

Using a Bayesian Network Induction Approach for Text Categorization

Wai Lam and Kon Fan Low and Chao Yang Ho

Department of Systems Engineering and Engineering Management

The Chinese University of Hong Kong

Shatin

Hong Kong

Abstract

We investigate Bayesian methods for automatic document categorization and develop a new approach to this problem. Our new approach is based on a Bayesian network induction which does not rely on some major assumptions found in a previous method using the Bayesian independence classifier approach. The design of the new approach as well as its justification are presented. Experiments were conducted using a large-scale document collection from Reuters news articles. The results show that our approach outperformed the Bayesian independence classifier as measured by a metric that combines precision and recall measures.

1 Introduction

Document categorization has become an important area in information retrieval and gathering. The objective of document categorization is to assign categories to a document. The categories are drawn from a set of pre-specified categories. It is possible to assign multiple categories to a document if they are appropriate. A document here refers to a piece of text. Traditionally this categorization task is performed manually by domain experts. Each incoming document has to be read and comprehended by an expert who assigns its categories. Obviously a large amount of human resources are required to carry out such categorization tasks. For instance, the OHSUMED document collection, composed of medical journal articles, requires a large amount of resources for manual categorization [Hersh *et al.*, 1994].

A promising way to solve this problem is to automatically categorize documents based on their content. This can be achieved by learning a categorization scheme from a set of training examples. This problem can be viewed as a machine learning problem where an element may be classified into multiple category labels. Recently, some work has been done on this problem [Apte *et al.*, 1994;

Lewis *et al.*, 1996; Yang, 1994; Larkey and Croft, 1996]. We present a new approach based on a Bayesian network induction. We show that it relaxes some major assumptions found in a previous approach based on the Bayesian independence classifiers [Lewis, 1992]. In this paper, the basic idea of the Bayesian independence classifier approach is first reviewed. We then introduce our approach for inducing Bayesian networks. We also contrast its relationship to the Bayesian independence classifier approach. Experiments were conducted using a large-scale document collection concerning with economic and financial news articles from Reuters. The results show that our approach outperformed the Bayesian independence classifier as measured by a metric that combines precision and recall measures.

2 Bayesian Independence Classifier Approach

In order to handle multiple category assignments, a separate classifier is constructed for each category. Each classifier is then used to determine whether a given document should be assigned to this category. This yields a set of categories for each document.

Lewis [Lewis, 1992] used a Bayesian approach for text categorization problems. Suppose we need to build a classifier for a particular category. Let the random variable C_j denote the membership of a document in the j -th category and it can take on binary values 0 and 1. Let d be an incoming document to be categorized. According to Bayes theorem,

$$P(C_j = 1 | d) = \frac{P(d | C_j = 1)P(C_j = 1)}{P(d)} \quad (1)$$

To represent document d , we use a vector $\vec{T}_j$ of chosen predictor terms $(T_{j_1}, \dots, T_{j_p})$. Each term usually represents a particular word. Note that we may use different terms to represent the same document under different categories. Each predictor term can take on binary values 0 and 1 denoting absence or presence of the

corresponding word in a document. Hence, Equation 1 becomes:

$$P(C_j = 1 | \vec{T}_j) = \frac{P(\vec{T}_j | C_j = 1)P(C_j = 1)}{P(\vec{T}_j)} \quad (2)$$

Suppose the value of each term in vector $\vec{T}_j$ is $(t_{j1}, \dots, t_{jp})$. Equation 2 can be expressed as:

$$\begin{aligned} P(C_j = 1 | \vec{T}_j) \\ = \frac{P(T_{j1} = t_{j1}, \dots, T_{jp} = t_{jp} | C_j = 1)P(C_j = 1)}{P(T_{j1} = t_{j1}, \dots, T_{jp} = t_{jp})} \end{aligned} \quad (3)$$

This formula requires estimates for the joint probabilities $P(T_{j1} = t_{j1}, \dots, T_{jp} = t_{jp} | C_j = 1)$ and $P(T_{j1} = t_{j1}, \dots, T_{jp} = t_{jp})$. Typically these probabilities are difficult to estimate from the training examples.

To overcome this problem, the following two assumptions are usually made regarding the marginal and conditional independence of predictor terms as follows:

$$\begin{aligned} P(\vec{T}_j | C_j = 1) \\ = P(T_{j1} = t_{j1} | C_j = 1) \cdots P(T_{jp} = t_{jp} | C_j = 1) \end{aligned} \quad (4)$$

and

$$P(\vec{T}_j) = P(T_{j1} = t_{j1}) \cdots P(T_{jp} = t_{jp}) \quad (5)$$

Substituting Equations 4 and 5 into Equation 3, we get:

$$P(C_j = 1 | \vec{T}_j) = P(C_j = 1) \prod_i \frac{P(T_{ji} = t_{ji} | C_j = 1)}{P(T_{ji} = t_{ji})} \quad (6)$$

Ideally, we would like a feature term which is strongly related to the ideas that is conveyed by the document. Sometimes the word appears in the document and we can of course assign it as a predictor feature. However, it is also possible that the word does not appear in a document and the term may still relate to the ideas conveyed by the document. We are therefore not sure whether an index term should be assigned to a document. To incorporate this kind of uncertainty, the probability $P(\vec{T}_j | d)$ is introduced to indicate how close $\vec{T}_j$ is to the ideal indexing of the document. Each element T_{ji} in $\vec{T}_j$ can have binary values 0 or 1. Instead of directly estimating $P(C_j = 1 | d)$ as in Equation 1, we need to consider $P(\vec{T}_j | d)$. Hence the formula for $P(C_j = 1 | d)$ is modified as follows:

$$P(C_j = 1 | d) = \sum_{\vec{T}_j \in \{0,1\}^p} P(C_j = 1 | \vec{T}_j)P(\vec{T}_j | d)$$

Applying Bayes' Theorem, we get

$$\begin{aligned} P(C_j = 1 | d) \\ = \sum_{\vec{T}_j \in \{0,1\}^p} P(C_j = 1) \frac{P(\vec{T}_j | C_j = 1)}{P(\vec{T}_j)} P(\vec{T}_j | d) \end{aligned} \quad (7)$$

In order to compute the above expression, the independence assumption in Equations 4 and 5 are still required. In addition, it requires the following assumption:

$$P(\vec{T}_j | d) = P(T_{j1} = t_{j1} | d) \cdots P(T_{jp} = t_{jp} | d) \quad (8)$$

Substituting Equations 4, 5 and 8 into 7, we get:

$$\begin{aligned} P(C_j = 1 | d) \\ = P(C_j = 1) \\ \sum_{t_{ji} \in \{0,1\}} \frac{P(T_{j1} = t_{j1} | C_j = 1) \cdots P(T_{jp} = t_{jp} | C_j = 1)}{P(T_{j1} = t_{j1}) \cdots P(T_{jp} = t_{jp})} \\ P(T_{j1} = t_{j1} | d) \cdots P(T_{jp} = t_{jp} | d) \\ = P(C_j = 1) \\ \sum_{t_{ji} \in \{0,1\}} \prod_i \frac{P(T_{ji} = t_{ji} | C_j = 1)P(T_{ji} = t_{ji} | d)}{P(T_{ji} = t_{ji})} \end{aligned} \quad (9)$$

To evaluate the expression in Equation 9, Lewis [Lewis, 1992] developed an approximation as follows:

$$\begin{aligned} P(C_j = 1) \\ \prod_i \left[\frac{P(T_{ji} = 1 | C_j = 1)P(T_{ji} = 1 | d)}{P(T_{ji} = 1)} + \frac{P(T_{ji} = 0 | C_j = 1)P(T_{ji} = 0 | d)}{P(T_{ji} = 0)} \right] \end{aligned} \quad (10)$$

3 Bayesian Network Approach

A major assumption in the Bayesian independence classifiers is that the features are marginally independent as well as conditionally independent to C_j or d . This assumption is used to estimate those high order probabilities such as $P(\vec{T}_j)$, $P(\vec{T}_j | C_j)$, and $P(\vec{T}_j | d)$ as shown in Equations 4, 5, and 8. The accuracy of these estimations directly affect the performance of the classifier. Poor estimation will probably not result in good classification performance. In our text categorization problem, it is likely that the predictor terms are not independent. We develop a new approach based on a Bayesian network induction that eliminates this assumption.

A Bayesian network consists of nodes and arcs connected together to form a directed acyclic graph. Each node represents a variable that can take on a discrete set of domain-specific states. Each (directed) arc represents a probabilistic dependency between the source and destination variables.

The idea is to learn a Bayesian network for the set of variables representing the predictor terms $\vec{T}_j$ as well as the category C_j . The network provides a means for us to exploit the inherent dependency among these variables. Instead of relying on estimating high order probabilities as described above, we can compute $P(C_j = 1 | \vec{T}_j)$ directly from the Bayesian network which already encodes

the dependencies among the terms in $\vec{T}_j$ as well as the category C_j .

Apart from an acyclic graph structure, a Bayesian network consists of a set of conditional probability parameters associated with each node in the network. Suppose X denotes a node representing one of the domain variables in a Bayesian network. Let Υ_X be the set of parents of node X in the network structure (i.e., there is a directed arc from each node in Υ_X pointing to X). There is a conditional probability distribution $P(X | \Upsilon_X)$ associated with X . If X does not have parents in the network structure, there is a prior probability distribution $P(X)$ associated with X .

After the network is constructed, it can be used for conducting reasoning. A common and useful kind of reasoning is to perform probabilistic inferences. In a typical scenario, the states of some variables are observed. We are interested in calculating the posterior probability distribution of some other variables given the instantiation of the observed variables.

To use Bayesian networks as classifiers in our text categorization problem, we first construct a Bayesian network for variables in $\{C_j\} \cup \vec{T}_j$ (i.e., $\{C_j, T_{j1}, \dots, T_{jp}\}$). Machine learning techniques can be used to automatically construct a network from training documents. Once this network is constructed, it can be used to categorize documents, using probabilistic inferences on the network. More specifically, the feature variables representing the document (i.e., variables in $\vec{T}_j$) are instantiated appropriately. The posterior probability $P(C_j = 1 | \vec{T}_j)$ is then computed. The document will be determined as being assigned to the category if this probability value exceeds a certain threshold. The main advantage of this approach over the Bayesian independence classifier is that it relaxes those assumptions regarding feature independence. The network structure can capture the inherent dependency during the induction process.

The next issue is how to learn a Bayesian network from the training documents. There has been some research on learning Bayesian networks from data [Heckerman *et al.*, 1995; Spirtes and Meek, 1995; Lam and Bacchus, 1994]. It is well known that learning arbitrary networks is quite computationally expensive. Since our purpose is to make use of the network for classification, we restrict our consideration to a family of network structures known as *classification-based network structures*. This can reduce the learning complexity considerably and at the same time achieve our objective for performing classification. A classification-based network structure is basically an arbitrary Bayesian network structure with the restriction that C_j has no parents (i.e., $\Upsilon_{C_j} = \emptyset$). Figure 1 shows an example of a classification-based network structure. The rationale behind these structures is

that the states of the feature variables may depend on the membership of the category, but not the other way around, if such dependencies exist. Bayesian networks have been adopted to build other classifiers [Friedman and Goldszmidt, 1996]. We focus on exploring Bayesian networks to solve large-scale document categorization problems and propose classification-based network structures to solve these tasks.

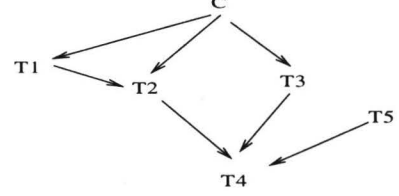


Figure 1: An Example of a Classification-Based Network Structure

4 Automatic Construction of Bayesian Networks

To learn a Bayesian network from training examples, we make use of a technique based on the *Minimum Description Length* (MDL) principle derived from [Lam and Bacchus, 1994]. Central to this approach is a *score metric* for a candidate network structure, where the score metric is a function representing the *total description length*, $L(B)$, of a candidate network structure B .

The total description length is decomposed into a function L_{total} associated with each individual node in the network as follows:

$$L(B) = \sum_X L_{\text{total}}(X, \Upsilon_X)$$

where X is a node in the network. L_{total} of a node X depends only on its parent set. Hence, L_{total} is a constant if the parent set is fixed. The domain variables in our current categorization problem are $\{C_j\} \cup \vec{T}_j$ (i.e., $\{C_j, T_{j1}, \dots, T_{jp}\}$). The network structures we consider are classification-based network structures as described in Section 3. Thus, node C_j has a fixed set of parents, i.e., empty parent set. As a result, the score metric of a candidate network becomes:

$$\sum_{T_{ji} \in \vec{T}_j} L_{\text{total}}(T_{ji}, \Upsilon_{T_{ji}})$$

The function L_{total} is composed of two components:

$$L_{\text{total}}(T_{ji}, \Upsilon_{T_{ji}}) = L_{\text{network}}(T_{ji}, \Upsilon_{T_{ji}}) + L_{\text{data}}(T_{ji}, \Upsilon_{T_{ji}})$$

where L_{network} and L_{data} are called *network description length* and *data description length* respectively.

As described in [Lam and Bacchus, 1994], the network description length for a variable X_i is given as follows:

$$L_{\text{network}}(X_i, \Upsilon_{X_i}) = |\Upsilon_{X_i}| \log_2(n) + Q(s_i - 1) \prod_{j \in \Upsilon_{X_i}} s_j$$

where n is the number of variables. s_j is the number of possible states of variable X_j , and Q represents the number of bits required to store a numerical value.

This network description length has the property that the higher the topological complexity of the network the longer is its description length. It is widely recognized that conducting inference on highly-connected networks is likely to be intractable [Cooper, 1990], thus they are not very useful in practice. By minimizing this length function, we favor more useful networks, that is, those having simple topology.

For our categorization problem, there are $p + 1$ variables and each variable can take on binary values. Thus the network description length should be refined to:

$$L_{\text{network}}(T_{j_i}, \Upsilon_{T_{j_i}}) = |\Upsilon_{T_{j_i}}| \log_2(p + 1) + Q2^{|\Upsilon_{T_{j_i}}|}$$

Apart from the simplicity issue, we also need to consider the accuracy issue which refers to how accurate a network structure represents the training data. The accuracy issue is captured by the data description length as follows:

$$L_{\text{data}}(T_{j_i}, \Upsilon_{T_{j_i}}) = \sum_{T_{j_i}, \Upsilon_{T_{j_i}}} M(T_{j_i}, \Upsilon_{T_{j_i}}) \log_2 \frac{M(\Upsilon_{T_{j_i}})}{M(T_{j_i}, \Upsilon_{T_{j_i}})}$$

where the summation is taken over all possible instantiations of that variable and its parents. $M(\cdot)$ is the number of cases that match a particular instantiation in the training data. The log function will be 0 if $M(T_{j_i}) = 0$. It can be shown that this data description length has the property that the more accurate the network structure the shorter is the description length. By minimizing this length function, we favor accurate networks.

Simplicity and accuracy cannot usually be maximized simultaneously. Given a particular network structure, it will become more accurate if we increase its topological complexity appropriately. As a result, we are faced with a tradeoff between simplicity and accuracy. It is desirable to learn an accurate network, and at the same time the structure should be as simple as possible. The score metric L_{total} , based on the MDL measure, offers a principled means to perform such tradeoff by considering the sum of the network description length L_{network} and the data description length L_{data} (note that both L_{network} and L_{data} are non-negative).

Within the framework, a shorter length L_{total} corresponds to a better network. In this text categorization

problem, we would like to find a classification-based network which has the lowest L_{total} . In situations where an optimal solution cannot be obtained due to limited computing resources, we wish to find a network with L_{total} as low as possible. To find a good structure, we employ a best-first search strategy guided by the MDL metric. The heuristic function of a candidate network is basically composed of the MDL metric of the network structure and the arc to be added to the network. When we generate a candidate network structure, we impose a restriction that an arc cannot direct from a node representing a term to the node representing the category. Thus classification-based network structure topology can be maintained. This constraint increases the search efficiency because non-classification-based structures will be ignored during the search process.

5 Experimental Results

We have conducted experiments to demonstrate the performance of our approach. We implemented our approach as well as the Bayesian independence classifier approach. Thus we can directly compare the performance of both approaches. The document collection used in this experiment are economic and financial news articles from Reuters. There are 22,173 full text documents in the corpus. We used 21,450 documents between Feb. 26 1987 and Oct. 20 1987 in our experiment. These texts vary in length from single line bulletins to multi-page articles. Each document is assigned to one or more categories of economic interest. We partitioned the corpus into training and testing collection in the same way as the experiment done in [Lewis, 1992]. The texts before Apr. 7 1987 belongs to the training collection while those after this date belongs to the testing collection. There are 14,704 and 6,746 documents in the training and testing collection respectively. There are 112 categories in the training collection and the performance evaluation is based on these categories.

For each category, we select features based on the mutual information gain of the candidate feature and the category. The mutual information of a term T_i and category C_j is defined as:

$$\sum_{t \in \{0,1\}} \sum_{c \in \{0,1\}} P(T_i = t, C_j = c) \log_2 \frac{P(T_i = t, C_j = c)}{P(T_i = t)P(C_j = c)}$$

After mutual information of each term is calculated, the terms are ranked in descending order of mutual information and the top p terms are chosen as the predictor features for the category.

For the evaluation of our categorization system, the performance of each category is measured. The overall effectiveness is then computed by the mean performance across each category. Consider a particular category,

category	Bayesian indep. approach	Bayesian network approach	category	Bayesian indep. approach	Bayesian network approach	category	Bayesian indep. approach	Bayesian network approach
fishmeal	1.000	1.000	jobs	0.636	0.714	aust. dlr	0.000	1.000
mpiah	1.000	1.000	rice	0.409	0.794	coffee	0.651	0.889
lit	1.000	1.000	trade	0.559	0.378	wool	1.000	1.000
castor seed	1.000	1.000	lin meal	1.000	1.000	grain	0.760	0.788
ringgit	1.000	1.000	sun meal	0.000	0.000	nickel	0.800	1.000
pork belly	0.000	1.000	fuel	0.000	0.353	wheat	0.652	0.879
corn gluten feed	1.000	1.000	nat gas	0.328	0.333	soy oil	0.316	0.846
tapioca	0.000	1.000	crude oil	0.677	0.673	iron steel	0.171	0.400
rand	0.000	0.000	carcass	0.476	0.642	soy meal	0.588	0.696
nzdrlr	0.000	0.000	groundnut	0.000	0.000	ship	0.769	0.726
jet	0.000	0.000	linseed	1.000	1.000	pet. chem	0.455	0.400
peseta	1.000	1.000	yen	0.364	0.517	cocoa	0.591	0.900
cpu	0.000	0.000	sorghum	0.522	0.774	orange	0.778	0.769
inventories	0.000	0.000	strategic metal	0.111	0.105	tin	0.167	0.914
corn oil	1.000	1.000	dkr	1.000	1.000	lei	0.750	0.800
castor oil	0.000	0.000	nkr	0.000	0.000	acq.	0.691	0.610
naphtha	0.000	0.400	skr	1.000	1.000	money supply	0.300	0.330
dfl	0.000	0.000	stg	0.000	0.000	veg. oil	0.810	0.684
copra cake	0.000	0.000	cotton	0.513	0.898	reserves	0.512	0.449
red bean	1.000	1.000	sun seed	0.545	0.667	oil seed	0.514	0.696
rye	0.000	0.000	copper	0.488	0.760	tea	0.250	0.667
housing	0.364	0.571	dmk	0.000	0.000	palm kernel	0.000	0.000
cotton oil	0.000	0.000	soybean	0.472	0.613	zinc	0.455	0.722
oat	0.500	0.667	lead	0.381	0.462	palm oil	0.960	0.615
potato	0.400	0.857	livestock	0.508	0.519	alum	0.615	0.686
sun oil	0.000	0.000	hog	0.286	0.316	gnp	0.612	0.634
rape oil	0.400	0.000	cpi	0.393	0.273	platinum	0.200	0.000
groundnut oil	0.000	0.000	rape seed	0.500	0.762	interest	0.472	0.289
lin oil	0.000	0.000	corn	0.611	0.894	ipi	0.516	0.345
palladium	0.000	0.000	barley	0.500	0.941	rubber	0.516	0.638
l. cattle	0.000	0.000	bop	0.746	0.513	retail	0.333	0.333
lumber	0.600	0.727	gas	0.400	0.571	money fx.	0.446	0.336
can	0.000	1.000	heat	0.615	0.533	income	0.167	0.222
cruzade	1.000	1.000	meal feed	0.667	0.750	dlr	0.526	0.231
instal debt	0.667	0.000	silver	0.118	0.125	wpi	0.222	0.462
propane	0.500	0.667	coconut	0.667	0.571	earn	0.814	0.740
coconut oil	0.000	0.400	gold	0.579	0.500			
plywood	0.000	1.000	sugar	0.740	0.764			

Table 1: The performance of each category measured by F_1 measure in the Bayesian independence classifier approach and our Bayesian network approach

the effectiveness of the categorization can be shown in a contingency table as follows:

	Expert Says Yes	Expert Says No	
System Says Yes	q	r	$q + r$
System Says No	s	t	$s + t$
	$q + s$	$r + t$	$q + r + s + t$

Table 2: A Contingency Table Showing Categorization Decisions For a Category

where q is the number of documents belonging to the category and assigned to the category; r is the number of documents not belonging to the category and assigned to the category; s is the number of documents belonging to the category but not assigned to the category; t is the number of documents not belonging to the category and not assigned to the category.

Some common effectiveness measures can then be defined in terms of these values:

$$\begin{aligned} (\text{recall})R &= q/(q + s) \\ (\text{precision})S &= q/(q + r) \end{aligned}$$

Recall is the proportion of documents belonging to the category that the system successfully assigns to the category. Precision is the proportion of documents assigned to the category by the system that really belong to the category. An ideal categorization system would have recall and precision equal to 1. However, perfect recall can be achieved by a system that puts every document in the category, while perfect precision can be achieved by a system that puts no documents in the category, so just using either recall or precision does not provide a fair evaluation to a system.

Therefore, an effectiveness measure called F measure, proposed by [Van Rijsbergen, 1979; Lewis *et al.*, 1996], combines the recall and precision into a single score as follows:

$$F_\beta = \frac{(\beta^2 + 1)SR}{\beta^2 S + R}$$

β ranges from 0 to infinity. When β is 0, F_β is the same as the precision. When β equals to infinity, F_β is the same as recall. The variation of β between 0 and infinity corresponds to the variation of the weight to recall and precision. In our evaluation, β is set to 1 (i.e., equal weight on recall and precision). After we compute all F_1 values for each category, the mean F_1 value across categories is computed so as to compute the overall eval-

uation of our categorization system.

Table 1 shows the performance of the Bayesian independence classifier approach and our Bayesian network approach measured using F_1 for each category in the experiment. The mean F_1 measure of the Bayesian independence classifier is 0.443 whereas the mean F_1 measure of our Bayesian network induction approach is 0.542. Therefore, there is roughly a 22.3% improvement for our Bayesian network approach. There are 88 out of 112 categories where the performance of the Bayesian network approach is the same or better than the Bayesian independence classifier approach.

We also investigate the sensitivity of the probability threshold on the performance of the Bayesian network approach. Figure 2 shows the performance of the Bayesian network approach when we varied the probability threshold.

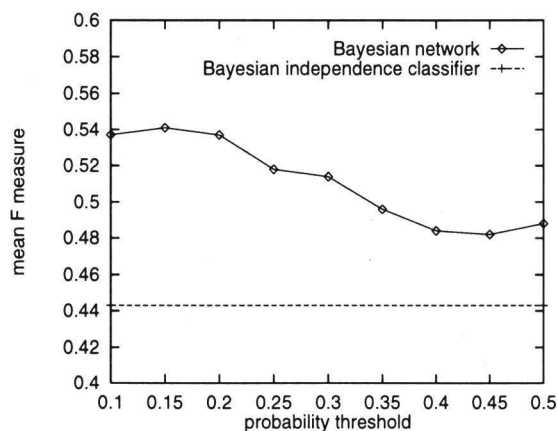


Figure 2: Sensitivity of Probability Threshold on the Bayesian Network Approach

6 Conclusion

Bayesian methods for automatic document categorization are investigated in this paper. We first reviewed an existing technique based on the Bayesian independence classifier approach. After discussing its drawbacks, we introduced a new approach based on Bayesian networks. The theoretical justifications of our new approach is also described. Experiments were conducted using a large document collection from Reuters news articles. The results show that our approach outperformed the Bayesian independence classifier as measured by a metric that combines precision and recall measures.

References

[Apte *et al.*, 1994] C. Apte, F. Damerau, and Weiss S.M. Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems*, 12(3):233–251, 1994.

[Cooper, 1990] G. F. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42:393–405, 1990.

[Friedman and Goldszmidt, 1996] N. Friedman and M. Goldszmidt. Building classifiers using Bayesian networks. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 1277–1284, 1996.

[Heckerman *et al.*, 1995] D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243, 1995.

[Hersh *et al.*, 1994] W. Hersh, C. Buckley, T.J. Leone, and D. Hickman. OHSUMED: An interactive retrieval evaluation and new large test collection for research. In *Proceedings of the Seventeenth International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 192–201, 1994.

[Lam and Bacchus, 1994] W. Lam and F. Bacchus. Learning Bayesian belief networks: An approach based on the MDL principle. *Computational Intelligence*, 10(3):269–293, 1994.

[Larkey and Croft, 1996] L. S. Larkey and W. B. Croft. Combining classifiers in text categorization. In *Proceedings of the Nineteenth International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 289–297, 1996.

[Lewis *et al.*, 1996] D. D. Lewis, R. E. Schapore, J. P. Callan, and R. Papka. Training algorithms for linear text classifiers. In *Proceedings of the Nineteenth International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 298–306, 1996.

[Lewis, 1992] D. D. Lewis. Feature selection and feature extraction for text categorization. In *Proceedings of Speech and Natural Language Workshop*, 1992.

[Spirtes and Meek, 1995] P. Spirtes and C. Meek. Learning Bayesian networks with discrete variables from data. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, pages 294–299, 1995.

[Van Rijsbergen, 1979] C. J. Van Rijsbergen. *Information Retrieval*. Butterworths, London, 1979.

[Yang, 1994] Y. Yang. Expert network: Effective and efficient learning from human decisions in text categorization and retrieval. In *Proceedings of the Seventeenth International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 13–22, 1994.