu 2015; Pang and Lee 2008).

Several researchers have addressed the challenge of **lexicon expansion**, automatically growing lexicons from an initial seed set (Hatzivassiloglou and McKeown 1997; Qiu et al. 2011). There is also work on handling multi-word phenomena such as negation (Wilson, Wiebe, and Hoffmann 2005; Polanyi and Zaenen 2006), and discourse (Somasundaran, Wiebe, and Ruppenhofer 2008; Bhatia, Ji, and Eisenstein 2015). However, the underlying theoretical foundations of lexicon-based classification remain poorly understood, and we lack principled means for automatically assigning weights to lexicon items without resorting to labeled instances.

This paper elaborates a set of assumptions under which lexicon-based classification is equivalent to minimum Bayes risk classification. We then derive expected error rates under these assumptions. These expected error rates are not matched by observations on real data, suggesting that the underlying assumptions are invalid. Of key importance is the assumption that each lexicon item is equally predictive. To relax this assumption, we derive a principled method for estimating word probabilities under each label,

[1] Some lexicons attach coarse-grained predefined weights to each word. For example, the OpinionFinder Subjectivity lexicon labels words as "strongly" or "weakly" subjective (Wilson, Wiebe, and Hoffmann 2005). This poses an additional burden on the lexicon creator.

using a method-of-moments estimator on cross-lexical co-occurrence counts.

Overall, this paper makes the following contributions:

- justifying lexicon-based classification as minimum Bayes risk classification in a multinomial model;
- mathematically analyzing this model to compute the expected performance of lexicon-based classifiers;
- extending the model to justify a popular variant of lexicon-based classification, which incorporates word presence rather than raw counts;
- deriving a method-of-moments estimator for the parameters of this model, enabling lexicon-based classification with unique weights per word, without labeled data;
- empirically demonstrating that this classifier outperforms lexicon-based classification and alternative approaches.

Lexicon-Based Classification as Minimum Bayes Risk Classification

We begin by showing how the lexicon-based classification rule shown in (1) can be derived as a special case of minimum Bayes risk classification for binary classification problems. Let us suppose we have a prior probability P_Y for each possible label, and a likelihood function $P_{X|Y}$, where X is a random variable corresponding to a vector of word counts. The conditional label probability can be computed by Bayesian inversion,

$$P(y \mid \boldsymbol{x}) = \frac{P(\boldsymbol{x} \mid y)P(y)}{\sum_{y'} P(\boldsymbol{x} \mid y')P(y')}. \tag{2}$$

Assuming that the costs for each type of misclassification error are identical, then the minimum Bayes risk classification rule is,

$$\begin{aligned}&\log Pr(Y=0) + \log P(\boldsymbol{x} \mid Y=0)\\ &\quad \gtrless \log Pr(Y=1) + \log P(\boldsymbol{x} \mid Y=1),\end{aligned} \tag{3}$$

where we have moved to the log domain for simplicity of notation. We now show that lexicon-based classification can be justified under this decision rule, given a set of assumptions about the probability distributions.

First, we assume that the labels have **equal prior likelihood**, $Pr(Y=0) = Pr(Y=1)$. It is trivial to relax this assumption by adding a constant term to one side of the decision rule in (1).

Next, we introduce some assumptions about the likelihood function, $P_{X|Y}$. The random variable X is defined over vectors of counts, so a natural choice for the form of this likelihood is the multinomial distribution. For a specific vector of counts $X = \boldsymbol{x}$, we write $P(\boldsymbol{x} \mid y) \triangleq P_{\text{multinomial}}(\boldsymbol{x}; \boldsymbol{\theta}_y, N)$, where $\boldsymbol{\theta}_y$ is a probability vector associated with label y, and $N = \sum_i x_i$ is the total word count for $\boldsymbol{x}$. The multinomial likelihood is proportional to a product of likelihoods of categorical variables corresponding to individual words (tokens),

$$Pr(W=i \mid Y=y; \boldsymbol{\theta}) = \theta_{y,i}, \tag{4}$$

where the random variable W corresponds to a single token, whose probability of being word i is equal to $\theta_{y,i}$ in a document with label y. We can write the multinomial log-likelihood as,

$$\log P(\boldsymbol{x} \mid y) = \log P_{\text{multinomial}}(\boldsymbol{x}; \boldsymbol{\theta}_y, N) \tag{5}$$

$$= K(\boldsymbol{x}) + \sum_i x_i \log Pr(W=i \mid Y=y; \boldsymbol{\theta}) \tag{6}$$

$$= K(\boldsymbol{x}) + \sum_i x_i \log \theta_{y,i}, \tag{7}$$

where $K(\boldsymbol{x})$ is a function of $\boldsymbol{x}$ that is constant in y.

The first necessary assumption about the likelihood function is that the lexicons are **complete**: words that are in neither lexicon have identical probability under both labels. Formally, for any word $i \notin \mathcal{W}_0 \cup \mathcal{W}_1$, we assume,

$$Pr(W=i \mid Y=0) = Pr(W=i \mid Y=1). \tag{8}$$

These words are therefore irrelevant to the classification boundary.

Next, we assume that each in-lexicon word is **equally predictive**. Specifically, for words that are in lexicon y, we assume,

$$\frac{Pr(W=i \mid Y=y)}{Pr(W=i \mid Y=\neg y)} = \frac{1+\gamma}{1-\gamma}, \tag{9}$$

where $\neg y$ is the opposite label from y. The parameter γ controls the predictiveness of the lexicon: for example, if $\gamma = 0.5$ in a sentiment classification problem, this would indicate that words in the positive sentiment lexicon are three times more likely to appear in documents with positive sentiment than in documents with negative sentiment, and vice versa. The word *atrocious* might be less likely overall than *good*, but still three times more likely in the negative class than in the positive class. In the limit, $\gamma = 0$ implies that the lexicons do not distinguish the classes at all, and $\gamma = 1$ implies that the lexicons distinguish the classes perfectly, so that the observation of a single in-lexicon word would completely determine the document label.

The conditions enumerated in (8) and (9) are ensured by the following definition,

$$\theta_{y,i} = \begin{cases}(1+\gamma)\mu_i, & i \in \mathcal{W}_y\\ (1-\gamma)\mu_i, & i \in \mathcal{W}_{\neg y}\\ \mu_i, & i \notin \mathcal{W}_y \cup \mathcal{W}_{\neg y},\end{cases} \tag{10}$$

where $\neg y$ is the opposite label from y, and $\boldsymbol{\mu}$ is a vector of baseline probabilities, which are independent of the label.

Because the probability vectors $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_1$ must each sum to one, we require an assumption of **equal coverage**,

$$\sum_{i \in \mathcal{W}_0} \mu_i = \sum_{j \in \mathcal{W}_1} \mu_j. \tag{11}$$

With these assumptions in hand, it is now possible to simplify the decision rule in (3). Thanks to the assumption of equal prior probability, we can drop the priors $P(Y)$, so that the decision rule is a comparison of the likelihoods,

$$\log P(\boldsymbol{x} \mid Y=0) \gtrless \log P(\boldsymbol{x} \mid Y=1) \tag{12}$$

$$K(\boldsymbol{x}) + \sum_i x_i \log \theta_{0,i} \gtrless K(\boldsymbol{x}) + \sum_i x_i \log \theta_{1,i}. \tag{13}$$

Canceling $K(\boldsymbol{x})$ and applying the definition from (10),

$$\sum_{i \in \mathcal{W}_0} x_i \log((1+\gamma)\mu_i) + \sum_{i \in \mathcal{W}_1} x_i \log((1-\gamma)\mu_i) \quad (14)$$

$$\gtrless \sum_{i \in \mathcal{W}_0} x_i \log((1-\gamma)\mu_i) + \sum_{i \in \mathcal{W}_1} x_i \log((1+\gamma)\mu_i). \quad (15)$$

The μ_i terms cancel after distributing the log, leaving,

$$\sum_{i \in \mathcal{W}_0} x_i \log \frac{1+\gamma}{1-\gamma} \gtrless \sum_{i \in \mathcal{W}_1} x_i \log \frac{1+\gamma}{1-\gamma}. \quad (16)$$

For any $\gamma \in (0,1)$, the term $\log \frac{1+\gamma}{1-\gamma}$ is a finite and positive constant. Therefore, (16) is identical to the counting-based classification rule in (1). In other words, lexicon-based classification is minimum Bayes risk classification in a multinomial probability model, under the assumptions of equal prior likelihood, lexicon completeness, equal predictiveness of words, and equal coverage.

Analysis of Lexicon-Based Classification

One advantage of deriving a formal foundation for lexicon-based classification is that it is possible to analyze its expected performance. For a label y, let us write the count of in-lexicon words as $m_y = \sum_{i \in \mathcal{W}y} x_i$, and the count of opposite-lexicon words as $m_{\neg y} = \sum_{i \in \mathcal{W}\neg y} x_i$. Lexicon-based classification makes a correct prediction whenever $m_y > m_{\neg y}$ for the correct label y. To analyze lexicon-based classification, we assess the likelihood that $m_y > m_{\neg y}$ under various parametrizations and conditions. Specifically, we compute the expectation and variance of the difference $m_y - m_{\neg y}$; under the central limit theorem, we can treat this difference as approximately normally distributed, and compute the probability that the difference is positive using the Gaussian cumulative distribution function (CDF).

We introduce the convenience notation $s_\mu \triangleq \sum_{i \in \mathcal{W}_0} \mu_i = \sum_{i \in \mathcal{W}_1} \mu_i$; recall that we have already taken the assumption that the sums of baseline word probabilities for the two lexicons are equal. Under the multinomial probability model, given a document with N tokens, the expected counts are,

$$E[m_y] = N \sum_{i \in \mathcal{W}y} \theta_{y,i} = N(1+\gamma)s_\mu \quad (17)$$

$$E[m_{\neg y}] = N \sum_{i \in \mathcal{W}\neg y} \theta_{\neg y,i} = N(1-\gamma)s_\mu \quad (18)$$

$$E[m_y - m_{\neg y}] = 2N\gamma s_\mu. \quad (19)$$

Next we compute the variance of this margin,

$$V[m_y - m_{\neg y}] = V[m_y] + V[m_{\neg y}] + Cov(m_y, m_{\neg y}). \quad (20)$$

Each of these terms is the variance of a sum of counts. Under the multinomial distribution, the variance of a single count

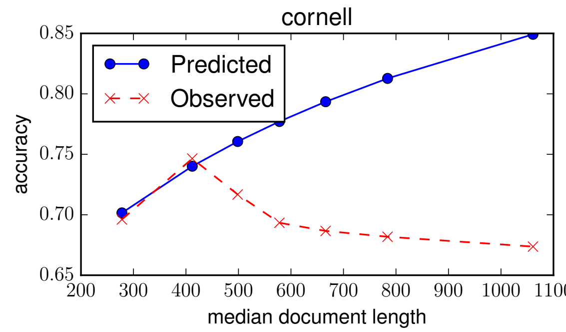

Figure 1: Accuracy of lexicon-based classification with document length on the Cornell review data, using the Liu sentiment lexicon. Documents are binned by length, and points are plotted at the median of each bin. The dataset and lexicon are described below.

is $V[x_i] = N\theta_i(1-\theta_i)$. The variance of the sum m_y is,

$$V[m_y] = \sum_{i \in \mathcal{W}y} N\theta_i(1-\theta_i) - \sum_{j \in \mathcal{W}y, j \neq i} N\theta_i\theta_j \quad (21)$$

$$= \sum_{i \in \mathcal{W}y} N\theta_i - N\theta_i^2 - \sum_{j \in \mathcal{W}y, j \neq i} N\theta_i\theta_j \quad (22)$$

$$\leq N \sum_{i \in \mathcal{W}y} \theta_i = N \sum_{i \in \mathcal{W}y} (1+\gamma)\mu_i \quad (23)$$

$$= N(1+\gamma)s_\mu. \quad (24)$$

An equivalent upper bound can be computed for the variance of the count of opposite lexicon words, $V[m_{\neg y}] \leq N(1-\gamma)s_\mu$. These bounds are fairly tight because the products of probabilities θ_i^2 and $\theta_i\theta_j$ are nearly always small, due to the fact that most words are rare. Because the covariance $Cov(m_y, m_{\neg y})$ is negative (and also involves a product of word probabilities), we can further bound the variance of the margin, obtaining the upper bound,

$$V[m_y - m_{\neg y}] \leq N(1+\gamma)s_\mu + N(1-\gamma)s_\mu = 2Ns_\mu\,. \quad (25)$$

By the central limit theorem, the margin $m_y - m_{\neg y}$ will be approximately distributed, with mean $2N\gamma s_\mu$ and variance $2Ns_\mu$. The probability of making a correct prediction is then equal to the cumulative density of a standard normal distribution $\Phi(z)$, where the z-score is equal to the ratio of the expectation and the standard deviation,

$$z = \frac{E[m_y - m_{\neg y}]}{\sqrt{V[m_y - m_{\neg y}]}} = \frac{2N\gamma s_\mu}{\sqrt{2Ns_\mu}}. = \gamma\sqrt{2Ns_\mu}. \quad (26)$$

Note that by upper-bounding the variance, we obtain a lower bound on the z-score, and thus a lower bound on the expected accuracy.

According to this approximation, we expect accuracy to increase with the predictiveness γ, the document length N, and the lexicon coverage s_μ. This helps to explain a dilemma in lexicon design: as more words are added, the coverage increases, but the average predictiveness of each word decreases (assuming the most predictive words are added first). Thus, increasing the size of a lexicon by adding marginal words may not improve performance.

The analysis also indicates that longer documents should be easier to predict. This is because the expectation of the gap $m_y - m_{\neg y}$ grows with N, while its standard deviation grows only with $\sqrt{N}$. This can be tested empirically; we focus on the Cornell dataset (described below), measuring how accuracy varies with document length. For this test, we estimate γ by maximum likelihood estimation on labeled data. The expected accuracy is indicated by the blue solid lines in Figure 1, in which documents are grouped into seven equal size groups by document length. As shown by the red dotted lines, these estimates are qualitatively wrong: the performance of lexicon-based classification does not increase monotonically with document length as predicted by the model. The decreased accuracy for especially long reviews may be due to these reviews being more complex, perhaps requiring modeling of the discourse structure (Somasundaran, Wiebe, and Ruppenhofer 2008).

Justifying the Word-Appearance Heuristic

An alternative heuristic to lexicon-based classification is to consider only the **presence** of each word type, and not its count. This corresponds to the decision rule,

$$\sum_{i\,\in\mathcal{W}_0} \delta(x_i > 0) \gtrless \sum_{j\,\in\mathcal{W}_1} \delta(x_j > 0), \tag{27}$$

where $\delta(\cdot)$ is a delta function that returns one if the Boolean condition is true, and zero otherwise. In the context of supervised classification, Pang, Lee, and Vaithyanathan (2002) find that word presence is a more predictive feature than word frequency. By ignoring repeated mentions of the same word, this heuristic (27) emphasizes the diversity of ways in which a document covers a lexicon, and is more robust to document-specific idiosyncrasies — such as a review of *The Joy Luck Club*, which might include the positive words *joy* and *luck* many times even if the review is negative.

This heuristic can also be explained in the framework defined above. Rather than using a multinomial for the likelihood $P_{X|Y}$, we use a **Dirichlet-compound multinomial** (DCM), also known as a multivariate Polya distribution (Madsen, Kauchak, and Elkan 2005). This distribution is written $P_{\text{dcm}}(\boldsymbol{x};\boldsymbol{\alpha}_y)$, where $\boldsymbol{\alpha}_y$ is a vector of parameters associated with label y, with $\alpha_{y,i} > 0$ for all i. The DCM is a "compound" distribution because it treats the parameter of the multinomial as a latent variable to be marginalized out,

$$P_{\text{dcm}}(\boldsymbol{x};\boldsymbol{\alpha}_y) = \int_{\boldsymbol{\nu}} P_{\text{multinomial}}(\boldsymbol{x} \mid \boldsymbol{\nu}) P_{\text{Dirichlet}}(\boldsymbol{\nu} \mid \boldsymbol{\alpha}_y) d\boldsymbol{\nu}. \tag{28}$$

Intuitively, we can think of the DCM distribution as encoding a model in which each document has its own multinomial distribution over words; this document-specific distribution is itself drawn from a prior that depends on the class label y.

Suppose we set the DCM parameter $\boldsymbol{\alpha} = \tau\boldsymbol{\theta}$, with $\boldsymbol{\theta}$ as defined in (10). The constant $\tau > 0$ is then the **concentration** of the distribution. Because $\sum_i \theta_i = 1$, the likelihood function under this model is,

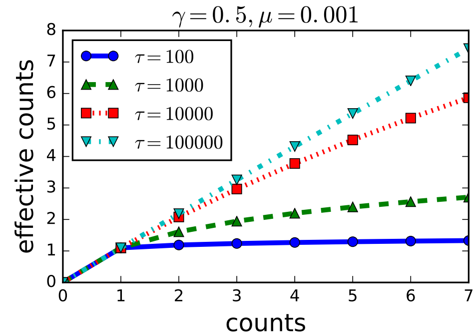

Figure 2: Effective counts for varying values of τ. For the datasets considered in this paper, τ usually falls in the range between 500 and 1000.

$$P_{\text{dcm}}(\boldsymbol{x} \mid y) = \frac{\Gamma(\tau)}{\Gamma(N+\tau)} \prod_i \frac{\Gamma(x_i + \tau\theta_{y,i})}{\Gamma(\tau\theta_{y,i})}, \tag{29}$$

where $\Gamma(\cdot)$ is the gamma function. Minimum Bayes risk classification in this model implies the decision rule:

$$\sum_{i\in\mathcal{W}_0} \log \frac{r_{\text{in}}(x_i)}{r_{\text{out}}(x_i)} \gtrless \sum_{i\in\mathcal{W}_1} \log \frac{r_{\text{in}}(x_{t,i})}{r_{\text{out}}(x_i)} \tag{30}$$

where,

$$r_{\text{in}}(x_i) \triangleq \frac{\Gamma(x_i + \tau(1+\gamma)\mu_i)}{\Gamma(\tau(1+\gamma)\mu_i)} \tag{31}$$

$$r_{\text{out}}(x_i) \triangleq \frac{\Gamma(x_i + \tau(1-\gamma)\mu_i)}{\Gamma(\tau(1-\gamma)\mu_i)}. \tag{32}$$

As $\tau \to \infty$, the prior on $\boldsymbol{\nu}$ is tightly linked to $\boldsymbol{\theta}$, so that the model reduces to the multinomial defined above. Another way to see this is to apply the equality $\Gamma(x+1) = x\Gamma(x)$ to (31) and (32) when $\tau\mu_i \gg x_i$. As $\tau \to 0$, the prior on $\boldsymbol{\nu}$ becomes increasingly diffuse. Repeated counts of any word are better explained by document-specific variation from the prior, than by properties of the label. This situation is shown in Figure 2, which plots the "effective counts" implied by the classification rule (30) for a range of values of the concentration parameter τ, holding the other parameters constant ($\mu = 10^{-3}, \gamma = 0.5$). For high values of τ, the effective counts track the observed counts linearly; for low values of τ, the effective counts barely increase beyond 1.

Minka (2012) presents a number of estimators for the concentration parameter τ from a corpus of text. When the label y is unknown, we cannot apply these estimators directly. However, as described above, we have taken the assumption that out-of-lexicon words have identical probability under both labels. We can use this assumption to estimate τ exclusively from the first and second moments of these out-of-lexicon words. Analysis of the expected accuracy of this model is left to future work.

Estimating Word Predictiveness

A crucial simplification made by lexicon-based classification is that all words in each lexicon are equally predictive.

In reality, words may be more or less predictive of class labels, for reasons such as sense ambiguity (e.g., *well*) and degree (e.g., *good* vs *flawless*). By introducing a per-word predictiveness factor γ_i into (10), we arrive at a model that is a restricted form of Naïve Bayes. (the restriction is that the probabilities of non-lexicon words are constrained to be identical across classes.) If labeled data were available, this model could be estimated by maximum likelihood. This section shows how to estimate the model without labeled data, using the method of moments.

First, note that the baseline probabilities μ_i can be estimated directly from counts on an unlabeled corpus; the challenge is to estimate the parameters γ_i for all words in the two lexicons. The key intuition that makes this possible is that highly predictive words should rarely appear with words in the opposite lexicon. This idea can be formalized in terms of **cross-label counts**: the cross-label count c_i is the co-occurrence count of word i with all words in the opposite lexicon,

$$c_i = \sum_{t=1}^{T} \sum_{j \in \mathcal{W}_{\neg y}} x_i^{(t)} x_j^{(t)}, \tag{33}$$

where $\boldsymbol{x}^{(t)}$ is the vector of word counts for document t, with $t \in \{1 \ldots T\}$. Under the multinomial model defined above, for a single document with N tokens, the expected product of counts for a word pair (i, j) is equal to,

$$\begin{aligned} E[x_i x_j] =& E[x_i]E[x_j] + Cov(x_i, x_j) && (34)\\ =& N\theta_i N\theta_j - N\theta_i\theta_j && (35)\\ =& N(N-1)\theta_i\theta_j. && (36) \end{aligned}$$

We will focus on the expected products of counts for cross-lexicon word pairs ($i \in \mathcal{W}_0, j \in \mathcal{W}_1$). The parameter $\boldsymbol{\theta}$ depends on the document label y, as defined in (10). As a result, we have the following expectations,

$$\begin{aligned} E[x_i x_j \mid Y=0] =& N(N-1)\mu_i(1+\gamma_i)\mu_j(1-\gamma_j) && (37)\\ =& N(N-1)\mu_i\mu_j(1+\gamma_i-\gamma_j-\gamma_i\gamma_j) && (38)\\ E[x_i x_j \mid Y=1] =& N(N-1)\mu_i(1-\gamma_i)\mu_j(1+\gamma_j) && (39)\\ =& N(N-1)\mu_i\mu_j(1-\gamma_i+\gamma_j-\gamma_i\gamma_j) \\ E[x_i x_j] =& P(Y=0)E[x_i x_j \mid Y=0] && (40)\\ &+ P(Y=1)E[x_i x_j \mid Y=1] && (41)\\ =& N(N-1)\mu_i\mu_j(1-\gamma_i\gamma_j). && (42) \end{aligned}$$

Summing over all words in $j \in \mathcal{W}_1$ and documents t,

$$\begin{aligned} E[c_i] =& \sum_{t=1}^{T} \sum_{j \in \mathcal{W}_1} E[x_i^{(t)} x_j^{(t)}] && (43)\\ =& \left(\sum_{t=1}^{T} N_t(N_t-1)\right)\left(\mu_i \sum_{j \in \mathcal{W}_1} \mu_j(1-\gamma_i\gamma_j)\right) && (44) \end{aligned}$$

Let us write $\boldsymbol{\gamma}^{(1)}$ to indicate the vector of γ_j parameters for all $j \in \mathcal{W}_1$, and $\boldsymbol{\gamma}^{(0)}$ for all $i \in \mathcal{W}_0$. The expectation in (44) is a linear function of γ_i, and a linear function of the vector $\boldsymbol{\gamma}^{(1)}$. Analogously, for all $j \in \mathcal{W}_1$, $E[c_j]$ is a linear function of γ_j and $\boldsymbol{\gamma}^{(1)}$.

Our goal is to choose $\boldsymbol{\gamma}$ so that the expectations $E[c_i]$ closely match the observed counts c_i. This can be viewed as form of **method of moments** estimation. We formulate the following objective,

$$J = \frac{1}{2}\sum_{i \in \mathcal{W}_0}(c_i - E[c_i])^2 + \frac{1}{2}\sum_{j \in \mathcal{W}_1}(c_j - E[c_j])^2, \tag{45}$$

which can be minimized in terms of $\boldsymbol{\gamma}^{(0)}$ and $\boldsymbol{\gamma}^{(1)}$. However, there is an additional constraint: the probability distributions $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_1$ must still sum to one. We can express this as a linear constraint on $\boldsymbol{\gamma}^{(0)}$ and $\boldsymbol{\gamma}^{(1)}$,

$$\boldsymbol{\mu}^{(0)} \cdot \boldsymbol{\gamma}^{(0)} - \boldsymbol{\mu}^{(1)} \cdot \boldsymbol{\gamma}^{(1)} = 0, \tag{46}$$

where $\boldsymbol{\mu}^{(y)}$ is the vector of baseline probabilities for words $i \in \mathcal{W}_y$, and $\boldsymbol{\mu}^{(0)} \cdot \boldsymbol{\gamma}^{(0)}$ indicates a dot product.

We can now formulate the following constrained optimization problem,

$$\begin{aligned} \min_{\boldsymbol{\gamma}^{(0)}, \boldsymbol{\gamma}^{(1)}} \; & \frac{1}{2}\sum_{i \in \mathcal{W}_0}(c_i - E[c_i])^2 + \frac{1}{2}\sum_{j \in \mathcal{W}_1}(c_j - E[c_j])^2 \\ s.t. \; & \boldsymbol{\mu}^{(0)} \cdot \boldsymbol{\gamma}^{(0)} - \boldsymbol{\mu}^{(1)} \cdot \boldsymbol{\gamma}^{(1)} = 0. && (47) \end{aligned}$$

This problem can be solved by **alternating direction method of multipliers** (Boyd et al. 2011). The objective is biconvex in $\boldsymbol{\gamma}^{(0)}$ and $\boldsymbol{\gamma}^{(1)}$, which suggests an iterative solution. Specifically, we solve for $\boldsymbol{\gamma}^{(0)}$ while holding fixed $\boldsymbol{\gamma}^{(1)}$ and an additional term $\boldsymbol{u}$, representing a penalty for violating the constraint. This is a convex optimization problem, which can be solved using standard unconstrained L-BFGS (Liu and Nocedal 1989). Next we solve for $\boldsymbol{\gamma}^{(1)}$, again by L-BFGS. Finally, we update the constraint term $\boldsymbol{u}$ in closed form (see section 9.2 of Boyd et al. 2011 for details). We iterate this procedure until the primal and dual residuals are below a small threshold.

Evaluation

To evaluate the effectiveness of the proposed method, we perform an empirical evaluation on four datasets in two languages. All datasets involve binary classification problems, and we evaluate using **a**rea-**u**nder-the-**c**urve (AUC), a measure of classification performance that is robust to unbalanced class distributions. A perfect classifier achieves AUC $= 1$; in expectation, a random decision rule gives AUC $= 0.5$.

Datasets We consider the following datasets:

Amazon English-language product reviews across four domains; of these reviews, 8000 are labeled and another 19677 are unlabeled (Blitzer, Dredze, and Pereira 2007).

Cornell 2000 English-language film reviews, labeled as positive or negative (Pang and Lee 2004).

CorpusCine 3800 Spanish-language movie reviews, rated on a scale of one to five (Vilares, Alonson, and Gómez-Rodríguez 2015). We use the 2606 reviews with ratings of four or above (positive) and two or below (negative).

IMDB 50,000 English-language film reviews (Maas et al. 2011). We use the test set of 25,000 reviews, of which half are positive and half are negative.

Lexicons Preliminary evaluation compared several English-language sentiment lexicons. The Liu lexicon (Liu 2015) consistently obtained the best performance on all three English-language datasets, so it was made the focus of all subsequent experiments. Our observations match previous research, which has also found that the Liu lexicon is one of the strongest lexicons for review analysis (Ribeiro et al. 2016). For Spanish, we use ISOL, which is a modified translation of the Liu lexicon (Molina-González et al. 2013).

Classifiers The evaluation compares the following unsupervised classification strategies:

LEXICON basic word counting, as in decision rule (1);

LEXICON-PRES counting word presence rather than frequency, as in decision rule (27);

LEXIMOM lexicon-based classification with word predictiveness γ_i estimated by the **M**ethod-**O**f-**M**oments technique described in the previous section, by solving the optimization problem in (47);

LEXIMOM-BAYES Same as LEXIMOM, but with the Bayesian decision rule shown in (30).

PMI An alternative approach, discussed in the related work, is to impute document labels from a seed set of words, and then compute "sentiment scores" for individual words from pointwise mutual information between the words and imputed labels (Turney 2002). Our implementation of this method is based on the description from Kiritchenko, Zhu, and Mohammad (2014), using the lexicons as the seed word sets.

As an upper bound on classification performance, we train a supervised logistic regression classifier, using 5-fold cross-validation. This is the only classifier in the evaluation with access to labeled data, so it can be considered an upper bound on the possible performance for this task.

Results Results are shown in Table 1. The superior performance of the logistic regression classifier confirms the principle that supervised classification is far more accurate than lexicon-based classification — when labeled data is available. Nonetheless, the unsupervised method-of-moments estimator developed in this paper (LEXIMOM) goes a considerable way towards closing the gap, with improvements in AUC ranging from 1.3% on the CorpusCine data to 7.6% on the IMDB data. The results with accuracy are very similar, with consistent improvements on all four datasets, ranging from 1.8% on CorpusCine to 7.0% on IMDB. Overall, these results offer strong evidence on behalf of the method-of-moments estimator for word predictiveness. The PMI approach performs poorly, improving over the simpler lexicon-based classifiers on only one of the four datasets. The word presence heuristic offers no consistent improvements on this data, and the Bayesian adjustment to the classification rule offers only small improvements on two of the four datasets.

	Amazon	Cornell	Cine	IMDB
LEXICON	.820	.765	.636	.807
LEXICON-PRES	.820	.770	.638	.805
PMI	.793	.761	.638	.868
LEXIMOM	.847	.822	**.651**	**.884**
LEXIMOM-BAYES	**.852**	**.831**	**.651**	.883
LOGREG	.897	.914	.889	.955

Table 1: Area-under-the-curve (AUC) for all classifiers. The best unsupervised result is shown in bold for each dataset.

Related work

Turney (2002) uses pointwise mutual information to estimate the "semantic orientation" of all vocabulary words from co-occurrence with a small seed set. This approach has later been extended to the social media domain by using emoticons as the seed set (Kiritchenko, Zhu, and Mohammad 2014). Like our approach, the basic intuition is to leverage co-occurrence statistics to learn weights for individual words; unlike our approach, the PMI is a heuristic score that is not justified by a probabilistic model of the text classification problem. PMI-based classification underperforms our proposed approach on all four datasets in our evaluation.

The method-of-moments has become an increasingly popular estimator in unsupervised machine learning, with applications in topic models (Anandkumar et al. 2014), sequence models (Hsu, Kakade, and Zhang 2012), and more elaborate linguistic structures (Cohen et al. 2014). Of particular relevance are "anchor word" techniques for learning latent topic models (Arora, Ge, and Moitra 2012). In these methods, each topic is defined first by a few keywords, which are assumed to be generated only from a single topic. From these anchor words and co-occurrence statistics, the topic-word probabilities can be recovered. A key difference is that we do not take the strong anchor word assumption in this work: none of the words are assumed to be perfectly predictive of either label. We take the much weaker assumption that words in a lexicon tend to co-occur less frequently with words in the opposite lexicon.

Conclusion

Lexicon-based classification is a popular heuristic that has not previously been analyzed from a machine learning perspective. This analysis yields two techniques for improving unsupervised binary classification: a method-of-moments estimator for word predictiveness, and a Bayesian adjustment for repeated counts of the same word. The method-of-moments estimator yields substantially better performance than conventional lexicon-based classification, without requiring any additional annotation effort. Future work will consider the generalization to multi-class classification, and more ambitiously, and the extension to multiword units.

Acknowledgment This research was supported by the National Institutes of Health under award number R01GM112697-01, and by the Air Force Office of Scientific Research.

References

Anandkumar, A.; Ge, R.; Hsu, D.; Kakade, S. M.; and Telgarsky, M. 2014. Tensor decompositions for learning latent variable models. *The Journal of Machine Learning Research* 15(1):2773–2832.

Arora, S.; Ge, R.; and Moitra, A. 2012. Learning topic models - going beyond SVD. In *FOCS*, 1–10.

Bhatia, P.; Ji, Y.; and Eisenstein, J. 2015. Better document-level sentiment analysis from rst discourse parsing. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*.

Blitzer, J.; Dredze, M.; and Pereira, F. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the Association for Computational Linguistics (ACL)*, 440–447.

Boyd, S.; Parikh, N.; Chu, E.; Peleato, B.; and Eckstein, J. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning* 3(1):1–122.

Cohen, S. B.; Stratos, K.; Collins, M.; Foster, D. P.; and Ungar, L. 2014. Spectral learning of latent-variable PCFGs: Algorithms and sample complexity. *Journal of Machine Learning Research* 15:2399–2449.

Hatzivassiloglou, V., and McKeown, K. R. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of the Association for Computational Linguistics (ACL)*, 174–181.

Hsu, D.; Kakade, S. M.; and Zhang, T. 2012. A spectral algorithm for learning hidden markov models. *Journal of Computer and System Sciences* 78(5):1460–1480.

Kiritchenko, S.; Zhu, X.; and Mohammad, S. M. 2014. Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research* 50:723–762.

Laver, M., and Garry, J. 2000. Estimating policy positions from political texts. *American Journal of Political Science* 619–634.

Liu, D. C., and Nocedal, J. 1989. On the limited memory bfgs method for large scale optimization. *Mathematical programming* 45(1-3):503–528.

Liu, B. 2015. *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*. Cambridge University Press.

Maas, A. L.; Daly, R. E.; Pham, P. T.; Huang, D.; Ng, A. Y.; and Potts, C. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the Association for Computational Linguistics (ACL)*.

Madsen, R. E.; Kauchak, D.; and Elkan, C. 2005. Modeling word burstiness using the dirichlet distribution. In *Proceedings of the 22nd international conference on Machine learning*, 545–552. ACM.

Minka, T. 2012. Estimating a dirichlet distribution. `http://research.microsoft.com/en-us/um/people/minka/papers/dirichlet/minka-dirichlet.pdf`.

Molina-González, M. D.; Martínez-Cámara, E.; Martín-Valdivia, M.-T.; and Perea-Ortega, J. M. 2013. Semantic orientation for polarity classification in spanish reviews. *Expert Systems with Applications* 40(18):7250–7257.

Pang, B., and Lee, L. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the Association for Computational Linguistics (ACL)*, 271–278.

Pang, B., and Lee, L. 2008. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval* 2(1-2):1–135.

Pang, B.; Lee, L.; and Vaithyanathan, S. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*, 79–86.

Polanyi, L., and Zaenen, A. 2006. Contextual valence shifters. In *Computing attitude and affect in text: Theory and applications*. Springer.

Qiu, G.; Liu, B.; Bu, J.; and Chen, C. 2011. Opinion word expansion and target extraction through double propagation. *Computational linguistics* 37(1):9–27.

Ribeiro, F. N.; Araújo, M.; Gonçalves, P.; Gonçalves, M. A.; and Benevenuto, F. 2016. Sentibench-a benchmark comparison of state-of-the-practice sentiment analysis methods. *EPJ Data Science* 5(1):1–29.

Settles, B. 2011. Closing the loop: Fast, interactive semi-supervised annotation with queries on features and instances. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 1467–1478. Association for Computational Linguistics.

Somasundaran, S.; Wiebe, J.; and Ruppenhofer, J. 2008. Discourse level opinion interpretation. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, 801–808. Association for Computational Linguistics.

Taboada, M.; Brooke, J.; Tofiloski, M.; Voll, K.; and Stede, M. 2011. Lexicon-based methods for sentiment analysis. *Computational linguistics* 37(2):267–307.

Tausczik, Y. R., and Pennebaker, J. W. 2010. The psychological meaning of words: LIWC and computerized text analysis methods. *Journal of Language and Social Psychology* 29(1):24–54.

Turney, P. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the Association for Computational Linguistics (ACL)*, 417–424.

Vilares, D.; Alonson, M. A.; and Gómez-Rodríguez, C. 2015. A syntactic approach for opinion mining on spanish reviews. *Natural Language Engineering* 21:139–163.

Wilson, T.; Wiebe, J.; and Hoffmann, P. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*, 347–354.