Machine Learning, Information Retrieval, and Record Linkage

William E. Winkler, bwinkler@census.gov, U.S. Bureau of the Census
Statistical Research, Room 3000-4, Washington DC 20233-9100

Keywords: Data Mining, Bayesian Networks, EM Algorithm, Optimization

ABSTRACT

Classification into groups using terms available in the data underlies machine learning, information retrieval, and record linkage. Classifiers such as Bayesian networks in machine learning and term weighting in information retrieval depend primarily on training data sets for which truth is known. These classifiers may be relatively slow to adapt to new situations in which new data have characteristics significantly different from the training data. Record linkage has been characterized by data in which training data can differ significantly from new data being classified. By using structuring ideas introduced by Fellegi and Sunter in their classic 1969 JASA paper, record linkage researchers have been able to apply the EM algorithm and Markov Chain Monte Carlo ideas to make classifiers automatically adapt to new data. We show how these ideas can be used to improve the learnability of Bayesian networks. We also use some of the ideas from machine learning that are superficially related to boosting to show how non-naïve Bayesian classifiers can make better use of training data.

1. INTRODUCTION

This paper covers classification of data into groups using textual information. In machine learning, text classification is used for classifying documents into categories. For newspaper articles, the categories might be different subject classes such as crude oil, acquisitions of companies, and international trade. For industry and occupation coding, the categories might be the first two or three digits of the North American Industrial Code. For disease and treatment, the categories might be polio, meningitis, or common flu and the text information might be extended to numerical categories corresponding to disease symptoms and responses to certain tests. In information retrieval, the text might be queries related to subjects that are used for a library search or an internet search. In record linkage, the categories might simply be the determination that a pair of records from two lists represents the same entity (is a match) or is not the same entity (non-match). Although some of the methods in machine learning such as nearest neighbor matching and neural nets originated with numeric data, this paper only considers nonnumeric data.

In machine learning and information retrieval, general text information is used for classifying. The information might be the title and first few paragraphs from a document or a set of query words. Record linkage typically has more structured information. Name and address parsing and standardization software puts person names and addresses into specific locations. First names can then be compared with first names, house numbers with house numbers, and street names with street names. Business names and person names cannot always be parsed into components that can be effectively compared. For instance, it is difficult to compare 'J K Smith and Co' with 'JKS Inc' without special rules and the help of corresponding addresses and other information.

Machine learning and information retrieval typically use training data for which the true classification status is known. The training data are used to develop a vocabulary that is used for comparing documents. Commonly occurring words such as 'the' that appear in most documents are removed via stoplists. Stemming may be to different words such as 'acquire', 'acquires', 'acquired', and 'acquiring' into one class 'acquire' making it easier to compare and use words. The vocabulary creates the structure of words. The words in the vocabulary typically occur in individual classes. The vocabulary is used to separate each class from other classes. Record linkage uses simpler stemming in which variants of words such as 'road', 'drive', 'p.o. box and 'doctor' are given common spellings. Because of the additional structure of knowing what words to compare, record linkage has not always needed training data. Guesses of some record linkage parameters can sometimes yield suitable decision rules.

Various methods in machine learning such as Bayesian networks make use of the relative frequencies of the occurrence of words in documents. The term weighting model of Salton (e.g. Salton and McGill 1983, Salton 1991) in information retrieval makes explicit use of the relative frequencies and has enhancements that allow the updating of the frequencies as additional data is processed. The original record linkage models of Newcombe (1959) made explicit uses of the relative frequencies of commonly occurring words such as 'Smith' and infrequently occurring words such as 'Zabrinsky.'

Fellegi and Sunter (1969) provided a formal mathematical model for record linkage and gave general proofs of the optimality of the methods. The record linkage theory and decision rules correspond precisely to Bayesian networks in machine learning (Mitchell 1997, Nigam, McCallum, Thrun, and Mitchell 2000). In practice, it is difficult to estimate accurately the probabilities associated with the decision rules in record

linkage and Bayesian networks. Fellegi and Sunter (1969) gave a method for automatically estimating record linkage parameters without training data. For the automatic estimation method to be valid, certain implicit assumptions about the structure of the data are required. The assumptions deal with the homogeneity of relationships of terms within classes and the mutual dependencies of terms.

To make computational more tractable, Fellegi and Sunter made a conditional independence assumption that corresponds to the naïve Bayes assumption for Bayesian networks. The assumption (made explicit later in this paper) is that the presence of one word in a class is independent on another word in a class. The assumption is that P(crude,oil | C_1) = P(crude | C_1) P(oil | C_1). In record linkage the analogous assumption is that P(agree on last name Smith, agree on first name Robert | C_1) = P(agree on last name Smith | C_1) P(agree on first name Robert | C_1). Winkler (1988) and Nigam et al. (2000) have shown how to use the EM algorithm (Dempster, Laird, and Rubin 1977) to obtain parameters (probabilities) for the classification decision rules in record linkage and Bayesian networks, respectively. Winkler (1989a, 1993) extended the EM procedures to situations where conditional independence does not hold and where convex constraints could be imposed on the parameters based on prior knowledge. For some applications, optimal record linkage parameters can be obtained automatically without training data. For a major Census application, the U.S. is divided into 550 areas and matched within three weeks (Winkler and Thibaudeau 1991). The optimal parameters vary substantially across areas.

In the applications of Nigam et al. (2000), suitable training data are necessary to create structure for the EM algorithm. Nigam et al. combine labeled training data with unlabeled additional data. They show that, if moderate amounts of training data are combined with the proper amounts of additional unlabelled data, classification decision rules are improved. They show that, if a small amount of training data is combined with a large amount unlabelled data, then decision rules will likely be poor. The intuition might be that a certain number of representative words are needed to represent classes. To get a sufficient number of words and documents to represent a class, a sufficiently large training sample is needed. If only unlabelled documents are used (referred to as *unsupervised learning*, Mitchell 1997), then the resultant classes may be very unlike the classifications that are ultimately needed. For exploratory purposes, unsupervised learning is used in data mining applications.

The formal mathematical model of Bayesian networks (Nigam et al. 2000) and the original model of Fellegi and Sunter (1969) are appealing because accurately estimated probabilities give good estimates of the error rates. As shown by Fellegi and Sunter, optimal decision rules can be obtained under a large range of error rates. In practice, because the underlying true probabilities have not been accurately estimated, estimated error rates are not accurate (Belin and Rubin 1995, Nigam et al. 2000). Classification decision rules, however, may still be good. Belin and Rubin (1995) gave an alternate EM-based for estimating error rates in a narrow range of situations. Winkler (1993, 1994) has shown the general EM procedures that account for dependencies can yield accurate estimates of error rates and highly accurate decision rules in some situations.

Although the original work of Nigam et al. (2000) appeared very promising, recent work by Yang and Liu (1999) has shown that new variants of other methods in machine learning work consistently better than Bayesian networks with a variety of representative test decks. Yang and Liu indicated that the comparisons in Nigam et al. were not entirely appropriate because they did not use entire sets of test data. Rather, Nigam et al. only used the largest classes in test decks such as the Reuters collection (e.g., Lewis and Ringuette 1994). Yang and Liu showed that methods such as Support Vector Machines (SVM), k-nearest neighbor (kNN), and Linear Least Squares Fit (LLSF) all worked better according to a variety of statistical measures on several test decks. Whereas Bayesian networks and Neural nets performed more poorly than SVM, kNN, and LLSF, Yang (1999) had shown that they typically perform better than other competing classifiers in the machine learning literature.

It is the intent of this paper to demonstrate how Bayesian network classification can be improved. The comparisons are between variants of Bayesian networks that account for dependencies and naïve Bayesian networks (for which conditional independence is assumed). No specific comparison to SVM, kNN, and LLSF will be done. Set-covering algorithms are used to obtain parsimonious vocabularies that cover most documents in each class. This improves understandings of the specific words that are needed for classification. With a parsimonious vocabulary, set-covering algorithms are again used to obtain small candidate sets of words within each class for which interactions are modeled. The smaller sets of words and small sets of interactions make it easier to estimate accurate probabilities and error rates.

The outline of the papers is as follows. Following the introduction, the second section gives background on some of the models of machine learning, some of the models in information retrieval, and the Fellegi-Sunter model of record linkage. Most of the background is on Bayesian networks and record linkage. The other models in machine learning are used to develop intuition. In particular, SVM and kNN implicitly make use of dependencies which may account for some of the better performance observed by Yang and Liu (1999). Information retrieval is only covered in terms of its

relationship to machine learning and record linkage. In the third section, the theory associated with the specific parameter estimation methods for non-naïve Bayesian networks is presented. The theory is a special case of methods in the record linkage model of Fellegi and Sunter (Winkler 1993). The fourth section describes the Reuters test deck and the set-covering algorithms that are used to obtain a vocabulary (e.g., Lewis 1992). Yang and Pederson (1997) have extensively described methods for obtaining vocabularies. In the fifth section, results comparing the ten classes used in Nigam et al (2000) are presented. The sixth section gives discussion and the final section is concluding remarks.

2. BACKGROUND

The background is divided into description of the Fellegi-Sunter model of record and shorter descriptions of information retrieval and machine learning.

2.1. Fellegi-Sunter Model of Record Linkage

Fellegi and Sunter (1969) provided a formal mathematical model for ideas that had been introduced by Newcombe (1959, 1962, see also 1988). They provided many ways of estimating key parameters. To begin, notation is needed. Two files **A** and **B** are matched. The idea is to classify pairs in a product space $\mathbf{A} \times \mathbf{B}$ from two files A and B into M, the set of true matches, and U, the set of true nonmatches. Fellegi and Sunter, making rigorous concepts introduced by Newcombe (1959), considered ratios of probabilities of the form:

$$R = P(\gamma \in \Gamma \mid M) / P(\gamma \in \Gamma \mid U) \qquad (1)$$

where γ is an arbitrary agreement pattern in a comparison space Γ. For instance, Γ might consist of eight patterns representing simple agreement or not on the largest name component, street name, and street number. Alternatively, each $\gamma \in \Gamma$ might additionally account for the relative frequency with which specific values of name components such as "Smith", "Zabrinsky", "AAA", and "Capitol" occur. The ratio R or any monotonely increasing function of it such as the natural log is referred to as a matching weight (or score).

The decision rule is given by:

If $R > T_\mu$, then designate pair as a match.

If $T_\lambda \le R \le T_\mu$, then designate pair as a possible match and hold for clerical review. (2)

If $R < T_\lambda$, then designate pair as a nonmatch.

The cutoff thresholds T_μ and T_λ are determined by a priori error bounds on false matches and false nonmatches. Rule (2) agrees with intuition. If $\gamma \in \Gamma$ consists primarily of agreements, then it is intuitive that $\gamma \in \Gamma$ would be more likely to occur among matches than nonmatches and ratio (1) would be large. On the other hand, if $\gamma \in \Gamma$ consists primarily of disagreements, then ratio (1) would be small. Rule (2) partitions the set $\gamma \in \Gamma$ into three disjoint subregions. The region $T_\lambda \le R \le T_\mu$ is referred to as the no-decision region.

Pairs with weights above the upper cut-off are referred to as *designated matches* (or links). Pairs below the lower cut-off are referred to as *designated nonmatches* (or nonlinks). The remaining pairs are referred to as *designated potential matches* (or potential links). If $T_\mu = T_\lambda$, then decision rule (1) can be used for separating records (correspondingly documents) into those that are in one class from those that are not. The probabilities P(agree first | M), P(agree last | M), P(agree age | M), P(agree first | U), P(agree last | U), and P(agree age | U) are called *marginal probabilities*. P(| M) & P(| U) are called the m- and u-probabilities, respectively. The natural logarithm of the ratio R of the probabilities is called the *matching weight or total agreement weight*. The logarithms of the ratios of probabilities associated with individual fields (marginal probabilities) are called the *individual agreement weights*. The m- and u-probabilities are also referred to as *matching parameters*. A *false match* is a pair that is designated as a match and is truly a nonmatch. A *false nonmatch* is pair designated as a nonmatch and is a truly a match.

Fellegi and Sunter showed that it is possible to compute the unknown m- and u- probabilities directly in the 3-variable, conditional independence case. More generally, in the conditional independence situation, the parameters can be computed via a straightforward application of the EM algorithm (Winkler 1988). If the conditional independence assumption does not hold, then the parameters can be computed by generalized EM methods (Winkler 1988, 1989a, 1993b, Armstrong and Mayda 1993, see also Meng and Rubin 1993), by scoring (Thibaudeau 1993), and by Gibbs sampling (Larsen 1996, Larsen and Rubin 2000). The methods of Larsen and Rubin (2000) are the most general. These methods can yield more accurate matching parameters and better decision rules. These parameter-estimation methods do not always yield sufficiently accurate probability estimates for estimating record linkage error rates. A false match error-rate estimation method that is somewhat supplemental to these is due to Belin and Rubin (1995). Although the method of Belin and Rubin requires calibration data, it is known to work well in a narrow range of situations (Winkler and Thibaudeau 1991, Scheuren and Winkler 1993). The situations are those in which there is substantial separation of the curves of log frequency versus matching weight for matches and nonmatches.

Because record linkage must merge large administrative files (as many as 550 million records versus 300 million

records), it cannot have training data that consists all possible words.

2.2. Information Retrieval

Methods of information retrieval are often associated with library science applications in which a user puts in a query consisting of a set of words. The query words are matched against a set of keywords in order to bring back documents that agree with the query words. The retrieved documents may receive a score based on how many words in the document match the query words and weights that are assigned to each query word. A number of information retrieval models are described in Frakes and Baeza-Yates (1992). The book has excellent chapters covering stoplists, lexical analysis, and stemming. A stoplist is a listing of words such as 'the,' 'in,' and 'who' that are commonly occurring in most or all documents and that have little ability to distinguish documents into classes. In record linkage, an analogous concept is to eliminate words such as 'The,' 'Corporation,' and 'Company' in business names. Stemming replaces words such as 'engineered,' 'engineering,' and 'engineer' into a common word class such as 'engineer.' In record linkage, variations in the spellings of a word in a street address such as 'Drive,' 'Dr,' 'Drrive,' are replaced by a common spelling such as 'Dr.'

The term query model used in information retrieval uses an appropriate set of words in documents to determine whether a returned document is *relevant* or not. An initial set of words is put in a query by a user. The documents are judged relevant or not by the user. The appropriate set of words can be given weights

$$W_{ij} = \log ((n\_r/(R-n\_r))/((n-n\_r)/(N-n-R+n\_r))) \quad (3)$$

where W_{ij} is term weight for term (word) i in query j, n_r is the number of relevant documents for query j having term i, R is the total number of relevant documents for query j, n is the number of documents in the collection having term I, and N is the number of documents in the collection. The weights W_{ij} in (3) can be used in a new set of queries for which relevance determination is done and new re-weightings with the resultant new set of terms (words) is performed. The reweighting to improve information retrieval is called *relevance feedback*. The weighting methods given above are almost identical to those introduced for record linkage by Newcombe (Newcombe et al. 1959, Newcombe et al. 1962).

Several measures are used to measure accuracy. *Precision* p is the probability that a document is relevant if it has been designated as relevant. *Recall* r is the probability that a document is designated as relevant if it is relevant. Because of its objectivity, the precision-recall breakeven point is often used in comparing two information retrieval methods. Any given fixed weight divides documents into two sets. Those above the given weight are designated relevant and those below are designated as not relevant. The weight at which the precision and recall are equal is called the precision-recall breakeven point. Another useful measure is the Van Rijsbergen F-test that is given by F=2rp/(r+p). The values r and p are chosen so that F is minimized.

In record linkage, an analogous procedure to relevance feedback is as follows. An initial guess of record linkage parameters such as P(agree first name | M) and P(agree first | U) is used to obtain an initial matching output. A sample of pairs is reviewed (possibly with additional materials or field follow-up) to determine those that are truly matches or not. Based on the initial resolution of the sample of clerical pairs, the matching parameters P(agree first name | M) and P(agree first | U) are re-estimated and the matching is repeating. In many situations, the separation between the weights in the set of matches and in the set of nonmatches is improved.

2.3. Machine Learning

Mitchell (1997) provides a good introduction to machine learning. He broadly defines learning as: "A computer program is said to learn from experience E with respect to some class of task T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E." For classification of text, several methods are known to work well. Support Vector Machines (SVM) of Cortes and Vapnik (1995), Nearest-Neighbor, Linear Least Squares Fit (see e.g., Yang 1999), and boosting (Schapire and Singer 2000) are among the top-ranked classification methods (Yang 1999, Yang and Liu 1999). In the study of Yang and Liu (1999), Bayesian networks and neural networks (still good) had performance slightly worse than the best methods.

Nigam et al. (2000) observed two strengths of Bayesian networks. The first is that the method is based on a formal probabilistic model that lends itself to statistical interpretation. The second is that it provides a straightforward way of combining labeled and unlabelled data during training. In most machine learning applications, labeled training data for which the true classification status is known is used. Because training data are very expensive and unlabelled data are easy to collect, Nigam et al. (2000) showed how to combine moderate amounts of labeled data with varying amounts of unlabelled data to produce classification decision rules that improved on classification rules that were based on the moderate amounts of labeled data alone. They showed that too small an amount of labeled training would not yield suitable decision rules. Furthermore, they showed that, if too large an amount of unlabeled training data were combined with a moderate amount labeled training data, then decision rules could also be worse than those based on labeled data alone.

Nigam et al. (2000) and others (e.g., Yang and Liu 1999, Lewis and Ringuette 1994) have shown that classification decision rules that are based on naïve Bayesian networks (i.e., conditional independence

assumption) work well in practice. The conditional independence is useful because it makes computation much more tractable (Nigam et al. 2000, Winkler 1988). Varying authors have observed that the words in documents are quite dependent and that the computed probabilities for documents do not even remotely correspond to the true underlying probabilities. Winkler (1989a, 1993) observed that, if dependencies are dealt with, computed probabilities can somewhat correspond to the true probabilities in a few situations. Dependencies can be computed with conventional hierarchical latent class methods as introduced by Winkler (1989a, 1993) when the number of fields is moderate (say, 20 or less). Because the number of fields corresponds to the number of words in the vocabulary used for classification (as much as 24,000 or more), new computationally tractable computational methods are needed.

3. GENERAL EM PARAMETER-ESTIMATION THEORY

This section contains a general theorem that represents an extension of the record linkage theory of Fellegi and Sunter and presents a computational theory that allows computing probabilities for groups of words given a class when dependencies hold. Some examples from record linkage are used to motivate the extensions and provide intuition.

The basic ideas of record linkage were introduced by Newcombe et al. (1959) and Newcombe and Kennedy (1962). Fellegi and Sunter provided the formal mathematical model for record linkage and gave many ideas for computing the probabilities needed for the decision rules. Smith and Newcombe (1975) gave an ad hoc method for dealing with dependency that is still very effectively used on the British National Health Files (Gill 1999). Winkler (1989a) gave some ad hoc methods for dealing with dependencies. Winkler (1989, 1993) and Thibaudeau (1989, 1993) gave specific methods for dealing with dependencies. The computational procedures of Winkler (1989, 1993) are a variant of the ECM Algorithm given by Meng and Rubin (1993).

3.1. Examples and Ideas on Dependency

This example is taken from record linkage applications in which records contain name and address information. Two household files are matched as follows. The set of pairs of records are those agreeing on a geographic identifier such as ZIP+4 (approximately 50-100 contingent households). Identifying information is first name, last name, age, house number, street name, phone number. The intent is to correctly identify all true matches in a household. There are multiple individuals in most households. Because each geographic area is small, if a record pairs agrees on house number, it will typically agree on last name, street name, and phone number. If the pair agrees on last name, then it will agree on house number, street name, and phone number. The EM procedures of Winkler (1988, 1989) and Nigam et al. (2000) were originally developed to divide the set of pairs into two classes. If the EM-latent class procedure is applied to the unlabelled set of pairs, then the set of pairs naturally divide into those agreeing on household characteristics (last name, house number, etc.) and those that do not.

Winkler (1992) introduced a 3-class EM that effectively divided the set of pairs into three classes: (1) within household agreeing on name characteristics, (2) within household not agreeing on name characteristics, and (3) outside the same household. Winkler demonstrated that the probabilities from the 3-class EM yielded dramatically improved decision rules (for matching persons) and that the departures from the conditional independence assumption were dramatic in the classes (1) and (2). Using truth data, Winkler showed that conditional on agreement on last name, pairs in classes (1) and (2) agreed on house number, street name, and phone number with probability very close to one. Because P(agree last | C_1) = P(agree house number | C_1) = P(agree last, house number | C_1) < 0.25 where C_1 represents class (1), the departure from independence is substantial. In applying the decision rule for classifying pairs into matches and nonmatches, Winkler used the probabilities for the first class and the weighted combination of the probabilities from the first two classes. Nigam et al. (2000) also compare one class with the others by using the weighted combination of pairs in the other classes.

3.2. Text Classification Models

In the models of Nigam et al. (2000) and of this paper, words are used to classify documents into different class. Specifically,

$$P(d_i \mid \Theta) = \sum_i^{|C|} P(d_i \mid C_j ; \Theta)\, P(C_j ; \Theta) \qquad (4)$$

where d_i is a specific document, C_j is a specific class, and the sum is over the set of classes. Under the Naïve Bayes or conditional independence, we have

$$P(d_i \mid C_j ; \Theta) = \prod_k P(w_{di,k} \mid C_j ; \Theta) \qquad (5)$$

where the product is over the words $w_{di,k}$ in document d_i. We explicitly assume that the ordering of the words in a document is not important. In some situations, we use a Dirichlet prior

$$P(\Theta) = \prod_j (\Theta_{Cj})^{\alpha-1} \prod_k (\Theta\, w_{di,k \mid Cj})^{\alpha-1} \qquad (6)$$

where the first product is over the classes Cj and the second product is over the words in the vocabulary. Nigam et al. (2000) set α equal to two and refer to the effect of the prior as Laplace smoothing. The prior (6) helps keep most of the estimated probabilities away from zero. We use D^u to denote unlabeled documents and D^l to

denote labeled documents. Given the document collection D, the log likelihood is given by

$$l(\Theta \mid D) = \log(P(\Theta)) + \sum_{i \in Du} \log \sum_j P(d_i \mid C_j ; \Theta)\ P(C_j ; \Theta) + \sum_{i \in Dl} \log P(d_i \mid C_j ; \Theta)\ P(C_j ; \Theta) \quad (7)$$

where the first sum is over the unlabeled documents and the second sum is over the labeled documents. If we let z_{ij} be a missing data indicator that document i in class j is observed, then we have the complete data equation (CDE)

$$l_c(\Theta \mid D; z) = \log(P(\Theta)) + \sum_{i \in D} \sum_j z_{ij} \log (P(d_i \mid C_j ; \Theta)\ P(C_j ; \Theta)) \quad (8)$$

where the first sum is over all documents and the second sum is over the classes. If labeled and unlabelled documents are mixed in proportions λ and 1-λ, $0 < \lambda < 1$,we have

$$l_c(\Theta \mid D; z) = \log(P(\Theta)) + (1-\lambda) \sum_{i \in Du} \sum_j z_{ij} \log (P(d_i \mid C_j ; \Theta)\ P(C_j ; \Theta)) + \lambda \sum_{i \in Dl} \sum_j z_{ij} \log (P(d_i \mid C_j ; \Theta)\ P(C_j ; \Theta)). \quad (9)$$

We use the EM algorithm to estimate (9). The specific form of the EM algorithm depends on the exact parametric form that we assume for $P(d_i \mid C_j ; \Theta)\ P(C_j ; \Theta)$. Here we let

$$P(d_i \mid C_j ; \Theta) = \prod_k \mu_{jk}^{\gamma k} (1-\mu_{jk})^{(1-\gamma k)} \quad (10)$$

where the product is over all words in the vocabulary and γk is an indicator of whether word $w_{di,k}$ is observed in the document d_i. The starting points for the EM are the estimates of μ_{jk} and $P(C_j ; \Theta)$ that are available from the labeled data. Under the conditional independence assumption, if $\Theta^t = (\mu_{jk}^t , P^t (C_j ; \Theta) : j, k)$ is the current estimate of Θ, then

$$\mu_{jk}^{t+1} = [(\alpha-1) + (1-\lambda) \sum_{i \in Du} E(z_{ij} \mid Cj)\ \gamma k + \lambda \sum_{i \in Dl} E(z_{ij} \mid Cj)\ \gamma k] / [2(\alpha-1) + (1-\lambda) \sum_{i \in Du} E(z_{ij} \mid Cj)\ 1 + \lambda \sum_{i \in Dl} E(z_{ij} \mid Cj)\ 1\ ] \quad (11)$$

and

$$P^{t+1} (C_j ; \Theta) = [[(\alpha-1) + (1-\lambda) \sum_{i \in Du} E(z_{ij} \mid Cj) + \lambda \sum_{i \in Dl} E(z_{ij} \mid Cj)\ ]/ [|C|(\alpha-1) + (1-\lambda) \sum_{i \in Du} 1 + \lambda \sum_{i \in Dl} 1] \quad (12)$$

If expected values $E(z_{ij} \mid Cj)$ are substituted in the (9), then Equation (11) follows by taking partial derivatives and setting the resultant equation equal to zero. Equation (12) follows by standard multinomial reasoning (e.g., McLachlan and Krishnan, pp. 17-19). The parameter α can be varied independently for μ_{jk} and $P(C_j ; \Theta)$. For the empirical example, we set α equal to 2 in (12) and α equal to 1.01 in (11). The smoothing via different values of α in the prior causes the successive estimates μ_{jk}^{t+1} and $P^{t+1} (C_j ; \Theta)$ to stay away from zero.

An alternative form of smoothing is to add a small value δ to every cell as suggested by Larsen (1996). With a moderate size of vocabulary or with different parametric representations of the underlying words in a document, some differing documents can have the same representation. In some instances, differing documents may have the same representation in words. This can happen with a small vocabulary and situations in which only the first instance of word such as corn is used in the parametric representation. We use freql(i, j) to represent the frequency of the jth pattern in the ith class of the labeled documents and frequ(j) be the frequency of the jth pattern in the unlabelled documents. With a slight abuse of notation, we let the sum over the labeled documents to be over all of the observed patterns in the labeled and unlabeled data. The understanding is that, for a given j, freql(i,j) is zero for all i if a pattern is observed only in the unlabeled data. Similar to Equation (9) we have,

$$l_c(\Theta \mid D; z) = (1-\lambda) \sum_{i \in Du} \sum_j z_{ij}\ frequ(j)\ \log (P(d_i \mid C_j ; \Theta)\ P(C_j ; \Theta)) + \lambda \sum_{i \in Dl} \sum_j z_{ij}\ (freql(i,j) + \delta)\ \log (P(d_i \mid C_j ; \Theta)\ P(C_j ; \Theta)). \quad (13)$$

In Equation (13), the value δ is added to each cell in every class of every observed data pattern from the labeled and unlabeled data. In analogy to Equations (9) and (10), we have estimates at step t of

$$\mu_{jk}^{t+1} = [(1-\lambda) \sum_{i \in Du} frequ(j)\ E(z_{ij} \mid Cj)\ \gamma k + \lambda \sum_{i \in Dl} (freql(i,j) + \delta)\ E(z_{ij} \mid Cj)\ \gamma k] / [(1-\lambda) \sum_{i \in Du} frequ(j)\ E(z_{ij} \mid Cj)\ 1 + \lambda \sum_{i \in Dl} (freq(i,j) + \delta)\ E(z_{ij} \mid Cj)\ 1\ ] \quad (14)$$

and

$$P^{t+1} (C_j ; \Theta) = [[(1-\lambda) \sum_{i \in Du} E(z_{ij} \mid Cj) + \lambda \sum_{i \in Dl} (freql(i,j) + \delta)\ E(z_{ij} \mid Cj)\ ]/ [(1-\lambda) \sum_{i \in Du} 1 + \lambda \sum_{i \in Dl} (freql(i,j) + \delta)\ 1] \quad (15)$$

3.2. Theoretical Methods for Computing Dependencies

This subsection presents theoretical methods for improving computational speed that will be applied in the results section. Preliminary background is needed before going into the details.

The identifying information $\gamma \in \Gamma$ is used in the probabilities that are used in the decision rules. If $\gamma \in \Gamma$

represents n identifying characteristics (i.e., fields or words), then the probabilities can be represented under the independence assumption by

$$P(\gamma \in \Gamma) = P(\gamma_1, \gamma_2, \ldots, \gamma_n) = P(\gamma_1)\ P(\gamma_2) \ldots\ P(\gamma_n), \quad (16)$$

where, for simplicity, each γ_i represents presence or absence of a characteristic. If the independence assumption does not hold, then we may need to compute the interactions associated with, say, the first k terms. Then the 2^k probabilities $P(\gamma_1^c, \gamma_2^c, \ldots, \gamma_k^c, \ldots)$, where γ_i^c, represents either the presence or absence of one of the ith term, $0 \le i \le k$, must be computed. If the conditional independence assumption is not made, then each maximization step of the EM can be done by iterative proportional fitting (e.g., Meng and Rubin 1993, Winkler 1989, 1993) and can be quite slow.

Nigam et al. (2000) were able to get their conditional independence EM to converge in less than 20 minutes for large dimensional situations by implicitly making simplifying assumptions that are valid in their situation. To develop a computationally tractable method for EM in situations when conditional independence does not hold, a more comprehensive theoretical development is needed.

To obtain appropriate candidate interactions of words to fit in different classes, set covering algorithms can be used to find the most frequently occurring n-tuples of words in records (documents). Let a 3-tuple of words occur in one class and not in other classes. Then it has the capability of completely separating the records in the class that contain the 3-tuple from the other classes. The following illustrates the situation. Assume that there are two primary classes C_1 and C_2 and no secondary classes of records (documents). Let (w_1, w_2, w_3) be a 3-tuple of words in a document. Let $P(w_1 \mid C_1) = P(w_2 \mid C_1) = P(w_3 \mid C_1) = 0.3$ and let $P(w_1 \mid C_2) = P(w_2 \mid C_2) = P(w_3 \mid C_2) = 0.1$. Assume that $P(w_1, w_2, w_3 \mid C_1) = 0.002$ and $P(w_1, w_2, w_3 \mid C_2) = 0.000$. Under conditional independence, the probability ratio associated with the 3-tuple (w_1, w_2, w_3) is

$$P(w_1, w_2, w_3 \mid C_1) / P(w_1, w_2, w_3 \mid C_2)$$

can be set to an arbitrarily high value that is sufficient to overcome other words in the record and assure that the document is correctly placed in class C_1.

The specific computational procedure can be best understood if the z_{ij} in Equation (13) can be replaced $E(z_{ij} \mid \Theta^t)$

$$L_e(\Theta^{t+1} \mid D; z) = (1-\lambda) \sum_{i \in Du} \sum_j E(z_{ij} \mid \Theta^t)\ \text{frequ(j)} \log (P(d_i \mid C_j ; \Theta^t)\ P(C_j ; \Theta^t)) + \lambda \sum_{i \in Dl} \sum_j E(z_{ij} \mid \Theta^t)\ (\text{freql(i,j)} + \delta) \log (P(d_i \mid C_j ; \Theta^t)\ P(C_j ; \Theta^t)). \quad (18)$$

We can assume that both first summations are over all of the observed patterns in the labeled and unlabeled by setting frequ(j) and freql(j) equal to zero when j is a pattern that is not in the unlabeled data and labeled data, respectively. If we renormalize, the coefficients in front of the logs so that the terms add to one (which does affect the maximization of the likelihood), then we have equations of the following form

$$L_e(\Theta^{t+1} \mid D; z) = \sum_{ij} p_{et}(i,j) \log (p_t(i,j))$$

where $p_{et}(i,j) = ((1-\lambda)\ E(z_{ij} \mid \Theta^t)\ \text{frequ(j)} + \lambda\ E(z_{ij} \mid \Theta^t)\ (\text{freql(i,j)} + \delta))/N_C$, N_C is the normalization constant, and $p_t(i,j) = P(d_i \mid C_j ; \Theta^t)\ P(C_j ; \Theta^t)$. Let P_j be the interaction patterns that are to be fit in class C_j. Each interaction pattern in P_j is represents a listing of the terms (words) that must be summed over. For pattern i in P_j, let I_i represent the specific subsets l of words For instance, if P_i represents the presence of k specific terms in a document, then I_i has 2^k subsets. The 2^k subsets in I_i partition the entire set of documents. In the following , the notion $i \in l$ means that the document i has the pattern of words represented by l. The specific fitting procedure F_t at step t is:

1. For each pattern i in Pj and each l in I_i, let $M_{tl} = \sum_{i \in l} p_t(i,j)$ and $E_{tI} = \sum_{i \in l} p_{et}(i,j$. For each class $k \neq j$, let $M_k = \sum_i p_t(i,k)$ and $E_k = \sum_i p_{et}(i,k)$.
2. If $i \in l$ in P_j, then $p_{t+1}(i,j) = p_t(i,j)\ E_{tl} / M_{tl;}$ and, if $k \neq j$, $p_{t+1}(i,k) = p_t(i,k)\ E_k / M_k$.
3. Repeat 1 and 2 for all classes C_j and all patterns i in P_j.

Then each F_t is one cycle of iterative proportional fitting (e.g., Winkler 1989, 1993, Meng and Rubin 1993) and increases the likelihood. The last equation in step 2 assures that the new estimates add to a proper probability. If necessary, the procedure can be extended to general I-Projections that also increase the likelihood and have strong constraints for keeping the probability estimates $p_t(i,j)$ from converging to zero or one (e.g., Winkler 1990). The smoothing with the constant delta in Equation (18) has the effect of assuring that most probability estimates $p_t(i,j)$ do not converge to zero. For a fixed pattern i, some of the probability estimates $p_t(i,j)$, however, may differ by more than ten orders of magnitude across the different classes C_j. If necessary, affine constraints may be use to restrict the differing relative sizes of the $p_t(i,j)$ (Winkler 1990).

4. REUTERS DATA AND SET-COVERING TO OBTAIN A VOCABULARY

This section describes the Reuters data and the methods used for obtaining a classification vocabulary.

4.1. Reuters data

The Reuters data have been used by various authors (Lewis 1992, Lewis and Ringuette 1992, Yang 1999, 1999,Yang and Liu 1999, Schapire and Singer 2000, Nigam et al. 2000) as a test deck for evaluating differing methods of text classification. The deck has been extensively cleaned by David Lewis and others so that it can be used in comparisons. The deck consists of newspaper articles in SGM format. Each article has identifiers into which classes it belongs and whether it was used as training data or test data by Lewis. Articles generally consist of a title plus text that corresponds to newspaper articles. Many articles consist of title information only. Most of the articles contain only a few lines of text. Only a few articles are more than three paragraphs. In many situations, the text information has been blanked or truncated. In the initial comparisons, the ten classes *acq, earn, corn, crude, grain, interest, money-fx, ship, trade,* and *wheat* that were used by Nigam et al. (2000) are used. Classes *earn* and *acq* contain more than 3000 and 1700 documents, respectively. Classes *corn, crude, grain, ship, trade*, and *wheat* contain between 30 and 100 documents. The stoplist of Lewis (1992) is supplemented with additional frequently occurring words such as 'within' that occur in most or all of the documents. The intuition is that, if 'within' occurs a moderate number of times in every class, it has little or no ability to distinguish between the classes.

4.2. Set Covering to Obtain a Vocabulary

Two criteria are used in obtaining an initial vocabulary of words in documents that are not in the stoplist. The first is that the words are the most frequently occurring in a class. The second is that the set of words associated with a class cover most documents in a class. A word *covers* a document in a class if it is one of the words in the document. An implicit assumption about training data is that they yield documents and words that are representative of a class. The intent is that the most frequently occurring words from the test deck will also cover the documents in the test data.

If the criteria of using the most frequently occurring words were relaxed, it might be possible to find a vocabulary on the training data such that the words associated with a given class formed a set that is disjoint from the words in any other class. In such a situation, the words in the vocabulary would perfectly separate the classes and there would be no possibility of error on the test data. If two classes had substantial overlap, then it would only be necessary to find covering words that separated the documents that are only in one class. If it were possible to find a cover of words that are somewhat frequently occurring (say in 5% of the documents in a class), then such a separating cover might also be used. The disadvantage of such a separating vocabulary of words is that it may not work well with the test data.

The specific set of words in the vocabulary is constructed as follows. All words such as 'within' that occur frequently and in most or all documents are added to the stoplist. They are no longer considered. The initial pass at the vocabulary takes all words in a class that occur in 3/4 or more of the documents. The vocabulary consists of the 1027 words in the naïve Bayes application and 1094 in the general interaction Bayes application. With the first vocabulary, approximately 3% of the documents in *acq*, *earn*, and *interest* and approximately 4% of the documents in money-fx are not covered. Rather than use all 24,000 words in the documents to look for pairs or triples of words that occur in documents, it is much more efficient to use 1027 words. How the interactions (dependencies) can be modeled was described in detail in section 3.

5. RESULTS

This section presents final results for naïve Bayes and preliminary results for interaction fitting.

5.1. Naïve Bayes

The results of this section (Table 1) compare the performance of the Bayesian network procedures of this paper with the Bayesian network procedures of Nigam et al. (2000). The results presented in this paper under the column Winkler are for parameters under EM-fitting procedures where the probabilities are assumed to be

Table 1. Classification Rates Under Independence
Vocabulary Size 1027, Number of Patterns is 11639
Training Data Size 9603, Test Data Size 3299

Class	Winkler		Nigam	
	Precision	Recall	Precision	Recall
acq	.933	.933	.839	.839
cor	.500	.524	.528	.528
cru	.806	.806	.754	.754
ear	.960	.958	.892	.892
gra	.708	.689	.723	.723
int	.580	.580	.523	.523
mon	.733	.733	.569	.569
shi	.837	.837	.525	.525
tra	.663	.652	.618	.618
whe	.679	.699	.678	.678
micro avg	.865	.859	.784	.784

products of multinomials rather than the multinomials used by Nigam et al. (2000). The microaverage is computed as suggested by Yang (1999).

Examination of Table 1 shows that the microaverage of the precision is 0.865 for procedures of this paper in contrast to the microaverage of 0.784 of Nigam et al. The microaverage of the recalls in 0.865 in contrast to 0.784.

Classes *acq* and *ear* account for more than 50% of the test documents.

5.2. Interaction Fitting – Preliminary Results

We present very preliminary results for fitting interaction models to third class cru. The precision and recall under interaction are .880 and .880, respectively. Under naïve Bayes, precision and recall are .806 and .806, respectively.

The preliminary algorithms are:

1. Algorithms use all training and test data to get a vocabulary and sets of pairs and triples of words that a frequently occurring in individual classes.
2. The general interaction EM uses a set of interactions from a file that is progressively updated with additional interactions.
3. A collection of algorithms that determine new two- and three-way interactions to fit based on the results of a classification pass. The training data is partitioned into three sets. Training is done three times with 2/3 of the data and the remaining 1/3 used as test data. Documents that are falsely classified into class *cru* with moderate weight and above may yield new interactions to fit in the class where the document belongs. Documents that are correctly classified into class *cru* and have low weight may yield new interactions to fit in class *cru*.
4. Steps 2 and 3 are repeated until accuracy improves sufficiently.

Specifically, the first classification pass yields precisions and recall of 0.4 and 0.4, respectively. In each class, the interactions associated with highest frequency pairs and triples of words are fit. For each successive fitting pass, additional interactions are added. The 11th pass yielded precision and recall above 0.8 and 0.8, respectively. The 12^{th} pass yielded 0.880 and 0.880, respectively.

6. DISCUSSION

The methods of this paper used a moderate size of vocabulary of words chosen from all of the training data and all of the test data. All of the data patterns observed in the training and in the unlabeled test data were used. Various authors (Weiss et al. 1999, Lewis and Ringuette 1994) have the observed that classification rules based on a subset of the vocabulary can work moderately well.

Nigam et al. (2000) used a modest proportion of the training data for the supervised portion of the learning and a modest proportion of the unused training data. Their purpose was to show how relatively modest amounts of training data could be combined with moderate amounts of test data. The differences between the naïve Bayes results of Nigam et al. and this paper are likely due to our use of slightly different probability model assumptions and much more of the available data.

We expect the interaction results to improve for the third class and work well as we more effectively increase the size of the vocabulary. They will improve further as we learn to use 2-way and 3-way interaction models more effectively. From review of the interaction-.88-.88 results, we can observe that some documents are not covered by a sufficient number of words. We need to increase the size of the vocabulary to improve the classification of a modest number of documents. As observed by earlier users of the Reuter's collection, a modest percentage of documents in the training set are almost certainly still misclassified. These misclassifications affect the results.

The software is slow. Although we are only fitting 8,000 out of a potential 120 million 2- and 3-way interactions, the set of generalized EM runs require approximately two hours. The remaining software runs also require approximately one hour. We have not yet developed efficient set-covering algorithms for determining the most parsimonious sets of interactions.

7. CONCLUDING REMARKS

This paper provides a general computational theory for estimating classification probabilities in machine learning, information retrieval, and record linkage. The record linkage model of Fellegi and Sunter (1969) can be shown to generalize the classification model for Bayesian networks given by Nigam et al. (2000). Theoretical and computational procedures are given for computing the probabilities used in the decision rules when the conditional independence assumption does not hold. This paper presents a somewhat different model for conditional-independence (Naïve) Bayesian networks. It compares results with those of Nigam et al. (2000) using the Reuters collection of newspaper articles.

This paper reports the results of research and analysis undertaken by Census Bureau staff. It has undergone a Census Bureau review more limited in scope than that given to official Census Bureau publications. This report is released to inform interested parties of research and to encourage discussion. A longer research report is available at http://www.census.gov/srd/www/byyear.html.

REFERENCES

Armstrong, J. B., and Mayda, J. E. (1993), "Estimation of Record Linkage Models Using Dependent Data," *Survey Methodology*, **19**, 137-147.

Belin, T. R., and Rubin, D. B. (1995), "A Method for Calibrating False- Match Rates in Record Linkage," *Journal of the American Statistical Association*, **90**, 694-707.

Cortes, C., and V. Vapnik, (1995), "Support Vector Networks," *Machine Learning*, **20**, 273-297.

Fellegi, I. P., and A. B. Sunter (1969), "A Theory for Record Linkage," *Journal of the American Statistical Association*, **64**, 1183-1210.

Frakes, W. B., and Baeza-Yates, R. (ed.) (1992), *Information Retrieval: Data Structures & Algorithms*,

Upper Saddle River, NJ: Prentice-Hall PTR.
Freund, Y. and R. E. Schapire (1996), "Experiments with a New Boosting Algorithm," *Machine Learning: Proceedings of the Thirteenth International Conference*, 148-156.
Freund, Y. and R. E. Schapire (1997), "A Decision-theoretic Generalization of On-line Learning and an Application to Boosting," *Journal of Computer and Systems Sciences*, **55**, 119-139.
Freund, Y. and R. E. Schapire (1999), "A Short Introduction to Boosting," *Journal of the Japanese Society For Artificial Intelligence*, **14**, 771-780.
Friedman, J., T. Hastie, R. Tibshirani (2000), "Additive Logistic Regression: a Statistical View of Boosting (with discussion)," *Annals of Statistics*, **28**, April 2000.
Gill, L. (1999), "OX-LINK: The Oxford Medical Record Linkage System," in *Record Linkage Techniques 1997*, Washington, DC: National Academy Press, 15-33.
Larsen, M. D. (1996), "Bayesian Approaches to Finite Mixture Models," Department of Statistics, Harvard University, Ph.D.Thesis.
Larsen, M. D., and D. B. Rubin (2000), "Iterative Automated Record Linkage Using Mixture Models," Statistics Department Technical Report, University of Chicago.
Lewis, D. D. (1992), "Representation and Learning in Information Retrieval," Technical Report 91-93, Department of Computer Science, University of Massachusetts, Ph.D. Thesis.
Lewis, D. D. and M. Ringuette (1994), "A Comparison of Two Learning Algorithms for Text Categorization," *Third Annual Symposium on Document Analysis and Information Retrieval*, 81-93.
McLachlan, G. J., and T. Krisnan, (1997), *The EM Algorithm and Extensions*, John Wiley: New York.
Meng, X., and D. B. Rubin (1993), "Maximum Likelihood Via the ECM Algorithm: A General Framework," *Biometrika*, **80**, 267-278.
Mitchell, T. M. (1997), *Machine Learning*, New York, NY: McGraw-Hill.
Newcombe, H. B. (1988), *Handbook of Record Linkage: Methods for Health and Statistical Studies, Administration, and Business*, Oxford: Oxford University Press (out of print).
Newcombe, H. B., J. M. Kennedy, S. J. Axford, and A. P. James (1959), "Automatic Linkage of Vital Records," *Science*, **130**, 954-959.
Newcombe, H. B. and J. M Kennedy, (1962), "Record Linkage making Maximum Use of the Discrimination Power of Identifying Information," *Communications of the ACM*, **5**, 563-566.
Nigam, K., A. K. McCallum, S. Thrun, and T. Mitchell (2000), "Text Classification from Labeled and Unlabelled Documents using EM, *Machine Learning*, **39**, 103-134.
Salton, G. (1991), "Developments in Automatic Text Retrieval," *Science*, **253**, 974-980.
Salton, G. and M. J. McGill (1983), *Introduction to Modern Information Retrieval*, McGraw-Hill.
Schapire, R. E. and Y. Singer (2000), "BoosTexter: A Boosting-based System for Text Categorization, *Machine Learning*, **39**, 135-168.
Thibaudeau, Y. (1989), "Fitting Log-Linear Models When Some Dichotomous Variables are Unobservable," in *Proceedings of the Section on Statistical Computing, American Statistical Association*, 283-288.
Thibaudeau, Y. (1993), "The Discrimination Power of Dependency Structures in Record Linkage," *Survey Methodology*, **19**, 31-38.
Titterington, D. M., A. F. M. Smith, U. E. Makov (1988), *Statistical Analysis of Finite Mixture Distributions*, NewYork: J. Wiley.
Weiss, S., C. Apte, F. J. Damerau, D. E. Johnson, F. J. Oles, T. Goetz, and T. Hammp (1999), Maximizing Text Mining Performance, IEEE Intelligent Systems, 2-8.
Winkler, W. E. (1988), "Using the EM Algorithm for Weight Computation in the Fellegi-Sunter Model of Record Linkage," *Proceedings of the Section on Survey Research Methods, American Statistical Association*, 667-671.
Winkler, W. E. (1989a), "Near Automatic Weight Computation in the Fellegi-Sunter Model of Record Linkage," *Proceedings of the Fifth Census Bureau Annual Research Conference*, 145-155.
Winkler, W. E. (1989b), "Methods for Adjusting for Lack of Independence in an Application of the Fellegi-Sunter Model of Record Linkage," *Survey Methodology*, **15**, 101-117.
Winkler, W. E. (1990), "On Dykstra's Iterative Fitting Procedure," *Annals of Probability*, **18**, 1410-1415.
Winkler, W. E. (1993), "Improved Decision Rules in the Fellegi-Sunter Model of Record Linkage," *Proceedings of the Section on Survey Research Methods, American Statistical Association*, 274-279.
Winkler, W. E. (1994), "Advanced Methods for Record Linkage," *Proceedings of the Section on Survey Research Methods, American Statistical Association*, 467-472 (longer version report 94/05 available at http://www.census.gov/srd/www/byyear.html).
Winkler, W. E. and Thibaudeau, Y. (1991), "An Application of the Fellegi-Sunter Model of Record Linkage to the 1990 U.S. Census," U.S. Bureau of the Census, Statistical Research Division Technical report RR91/09.
Yang, Y. (1999), "An evaluation of Statistical Approaches to Text Categorization," *Information Retrieval*, **1/2**, 67-88.
Yang, Y. and X. Liu, (1999), "A Re-examination of text categorization methods," Association of Computing Machinery, Proceeding of the SIGIR, 42-49.
Yang, Y. and J. P. Pedersen, (1997), "A Comparative

Study on Feature Selection in Text Categorization," in D. H. Fisher, Jr. (ed.), *The Fourteenth International Conference on Machine Learning*, Morgan Kaufmann, 412-420.