

Data Mining from Noisy Learners

John M. Pierre

Abstract

In this paper we discuss issues related to data mining from a noisy database such as what might be generated by a machine learning system. We describe an approach for estimating joint probability distributions of the noise-free case in terms of noisy observables and conditional probabilities which can be estimated using statistical sampling and error analysis. Several experiments are presented to test this approach.

Keywords: data mining, joint probability distributions, machine learning, error analysis

1 Introduction.

Traditional data mining includes a wide variety of tools and techniques used to analyze large databases to discover interesting and previously unknown knowledge embedded in the data[3]. Most approaches to data mining assume that a database exists with the appropriate data dimensions and relatively noise-free contents. Data cleaning is sometimes used to eliminate or correct a relatively small fraction of the data elements with missing values or those with easily detected inconsistencies. However, in some cases the database may contain data that is inherently noisy and not easily cleaned. A good example is a database containing concepts that have been generated from one or more machine learning algorithms. Since almost all approaches to machine learning imply some intrinsic error rate[11], such a database would contain a fair amount of noisy data. Therefore it is important to understand how data mining is affected by the presence of noise.

Many of the approaches to data mining can be cast in terms of applying techniques of unsupervised learning to large databases to discover knowledge, patterns, and rules of thumb. The general problem of unsupervised learning can be characterized in terms of inferring properties of joint probability distributions[5]. Consider the case of a random variable X and a sample of N observations (x_1, x_2, \dots, x_N) . The goal of unsupervised learning is to determine interesting properties of the underlying joint density $P(X)$. For example, cluster analysis attempts to discover if $P(X)$ can be cast in terms of a mixture of simpler underlying probability densities that

represent distinct classes among the observations. Association rules attempt to describe regions of high probability density corresponding to simple conjunctive rules.

In this paper we study the relationship between the joint probability distributions that are observed from the output of a noisy process versus the true underlying distribution in the noise-free environment. From this we hope to understand some general issues and limitations when mining knowledge from noisy data. We emphasize the case where the database has been created from a machine learning system. In section 2 we describe a general approach based on elementary probability and statistics in which we can estimate the true joint probability distribution based on observed quantities. We test this approach on some simulated and real world data in section 3. In section 4 we discuss related work, and we state our conclusions in section 5.

2 Models and Approach.

In this section we attempt to understand the relationship between probability distributions of database variables obtained from a noisy machine learning algorithm versus their underlying "true" values.

In our framework we assume the underlying data set is described by a random vector X drawn from a probability distribution \mathcal{D} . We also assume there is a set of functions $f_i(X)$ that map values of X to a set of higher level concepts $C_i = f_i(X)$, where each concept can take on values from a discrete set $C_i \in \mathcal{S}_i$. This defines the hypothetical *noise-free* case.

Let there be another set of functions and concepts $\tilde{C}_i = L_i(X)$ where $\tilde{C}_i \in \mathcal{S}_i$. The extent to which $L_i(X)$ and $f_i(X)$ differ characterizes the amount of noise in the system. The function $L_i(X)$ can be considered as describing a learned approximation to the target function $f_i(X)$ or a process that adds a certain amount of noise to the system (*i.e.* $L_i(f_i(X))$).

2.1 General Case. In the general case consider a conjunction of N discrete valued random concept variables $C \equiv (C_1 \wedge \dots \wedge C_N)$. Let the observables \tilde{C}_i take on values from a set $\tilde{c}_i \in \mathcal{S}_i$. Let \mathcal{P} be the set of permutations of all possible values of $\tilde{C}_i = \tilde{c}_i$ and let p be a single permutation, then we write $\tilde{C}(p) \equiv (\tilde{C}_1(p) \wedge \dots \wedge \tilde{C}_N(p))$.

By straightforward application of the theorem of total probability we have

$$(2.1) \quad P(C) = \sum_{p \in \mathcal{P}} P(C|\tilde{C}(p))P(\tilde{C}(p))$$

where $\sum_{p \in \mathcal{P}} P(\tilde{C}(p)) = \sum_{p \in \mathcal{P}} P(\tilde{C}_1(p) \wedge \dots \wedge \tilde{C}_N(p)) = 1$ since each permutation is a mutually exclusive outcome.

Our general approach is to estimate the conditional probabilities based on a representative sample of the database using a suitable statistical method (*i.e.* cross-validation, bootstrap, *etc.*). For example, if the database is derived from a machine learning algorithm these quantities can be estimated using cross-validation on the training data. By definition, the training data contains a sample of the “true” values of C across a number of examples. By comparing the output of the learning algorithm on the training data we can create a contingency table to observe the relative frequencies of C and $\tilde{C}(p)$ for each permutation p . By averaging the results obtained from the contingency table created for each training-test sample we can derive estimates of the conditional probabilities which we write as

$$(2.2) \quad P(C|\tilde{C}(p)) \simeq \mu(C, \tilde{C}(p)).$$

The joint probabilities of the observable variables are then determined from the output of the learning algorithm applied to the rest of the database. We write these estimates as

$$(2.3) \quad P(\tilde{C}(p)) \simeq \mu(\tilde{C}(p)).$$

Therefore we can derive an estimate for $P(C) \simeq P_{est}(C)$ in terms of sampled quantities,

$$(2.4) \quad P_{est}(C) = \sum_{p \in \mathcal{P}} \mu(C, \tilde{C}(p))\mu(\tilde{C}(p))$$

Because these estimates are determined from a finite sample of the database it is appropriate to consider the variance of the estimates. Using standard formulas for propagation of errors we can write the variance $\sigma^2(C)$ in our calculated value of $P_{est}(C)$ in terms of observed sample variances,

$$(2.5) \quad \sigma^2(C) = \sum_{p \in \mathcal{P}} (\sigma^2(C, p)\mu^2(\tilde{C}(p)) + \sigma^2(\tilde{C}(p))\mu^2(C, \tilde{C}(p))),$$

where $\sigma^2(C, \tilde{C}(p))$, $\sigma^2(\tilde{C}(p))$ are observed sample variances in $\mu(C, \tilde{C}(p))$, $\mu(\tilde{C}(p))$ respectively. Here we have

kept only leading terms under the standard assumption that cross terms will tend to cancel out[2].

The above analysis only has practical value to the extent that the conditional probabilities can be estimated accurately. Because these estimates require a good sample of observations that include both the noisy values \tilde{C}_i as well as the noise-free values C_i , this can be a problem. In the rest of the paper we present evidence that for small values of N , these quantities can indeed be estimated at an acceptable level of accuracy by sampling a small subset of the total database.

Using computational learning theory we can compute an upper bound on the number of examples needed to achieve a given level of accuracy in our estimates of the conditional probabilities. Assuming that each training example can be considered as an independent Bernoulli trial, we can write down the Chernoff bound for the probability that an estimate $\mu(C, \tilde{C}(p))$ differs from the true conditional probability $P(C|\tilde{C}(p))$ by a fixed amount ϵ ,

$$(2.6) \quad Pr[|\mu(C, \tilde{C}(p)) - P(C|\tilde{C}(p))| \geq \epsilon] \leq 2e^{-2m\epsilon^2}$$

where m is the number of examples that are used to estimate $\mu(C, p)$. Because the total number of permutations is given by $|\mathcal{P}| = \prod_{i=1}^N |\mathcal{S}_i|$ the probability that any of the estimates differs from its true value by more than ϵ is

$$(2.7) \quad \begin{aligned} \delta &\equiv Pr[(\exists p \in \mathcal{P})(|\mu(C, p) - P(C|\tilde{C}(p))| \geq \epsilon)] \\ &\leq 2\left(\prod_{i=1}^N |\mathcal{S}_i|\right)e^{-2m\epsilon^2}. \end{aligned}$$

Then the number of examples we need to hold δ some fixed value is bounded by

$$(2.8) \quad m \geq \frac{1}{2\epsilon^2} \left[\ln \frac{1}{\delta} + \ln 2 + \sum_{i=1}^N \ln |\mathcal{S}_i| \right].$$

In the case of binary valued variables $|\mathcal{S}| = 2$, so we have,

$$(2.9) \quad m \geq \frac{1}{2\epsilon^2} \left[\ln \frac{1}{\delta} + (N+1)\ln 2 \right].$$

Therefore the number of training examples grows linearly with the number of variables in our conjunctive rules. For example, if we desire a 95% probability that every estimated conditional is accurate to within 0.1 we

have

$$\begin{aligned} m &\geq \frac{1}{2(0.1)^2} \left[\ln \frac{1}{0.05} + (N+1) \ln 2 \right] \\ &\approx 50[3.7 + 0.7N] \end{aligned}$$

and therefore, for small N , we need at least a couple hundred training examples.

2.2 Example 1. As a simple example, we study the case of a single binary random variable $C \in \{0, 1\}$ which is transformed into another variable \tilde{C} by a noisy learner $\tilde{C} = L(X)$. Let there be a large number of examples $X \in D$ in our database sample D drawn from \mathcal{D} . Let us partition D into two disjoint sets $D = \mathcal{T} \cup \mathcal{U}$ where $\mathcal{T} \cap \mathcal{U} = \emptyset$. The set \mathcal{T} serves as our training/test set and must be labeled with the correct values of $C = f(X)$ using some reliable (though not necessarily efficient) process. The set \mathcal{U} comprises the remaining database of unlabeled examples.

In this case (2.1) becomes

$$(2.10) \quad P(C) = P(C|\tilde{C})P(\tilde{C}) + P(C|\neg\tilde{C})P(\neg\tilde{C}).$$

The conditional probabilities are estimated during the training phase by evaluating the performance of the learner on the test set by comparing the agreement between $f(\forall X : X \in \mathcal{T})$ and $L(\forall X : X \in \mathcal{T})$. The frequency of each possible outcome can be represented in a contingency table as shown in Table 1.

Table 1: Contingency Table

	$C = 1$	$C = 0$
$\tilde{C} = 1$	a	b
$\tilde{C} = 0$	c	d

Then we can compute the estimated conditional probabilities based on observed frequencies,

$$(2.11) \quad \begin{aligned} P(C|\tilde{C}) &\simeq \mu(C, \tilde{C}) = \frac{a}{a+b} \\ P(C|\neg\tilde{C}) &\simeq \mu(C, \neg\tilde{C}) = \frac{c}{c+d}. \end{aligned}$$

By utilizing a cross-validation procedure we can estimate the sample means and standard deviations for the above quantities.¹

The probabilities $P(\tilde{C}) + P(\neg\tilde{C}) = 1$ are estimated by applying the learner to a large number of unlabeled

examples and observing the frequencies of occurrence for the values of $\tilde{C} = L(\forall X : X \in \mathcal{U})$.

In practice it should hopefully be the case that there are many fewer training examples than unlabeled examples, so $|\mathcal{U}| \gg |\mathcal{T}|$. Then we can assume that the statistical errors in our estimates for the conditional probabilities will dominate. Using (2.5) we can then approximate the total variance in our estimate for $P_{est}(C)$ as

$$(2.12) \quad \sigma^2(C) \simeq \sigma^2(C, \tilde{C})\mu^2(\tilde{C}) + \sigma^2(C, \neg\tilde{C})\mu^2(\neg\tilde{C}).$$

2.3 Example 2. Now let us consider the case of two binary random variables A, B and their noisy counterparts \tilde{A}, \tilde{B} . We can write the “true” joint probability distribution in terms of observable quantities,

$$(2.13) \quad \begin{aligned} P(A \wedge B) &= P(A \wedge B|\tilde{A} \wedge \tilde{B})P(\tilde{A} \wedge \tilde{B}) \\ &\quad + P(A \wedge B|\tilde{A} \wedge \neg\tilde{B})P(\tilde{A} \wedge \neg\tilde{B}) \\ &\quad + P(A \wedge B|\neg\tilde{A} \wedge \tilde{B})P(\neg\tilde{A} \wedge \tilde{B}) \\ &\quad + P(A \wedge B|\neg\tilde{A} \wedge \neg\tilde{B})P(\neg\tilde{A} \wedge \neg\tilde{B}) \end{aligned}$$

The formalism proceeds very much the same as in Section 2.2. We estimate the conditional probabilities by evaluating the performance of the learner on the test examples and using a larger more complicated contingency table. The estimates for $P(\tilde{A} \wedge \tilde{B})$, etc. are obtained by applying the learner to the full database of unlabeled examples. The value of $P_{est}(A \wedge B)$ is determined from (2.4). Error estimates are obtained by application of (2.5).

Applying this method to cases with more random variables and with greater numbers of allowed discrete values is straightforward; however the computational complexity increases significantly with the total number of allowed permutations.

3 Experiments.

In this section we describe and present results for several experiments that we performed in order to test the feasibility and performance of our general approach. We performed two experiments with simulated data involving one and two random variables respectively. This allowed us to study in detail the relationship between sources of error and the resulting joint probability distributions. In the final experiment we applied a well-known machine learning algorithm to real-world data and compared results among the estimated, observed, and true joint probability distributions.

¹Note that $P(C|\tilde{C})$ is sometimes known as the *Precision* of the learner[8].

3.1 One Variable. In the first experiment we studied the effect of noise on the probability distribution of a single binary random variable. To generate the simulated data we applied Monte Carlo techniques to the following probability model. First we consider each example of a random variable C to be generated according to a Bernoulli model,

$$\begin{aligned} P(C = 1) &= p \\ P(C = 0) &= 1 - p. \end{aligned}$$

Then we apply noise by randomly flipping the value of C in each example according to the conditional probabilities

$$\begin{aligned} P(\tilde{C} = 1|C = 1) &= 1 - \epsilon \\ P(\tilde{C} = 0|C = 1) &= \epsilon \\ P(\tilde{C} = 0|C = 0) &= 1 - \eta \\ P(\tilde{C} = 1|C = 0) &= \eta \end{aligned}$$

where ϵ is the probability that a positive example gets flipped to negative and η is the probability that a negative example gets flipped to positive.

During the training phase we generated examples and compared the values of C versus \tilde{C} in order to build the contingency table described in Section 2.2. Several trial runs were performed to simulate a cross-validation process. Estimates for the conditional probabilities were obtained using (2.11).

During the evaluation phase, the above model was used to generate examples. From this data we observed the frequencies of \tilde{C} to determine estimates of $P(\tilde{C})$ and $P(-\tilde{C})$. The final estimates for $P_{est}(C)$ and error estimates we computed using Equations (2.10) and (2.12). We varied the values for p , ϵ , and η and compared the discrepancies between the true probability $P(C)$, the observed probability $P(\tilde{C})$, and the estimated probability $P_{est}(C)$.

A typical result is shown in Figure 1. In this case $p = 0.2$ and $\epsilon = \eta$. Fifty training examples were generated during a total of five independent trials. For the evaluation phase we generated 5000 examples. As can be seen from the figure, the model given in (2.10) was able to provide a more accurate estimate of the true probability (to within a standard deviation) across a wide range of error rates.

3.2 Two Variables. In the second experiment we studied the effect of noise on the joint probability distribution for two correlated binary random variables A and B . Again we used Monte Carlo techniques with a simple probability model where A was generated according to

$$P(A = 1) = p$$

$$P(A = 0) = 1 - p$$

and B generated using

$$\begin{aligned} P(B = 0|A = 0) &= 1 - q + \alpha q \\ P(B = 1|A = 0) &= q - \alpha q \\ P(B = 0|A = 1) &= 1 - q - \alpha(1 - q) \\ P(B = 1|A = 1) &= q + \alpha(1 - q) \end{aligned}$$

where α parametrizes the degree of correlation. If $\alpha = 0$ then B is independently distributed with $P(B = 1) = q$. If $\alpha = 1$ then B is completely dependent on A . As $0 \leq \alpha \leq 1$ increases then B becomes more correlated with A , and the true joint probability is given by

$$P(A \wedge B) = P(B|A)P(A) = (q + \alpha(1 - q))p.$$

Error is introduced to the system according to the conditional probabilities

$$\begin{aligned} P(\tilde{A} = 1|A = 1) &= 1 - \epsilon_A \\ P(\tilde{A} = 0|A = 1) &= \epsilon_A \\ P(\tilde{A} = 0|A = 0) &= 1 - \eta_A \\ P(\tilde{A} = 1|A = 0) &= \eta_A \end{aligned}$$

and,

$$\begin{aligned} P(\tilde{B} = 1|B = 1) &= 1 - \epsilon_B \\ P(\tilde{B} = 0|B = 1) &= \epsilon_B \\ P(\tilde{B} = 0|B = 0) &= 1 - \eta_B \\ P(\tilde{B} = 1|B = 0) &= \eta_B. \end{aligned}$$

Experiments were carried out according to a procedure similar to the one described in Section 3.1. The true joint probability was estimated using Equation (2.13) and results were obtained for different values of p , q , α , ϵ_A , η_A , ϵ_B , and η_B .

A typical result is shown in Figure 2. In this case $p = q = 0.2$, $\alpha = 0.5$ and $\epsilon_A = \eta_A = \epsilon_B = \eta_B$. Fifty training examples were generated during a total of five independent trials, and for the evaluation phase we generated 5000 examples. As can be seen from the figure, the model given in (2.4) was able to accurately estimate the true probability (to within a standard deviation) even for high error rates.

3.3 Real World Data In this experiment we generated noisy data using the C4.5 decision tree learning algorithm[13]. For raw data we used the ‘‘Adult Database’’[15] which includes 48,842 examples of individual’s Census Income data. We partitioned the full data set into 32,561 unlabeled examples for the evaluation phase and 16,281 labeled examples for the training

and testing phase. The 16,281 labeled examples were further partitioned for 3-fold cross-validation.

We trained two sets of decision trees to classify the raw examples into two types of classes,

$$\begin{aligned} A &\in \{Grade, HS, HS+, College, Graduate\} \\ B &\in \{\leq 50K, > 50K\} \end{aligned}$$

where the variable A represents the maximum attained education level and B represents the income level for each individual.² Our goal was to estimate the joint probability distribution $P(A \wedge B)$ for each pair $A \wedge B$.

At each step in the cross-validation procedure, 10,854 training examples were used to train an “A-type” and “B-type” decision tree learner, and 5,427 test examples were used to estimate the classifier accuracy and the conditional probabilities (*i.e.* $P(A \wedge B | \tilde{A} \wedge \tilde{B})$). For the “A-type” learner the classification error rate was approximately 50% while for the “B-type” learner it was about 16%.

During the evaluation procedure the two types of decision trees were used to classify the remaining 32,651 examples to generate joint probability estimates for the noisy observables $P(\tilde{A} \wedge \tilde{B})$.³ Using the formulas (2.4) (2.5) we obtained estimates for $P_{est}(A \wedge B)$ for each of the ten possible pairings of $A \wedge B$ (pairs are defined in Table 2).

In Figure 3 we show the various measured joint probabilities for each pair in this experiment. In most cases the computed value of $P_{est}(A \wedge B)$ has better agreement with the true value (to within about one standard deviation) than the observed value $P(\tilde{A} \wedge \tilde{B})$.

In Table 2 we show each pair of $A \wedge B$ along with the measured percentage error between the true and observed (or estimated) joint probabilities,

$$\begin{aligned} \%Err(obs) &= \frac{|P(A \wedge B) - P(\tilde{A} \wedge \tilde{B})|}{P(A \wedge B)} \\ \%Err(est) &= \frac{|P(A \wedge B) - P_{est}(A \wedge B)|}{P(A \wedge B)}. \end{aligned}$$

In most cases it can be seen that the estimate $P_{est}(A \wedge B)$ is much closer to the true value than the observed value $P(\tilde{A} \wedge \tilde{B})$, and the average error is clearly reduced by using our procedure.

4 Related Work.

The topic of supervised learning in the presence of noisy training data has been studied previously[7][6]. The

²We collapsed the 16 values for “education” in the original data into the five values shown.

³Since these examples also contained the correct labels we also used them to estimate the true joint probabilities for $P(A \wedge B)$.

general approach has typically been to assume some underlying model of noise and to study its general implications for learning using the *probably approximately correct* (PAC) learning model or other methods of computational learning theory. In this way it is possible to establish some theoretical bounds on such things as the maximum tolerable noise rate, or the minimum number of noisy training examples required to achieve a given error rate. In addition, algorithms for efficiently learning from noisy examples are described.

Mining rules from databases derived from machine learning algorithms has been discussed in several works, especially in relation to text data mining. In [12] a methodology is described for constructing a database using an information extraction learning system applied to collections of text documents, and the quality of discovered rules is evaluated using a type of self-consistency test. Loh *et al.*[9] use automated categorization to assign a collection of pre-defined concepts to a corpus of documents. Statistical techniques were then applied to the sets of assigned concepts to find associative rules and concept distributions. In [4] a database of company information is automatically constructed from a large collection web pages using a combination of techniques including custom wrappers, information extraction, and text categorization, and the issue of noisy data is discussed in qualitative terms. However the relationship between errors from the learning algorithms and their effect on the outcome of the data mining results has not been studied in any detail.

Different aspects of studying the quality of association rule mining from a probabilistic framework have been considered previously. In [14] a Bayesian framework is used to derive a relationship between support, confidence, and predictive accuracy. In [10] the predictive quality of association rules is studied by estimating the number of false discoveries. *p-values* are used to estimate the likelihood that a rule violates a null-hypothesis, and confidence intervals for the support and confidence are derived. In [16] statistical sampling is used to improve computation efficiency of mining association rules in very large databases, and rule accuracy is measured with respect to sampling strategy.

5 Conclusion.

In this paper we studied the relationship between the observed joint probability distributions obtained from noisy data and compared them to the “true” values that would be seen in a noise-free environment. We constructed a simple probabilistic model and a procedure for correcting the noisy observed values based on estimates of conditional probabilities that profile the effects of the noise. Using computational learning theory

Table 2: C4.5 Results

$(A \wedge B)$ Pair	A	B	%Err(obs)	%Err(est)
1	Grade	>50K	0.725	0.016
2	Graduate	≤50K	0.082	0.132
3	HS	>50K	0.187	0.042
4	Graduate	>50K	0.038	0.097
5	HS+	>50K	0.269	0.026
6	College	>50K	0.125	0.046
7	College	≤50K	0.146	0.019
8	Grade	≤50K	0.380	0.019
9	HS+	≤50K	0.109	0.037
10	HS	≤50K	0.452	0.009
Average Error			0.252	0.044

we established a theoretical bound on the number of examples needed to accurately estimate these conditional probabilities with confidence. Finally, we performed a series of computational experiments to test these ideas and demonstrate the effectiveness of our approach. It seems that in many situations it may be possible to analyze a small subset of the full database in order to evaluate the parameters of our model, and to thereby obtain significantly more accurate estimates of the joint probability distributions.

In this paper we concentrated on the output of a learner as the source of noise in the data. It would be interesting to study and compare the effects of other sources of noise in databases such as sensor error or statistical sampling error.

Estimating joint probabilities are an essential part of association rule mining[1]. It should be possible to apply the methods described in this paper to obtain more accurate association rules from a noisy database. In addition it would be interesting to address the issue computational efficiency for applying this approach to large databases with large numbers of items, and to test the performance on other types of real-world data.

References

- [1] R. Agrawal, T. Imielinski, and A. Swami. Mining Association Rules Between Sets of Items in Large Databases. In *ACM SIGMOD Intl. Conf. Management of Data*, 1993.
- [2] P. R. Bevington. *Data Reduction and Error Analysis for the Physical Sciences*. McGraw-Hill, New York, 1969.
- [3] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy (Eds.). *Advances in Knowledge Discovery and Data Mining*. MIT Press, Cambridge, 1996.
- [4] R. Ghani, R. Jones, D. Mladenic, K. Nigam, and S. Slattery. Data Mining on Symbolic Knowledge Extracted from the Web. In *Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining (KDD-2000) Workshop on Text Mining*, 29-36, 2000.
- [5] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, New York, 2001.
- [6] M. J. Kearns and U. V. Vazirani. *Introduction to Computational Learning Theory*. MIT Press, Cambridge, 1994.
- [7] P. E. Laird. *Learning from Good and Bad Data*. Kluwer Academic, Boston, 1988.
- [8] D. Lewis. Evaluating Text Categorization. In *Proceedings of the Speech and Natural Language Workshop*, 312-318, 1991.
- [9] S. Loh, L. Wives, J. P. M. de Oliveira. Concept-based Knowledge Discovery in Texts Extracted from the Web. In *SIGKDD Explorations*, 2(1): 29-39, 2000.
- [10] N. Megiddo and R. Srikant. Discovering Predictive Association Rules.
- [11] T. M. Mitchell. *Machine Learning*. WCB/McGraw-Hill, Boston, 1997.
- [12] U. Nahm and R. Mooney. Text Mining with Information Extraction. In *Proceedings of the AAAI 2002 Spring Symposium on Mining Answers from Texts and Knowledge Bases*, 2002.
- [13] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, 1993.
- [14] T. Scheffer. Finding Association Rules that Trade Support Optimally Against Confidence. In *Proceedings of the European Conference of Principles and Practice of Knowledge Discovery in Databases (PKDD-01)*, 2001.
- [15] UCI Machine Learning Repository. <ftp://ftp.ics.uci.edu/pub/machine-learning-databases/>
- [16] M. J. Zaki, S. Parthasarathy, W. Li, M. Ogihara. Evaluation of Sampling for Data Mining of Association

Rules.

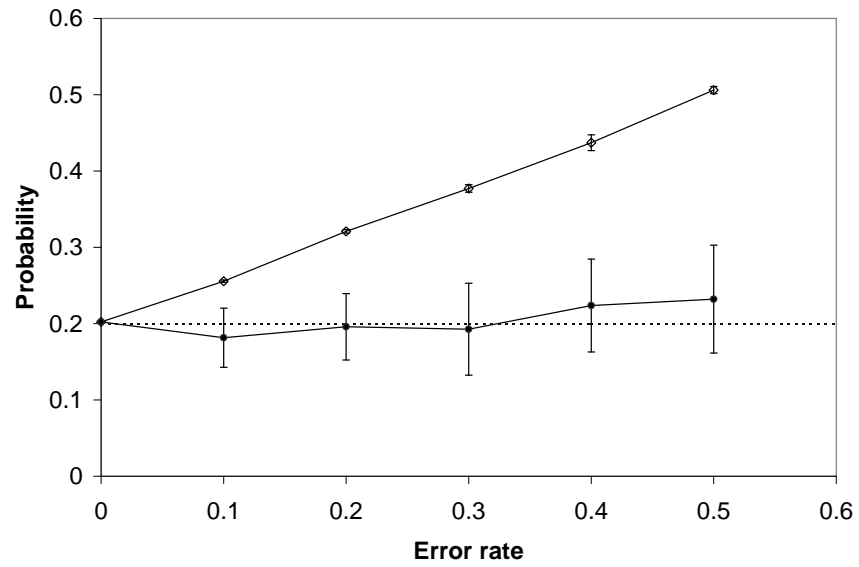


Figure 1: Probability estimates for $P_{est}(C)$ (marked by \diamond) and $P(\tilde{C})$ (marked by \bullet) as a function of error rate. The true probability $P(C) = 0.2$ is marked with a dotted line.

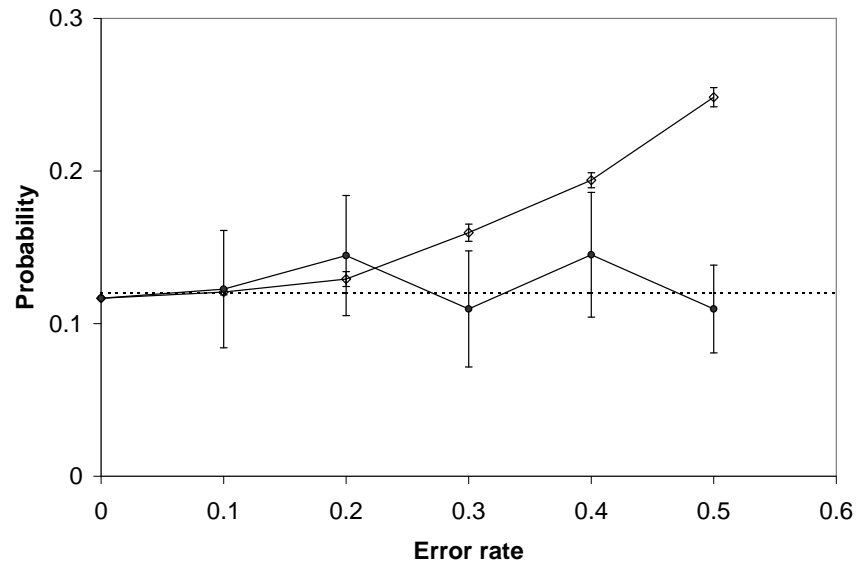


Figure 2: Probability estimates for $P_{est}(A \wedge B)$ (marked by ●) and $P(\tilde{A} \wedge \tilde{B})$ (marked by ◇) as a function of error rate. The true probability $P(A \wedge B) = 0.12$ is marked with a dotted line.

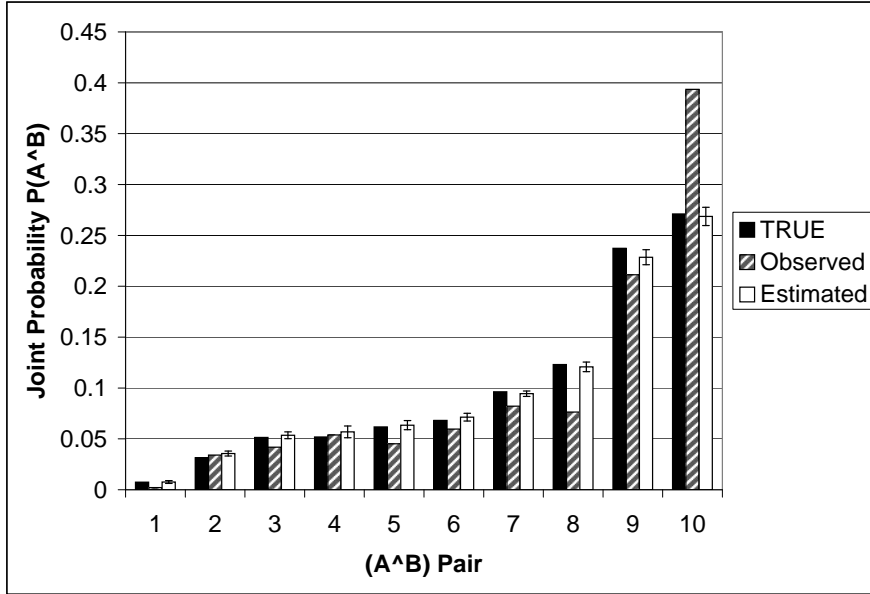


Figure 3: Results for data generated using decision trees on Census data. Probability estimates for $P_{est}(A \wedge B)$ (white bar with error estimates), $P(\tilde{A} \wedge \tilde{B})$ (striped bars), and true probability $P(A \wedge B)$ (black bar). Description of $(A \wedge B)$ pairs is given in Table 2.