Magical Thinking in Data Mining: Lessons From CoIL Challenge 2000

Charles Elkan
Department of Computer Science and Engineering 0114
University of California, San Diego
La Jolla, California 92093-0114
elkan@cs.ucsd.edu

ABSTRACT

CoIL challenge 2000 was a supervised learning contest that attracted 43 entries. The authors of 29 entries later wrote explanations of their work. This paper discusses these reports and reaches three main conclusions. First, naive Bayesian classifiers remain competitive in practice: they were used by both the winning entry and the next best entry. Second, identifying feature interactions correctly is important for maximizing predictive accuracy: this was the difference between the winning classifier and all others. Third and most important, too many researchers and practitioners in data mining do not appreciate properly the issue of statistical significance and the danger of overfitting. Given a dataset such as the one for the CoIL contest, it is pointless to apply a very complicated learning algorithm, or to perform a very time-consuming model search. In either case, one is likely to overfit the training data and to fool oneself in estimating predictive accuracy and in discovering useful correlations.

1. INTRODUCTION

This paper explains some general guidelines and principles that are fundamental to practical success in data mining, but which many practitioners do not follow or do not know. The empirical evidence that data mining practitioners are unaware of these lessons comes from the reports written by participants in a data mining contest organized in the spring of 2000. This contest, known as CoIL Challenge 2000, was sponsored by a consortium of research groups funded by the European Union. The contest attracted 147 registered participants, of whom 43 submitted entries. Of the 43 participants who submitted entries, 29 later wrote reports explaining their methods and results. The authors of these reports appear to be data mining practitioners or researchers, as opposed to students. The reports have been published by [8].

The CoIL contest was quite similar to the competitions organized in conjunction with the KDD conference in recent years, and to other data mining competitions. The contest task was to learn a classifier capable of identifying which customers of a Dutch insurance company have an insurance policy covering a caravan. (Caravans are mobile homes that are normally towed by cars. They are called trailers in North America.) The training set contained information on 5822 customers, of which 348, about 6%, had caravan policies. The test set contained information on 4000 customers randomly drawn from the same population. For each customer, the values of 85 features were given. Contest participants were asked to identify which 800 of the test customers were most likely to have caravan policies. The training and test sets used in the contest are now available in the UCI repository.

Like other data mining contests, the CoIL contest was a valuable testbed for measuring the “end to end” successfulness of data mining methods. The real-world usefulness of an algorithm depends not only on its theoretical properties, but also on the ease with which practitioners can apply it without falling victim to overfitting and other traps. In principle overfitting can be avoided with almost all methods, but in practice some methods are much more likely to lead their users astray. Contests such as this one provide guidance about which methods are *de facto* more robust, and about where data mining practitioners need guidance most.

2. THE WINNING ENTRY

Figure 1, adapted from [8], shows a histogram of the scores achieved by the 43 individuals or teams that submitted entries to the contest. The winning entry, which was mine, identified 121 caravan policy holders among its 800 top predictions. The next best methods identified 115 and 112 policy holders. The mean score was 95.4, with a standard deviation of 19.4. The distribution of scores is clearly not normal, with a long tail to the left. The most common score was 109, and the median score was 103.

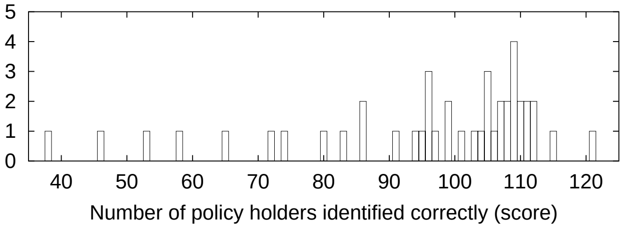

Figure 1: Distribution of scores achieved by 43 entries.

The data mining algorithm used in the winning entry was standard naive Bayesian learning. The second best entry, due to Petri Kontkanen, also used a naive Bayesian classifier. Many of the methods described in the 29 reports by participants are much more sophisticated. They include combinations of backpropagation neural networks, self-organizing maps (SOMs), evolutionary algorithms, C4.5, CART, and other decision tree induction algorithms, fuzzy clustering and rule discovery, support vector machines (SVMs), logistic regression, boosting and bagging, and more. The strong performance of naive Bayesian learning against this competition is noteworthy.

An important aspect of the CoIL contest was that the values of all numerical features were made discrete in advance by the contest organizers. For example, instead of a real-valued feature giving the precise monetary amount that a customer pays for car insurance, the CoIL datasets include only a discrete-valued feature that categorizes this amount into one of seven different discrete levels. Because all features were discretized in advance, the CoIL competition could not serve as a test of discretization methods.

The predictive accuracy of a naive Bayesian classifier can often be improved by boosting [3], and by adding new attributes derived from combinations of existing attributes. Both boosting and derived attributes are ways of relaxing the conditional independence assumptions that constitute the naive Bayes model. For the CoIL contest, after two important derived attributes were added, boosting did not give any significant increase in accuracy on a validation set taken from the training set, nor did adding more derived attributes. Therefore, the final entry I submitted did not use boosting, and used only two derived attributes.

The two derived attributes that I added give the most detailed information possible about a customer's existing car and fire insurance policies. The set of values of one new attribute is the cross-product of the sets of values of the existing attributes named PPERSAUT and APERSAUT. These attributes are explained as "contribution [level for] car policies" and "number of car policies." The other new attribute is a similar cross-product for two fire insurance attributes.

Creating a derived attribute that is a cross-product allows the naive Bayesian classifier to associate a different probability of having a caravan policy with each alternative pair of values for the attributes PPERSAUT and APERSAUT, and similarly for the two fire insurance attributes. As mentioned above, all attributes are already discretized into a small number of so-called levels, so each cross-product attribute also has a finite number of values. Any alternative derived attribute constructed from two attributes that have already been made discrete, for example an average policy premium defined as "contribution level" divided by "number of policies," must lose information compared to a cross-product attribute.

The strongest single predictor of having a caravan insurance policy is having two car insurance policies, or having one car policy where the contribution level is not low (not level 5). The other predictors that are most statistically significant are (i) a high value for "purchasing power class," specifically 5 or higher and especially 7 or higher, (ii) having a private third party insurance policy, (iii) having a boat insurance policy, (iv) having a social security insurance policy, and (v) having a single fire policy with a high contribution (level 4). Intuitively, these predictors identify customers who have a car and are wealthier than average, and who in general carry more insurance coverage than average. It is not surprising that these are the people who are most likely to have caravan insurance.

Statistical significance is easy to evaluate quantitatively but approximately for findings like the ones just stated. For example, 2977 customers in the training set have a car insurance policy. Of these, 276 have a caravan policy, that is 9.3% compared to 6% in the population of all customers. The number of customers with a car insurance contribution level of 5, all of which have only one car policy, is 613. If having a contribution level of 5 was not correlated with having a caravan policy, we would expect 9.3%, that is 57, of these customers to have a caravan policy, with a standard deviation of $\sqrt{57 \cdot (1-0.093)} = 7.2$. In fact, among customers with a car insurance contribution level of 5, only 14 have a caravan policy. The z-score of this discrepancy is $(14-57)/7.2 = -6.0$ standard deviations, which is unquestionably statistically significant.

The CoIL contest organizers explicitly set two tasks: develop a model of which customers are most likely to have a caravan policy, and separately, provide insight into the characteristics of these customers. Many data mining projects similarly have two useful deliverables: a predictive model, and new insights into the phenomenon being modeled. An insight is particularly useful if (a) it is statistically reliable, (b) it was not known previously, and (c) it is actionable, meaning that some action can be taken to exploit the insight. In many data mining projects, few insights satisfy all three of these criteria.

For the CoIL contest, the discovery that a customer who has a car insurance policy but whose premium amount is low is less likely than average to have caravan insurance is an insight that possesses the three properties. This finding was not obvious in advance, the argument above shows that it is statistically reliable, and it is actionable, since different promotions can be targeted specifically at these customers. A customer of this type is perhaps less wealthy or less risk-averse, so he or she is less likely to own a caravan, or less likely to buy insurance for it if he or she does own one.

It is not clear whether any additional insights can be gained from the CoIL training data that meet all three criteria explained above. Using algorithms to discover association rules, several participants enumerated additional claimed correlations, but they never measured the statistical significance of these correlations. The results of the contest indicate that none of these correlations covered enough customers or were reliable enough to improve overall predictive accuracy.

In addition to revealing reliable correlations that have predictive value, statistical significance testing can also be useful in showing that plausible potential correlations are in fact unproven. For example, in commercial data mining, significance testing often shows that demographic attributes, such as customer segmentations by lifestyle, income, etc., do not add any extra predictive power when behavioral data is available. This phenomenon is clearly visible in the CoIL dataset. When a feature provides no additional predictive value, it is generally beneficial not to use it in modeling. The naive Bayes model used for the winning entry to the CoIL contest was trained with all demographic attributes discarded, except the attribute named MKOOPKLA, "purchasing power class."

3. OVERFITTING

It is vital for data mining practitioners to understand the issue of overfitting in an intuitive way, and also to understand basic statistics in an intuitive way. Most introductory courses on statistics do not discuss overfitting explicitly, but overfitting is a concern even with the simplest data analysis procedures. Consider for example the mean and standard deviation of a sample of n observations x_i randomly selected from some parent population. The mean of the sample is defined of course as $\mu = \frac{1}{n}\sum_{x=1}^{n} x_i$. The standard deviation of the sample is the square root of the average squared deviation from the mean, that is the square root of $\frac{1}{n}\sum_{x=1}^{n}(x_i-\mu)^2$. But statistical textbooks always say that the standard deviation of the parent population should be estimated using the denominator $n-1$

instead of n. We should use the square root of $\frac{1}{n-1}\sum_{x=1}^{n}(x_i-\mu)^2$, which is always bigger since $\frac{1}{n-1} > \frac{1}{n}$ if $n \geq 2$. Why?

The reason is that using a denominator of n overfits the training data, the sample x_1 through x_n. The mean μ is the number relative to which the sum of squared deviations $\sum_{x=1}^{n}(x_i-\mu)^2$ is minimized. Clearly the mean of the parent population is in general not exactly the same as the mean of the sample. Therefore the average deviation of the sample relative to the unknown true mean of the population is typically greater than the square root of $\frac{1}{n}\sum_{x=1}^{n}(x_i-\mu)^2$. One can prove mathematically that replacing $\frac{1}{n}$ by $\frac{1}{n-1}$ yields an improved estimate of the true standard deviation of the parent population that will not systematically be too small. Technically, using $\frac{1}{n-1}$ gives the minimum variance unbiased estimator [7].

Many contest submissions reveal basic misunderstandings about the issue of overfitting. For example, one team wrote that they used " ... evolutionary search for choosing the predictive features. The result is a predictive model that uses only a subset of the original features, thus simplifying the model and reducing the risk of overfitting while maintaining accuracy."[1] As shown by the discussion above of alternative standard deviation estimators, overfitting is not prevented simply by choosing a model that is simple syntactically. Overfitting occurs when a model is chosen whose high apparent accuracy arises from fitting patterns in the training set that are not statistically reliable, and are not valid in test data.

Even if a predictive model uses only a small subset of features, if that subset is chosen as the best, or one of the near best, among many candidate subsets of features, then overfitting is a real danger. In data mining, overfitting is typically caused by searching a space of candidate models that is too large. Making predictions with a final model that uses only a few features is neither necessary nor sufficient to prevent overfitting. If this final model is found by a protracted search process that explicitly or implicitly considers many alternative models with different subsets of features, then the apparent simplicity of the final model is misleading. Similarly, the danger of overfitting is not reduced just because a learning algorithm outputs a model with only a few parameters. For the risk of overfitting to be lessened, all models that the algorithm could have output must be equally simple.

A few participants were quite unaware of the danger of overfitting. For example, after using backpropagation networks with 20 hidden units but with subsampled training sets of only 1200 examples, one team wrote "We haven't got the best score. The possible ways for the improvement of the model are (i) increase of the number of neurons in the hidden layer what let us to consider a greater number of the relationships among the data ... (ii) use of the genetic algorithm for the neural network learning since one is more effective in global extremum finding." These suggestions would only increase the severity of overfitting. What one should do instead is use all available training examples and apply a regularization method such as early stopping to reduce the risk of overfitting.

4. FEATURE SELECTION

Most participants used heuristics to identify which of the 85 available features have the most predictive value, and to choose a subset of features for input into their learning algorithm. None of the authors of reports applied a well-defined algorithm for feature selection that did not require human guidance. Moreover, none of the authors, including myself, used a systematic method to detect important feature interactions.

Some participants used methods designed to evaluate subsets of features, as opposed to individual features, but these methods failed to detect the interaction between the number of car policies and the total premium paid for car policies, because these two features are highly correlated. For example, one author incorrectly wrote "if 'Contribution car policies' is chosen, the information contained in 'Number of car policies' is already included in the model." Similarly, despite using a "multidimensional visualisation tool," another team also excluded the "number of car policies" feature after including the "contribution car policies" feature. While the number of car policies and the total premium paid for car policies are correlated, the features together do provide more information than either by itself.

A different team wrote that in the output of their commercial tool, "some contribution[s] may be related to 2 or more variables. In this last case, the contribution expresses the fact that the important information is brought by the 'additional knowledge' brought by the second variable when the first one is already known." Their tool correctly identified the "number of car policies" feature as highly predictive, but it did not detect the additional predictiveness of the "contribution car policies" feature or the interaction between these two features. Identifying this interaction is important for two reasons. First, exploiting it gives better predictive accuracy. Second, as discussed in Section 2, it is the only pattern in the training data that has been shown to be statistically reliable *and* surprising *and* actionable.

5. COMPARING THE ACCURACY OF LEARNING METHODS

A simple calculation shows that the test dataset provided in the CoIL contest is too small to reveal reliably which learning methods are more accurate. Consider the null hypothesis that a learning method achieves $p = 12\%$ accuracy in the top 20% of its predictions. On a randomly chosen test set of 4000 examples, the expected number of correct predictions is $\mu = pn = 96$ where $n = 0.2 \cdot 4000 = 800$, which is similar to the average number of correct predictions achieved by CoIL contest participants. The anticipated standard deviation of the number of correct predictions is $\sigma = \sqrt{np(1-p)} = 9.2$. In order for us to reject the null hypothesis with a confidence of over 95%, a learning method would have to score more than two standard deviations above or below the expected number, that is less than 78 correct or more than 114 correct. Any method that scores between 78 and 114 correct is not statistically distinguishable from the null hypothesis method. Only the winning entry and possibly the second best entry are significantly better than the average CoIL contest entry.

It is possible to compare two different learning methods with the same training and test datasets in a way that is more sensitive than the simple binomial calculation above, using McNemar's hypothesis test [2]. Let A and B designate two learning algorithms and let n_{10} be the number of test examples classified correctly by A but incorrectly by B. Similarly, let n_{01} be the number classified incorrectly by A but correctly by B. McNemar's test is based on calculating the statistic

$$s = \frac{(|n_{01}-n_{10}|-1)^2}{n_{01}+n_{10}}.$$

[1] Unless stated otherwise, all quotations in this paper are taken from reports written by CoIL contest participants and published by [8]. The quotations have the original spelling, grammar, and italics of their authors. The point of this paper is to explain some widespread misunderstandings, not to criticize individuals, so the names of the authors of quotations are omitted.

The null hypothesis is that both learning algorithms are equally accurate. If this hypothesis is true, then the numerator of the test statistic should be small. One can prove that under the null hypothesis, s has approximately a chi-squared distribution with one degree of freedom. The term -1 in the numerator is called a continuity correction and adjusts for the fact that s is discrete while the chi-squared distribution is continuous. In general, a chi-squared distribution with k degrees of freedom has mean k and variance $2k$. Therefore the distribution of s, under the null hypothesis, has mean 1 and standard deviation $\sqrt{2}$ approximately.

The numbers n_{01} and n_{10} needed to apply McNemar's test have not been published for most pairs of entries to the CoIL contest. However this information is available for the top entry and the second-best entry. As mentioned earlier, the contest task was to identify the 800 test examples most likely to be positive. Each entry can be viewed as labeling 800 examples as positive, and the remainder as negative. On the test dataset, the top entry A and the next entry B gave 729 predictions in common, of which 110 were true positives while 619 were false positives. Each entry gave 71 predictions that the other did not. Of the predictions unique to A, 11 were true positives and 60 were false positives. Of those unique to B, 5 were true positives and 66 were false positives. We have here that $n_{10} = 11 + 66$ while $n_{01} = 5 + 60$, so

$$s = \frac{(|65 - 77| - 1)^2}{65 + 77} = \frac{121}{142} < 1.$$

According to McNemar's test, the difference in accuracy between methods A and B is not statistically significant.

Even when participants did evaluate the performance of their models statistically, the evaluation was often inappropriate. For example, one team wrote " ... we have repeated the modelling process five times, finding ... 105 ± 2.55 in terms of mean and standard deviation." But a standard deviation computed on a fixed test set, over different runs of a learning algorithm, reveals only how stable a learning method is. It has no connection with the reproducibility of the generalization performance of the algorithm, which would be revealed by a standard deviation computed over independent test sets.

If one does not take a statistical view of the prediction task, then it is easy to believe incorrectly that a sufficient amount of training data is available. For example, one author wrote "This appears to be a complex, data rich, theory poor type of problem with substantial historical data ... " In fact, the number of positive training examples available, 348, is very small. Complex models cannot be learned with confidence from so few data points.

The mistake of believing that the number of training examples is large is connected with another common mistake, that of underestimating the knowledge and skill of existing experts in a particular domain. For example, the knowledge of experts in the insurance industry is derived informally from millions of training examples, and so is potentially much deeper than any knowledge that can be extracted from a single dataset. Of course, both the real world and the knowledge of experts are subject to uncertainty and ignorance. In the CoIL training dataset 2.5% of customers who do not have a car insurance policy nevertheless have a caravan policy. Presumably there is an explanation for this phenomenon.

6. MAGICAL THINKING

The previous sections of this paper have shown that many CoIL contest participants were misled by an insufficient intuitive understanding of the issue of statistical randomness. Many reports written by participants reveal an implicit belief that somehow one particular learning method is ideal, and with a little more intelligence, or luck, or data, a person could have discovered this method. For example, one author wrote "But after the announcement of the true policy owners it has become clear that the application of neural networks isn't necessary, [my] scoring system itself could be used instead and, even more, would have performed better." Later he wrote "Unfortunately I have discovered this fact *after* the announcement of the true policy owners."

This type of thought process is called "magical thinking" by anthropologists. In the words of a famous passage by Gregory Bateson, "The practitioner of magic does not unlearn his magical view of events when the magic does not work. In fact, the propositions which govern punctuation have the general characteristic of being self-validating" [1]. In any culture, humans have a certain set of expectations that they use to explain the results of their actions. When something surprising happens, rather than question the expectations, people typically believe that they should have done something slightly different. Unless people understand the issue of randomness and statistical significance in an intuitive way, they are liable to believe always that if only they had done something differently in the model building process, their model would have performed better.

In the quotation above from Bateson, the word "punctuation" means the way in which we find patterns in our perceptions of our experiences. The "propositions" that govern punctuation are the principles or expectations that we have learned to use to organize our perceptions. Whatever these principles are, if we have learned them, it is because they appeared to be useful in the past. As pointed out in a different context by Thomas Kuhn [6], practitioners of science do not unlearn their scientific worldview when science cannot explain a certain phenomenon. Instead, they either ignore the phenomenon, or they redouble their efforts to understand it scientifically. Whether scientific or magical, it is difficult, and indeed occasionally dangerous, to set aside an entire worldview. Nevertheless, sometimes progress comes from doing so, and magical thinking in data mining is often counter-productive.

There are several related manifestations of magical thinking in data mining. One manifestation is non-probabilistic thinking. For example, a participant wrote "On analyzing the data I found many inconsistent instances (i.e. same values for all attributes but differing [class] values) and removed the minority class of them for training since they might confuse [sic] the algorithm." Similarly, other participants wrote "The original dataset was cleaned in the way that all contradictory records were reclassified as positive."

Another manifestation of magical thinking is the belief that an ideal data mining method does exist that yields classifiers with ideal accuracy. Perhaps the clearest example of this belief is provided by the author who wrote "Two important points [sic] about the CART software from Salford systems is its ability to identify an optimal tree, one providing the least amount of error between the learning data set and the test data set." In fact, even for a fixed learning task and population of examples, different methods may be best given different training or test datasets, because of randomness in how these datasets are drawn from the population.

A third manifestation of magical thinking in data mining is the belief that if a particular learning method or classifier has not performed well, this is always due to some specific cause that can be and should be fixed. This belief is best illustrated by comments from a team that submitted a classifier that would have scored 115, but then withdrew this entry and submitted a classifier that scored

107. This team wrote that the first classifier “had much better luck with the test set” but instead of quantifying this phenomenon and recognizing that the difference between 107 and 115 is easily due to randomness in the test data, they suggested a modification to their heuristics and wrote “If some extra effort had been put in this direction, we believe that SVMs could have given even better results.” If a spell fails to work, the magician always thinks that he must correct some small mistake, not that the spell may have no chance of working. If a spell appears to have the desired outcome, the magician does not pause to consider whether or not he can be sure that the outcome was caused by the spell.

Magical thinking is not always less productive than traditional scientific thinking. Being primed to see patterns in small datasets is an innate characteristic of humans and perhaps other animals also, and this characteristic is often useful for success in everyday life [5]. Moreover, the starting point of scientific thinking is often a type of magical thinking: scientists commonly posit hypotheses based on a low number of observations. These hypotheses are frequently useful because scientists tend to explore small and well-considered hypothesis spaces that are based on background knowledge and hence implicitly on a large number of previous observations. In data mining however, patterns in a dataset are typically not genuine unless they satisfy formal scientific criteria of statistical significance, because the space of patterns explored is large.

7. ECONOMIC ISSUES

Both the contest submissions and the design of the contest hint at a lack of understanding of the economic issues inherent in a data mining task such as the CoIL contest task. In similar real-world domains the task should not be to identify a fixed percentage of customers most likely to have a caravan insurance policy. Instead, the task should be to identify which customers should be offered such a policy, however few or many these customers are. This task is very different for at least two reasons.

First, it is usually economically irrational to offer an insurance policy to some arbitrary percentage of customers. Instead, an offer should be made to a customer if and only if the expected profit from making the offer is greater than the cost of making the offer. Therefore, the aim of data mining should be to estimate the probability that a customer would accept an offer, and also the costs and benefits of the customer accepting or declining [10].

Second, a customer should not be offered an insurance policy just because he or she resembles other customers who have the same type of policy. The characteristics that predict who is most likely to accept a special solicitation may be very different from the characteristics that predict who already has a particular type of policy.

As an obvious special case of both points above, customers who already have a caravan policy should presumably not be offered one again. The economically correct test set for a scenario similar to that of the CoIL contest would consist entirely of customers who do not have a caravan policy. Hence the training and test sets would not be random samples from the same population.

Some participants appear to have basic misconceptions about the centrality of rational decision-making to data mining. For example, one person wrote “Main difficulty of the dataset is its noisiness [...] This is the reason why in most of the communications between participants, other terms like lift-of or up-lift were used instead of accuracy.” This statement is incorrect. Lift was used as a metric of success because the real task is to make optimal cost-sensitive decisions, and maximizing lift is a better proxy for this objective than maximizing accuracy.

Any formal contest must have a definite structure and cannot mimic a real world scenario with complete fidelity. Nevertheless, measuring accuracy on an arbitrary percentage of the test set is too far from reality for a data mining contest. In contests that are better designed, the performance of participants is evaluated using a matrix of real-world costs and benefits. In this respect the contest organized in conjunction with the 1998 KDD conference, for example, was more realistic than the CoIL contest.

An open issue for future data mining contests is how they can be used to compare and evaluate not just data mining algorithms, but also methodologies for data mining. It is noteworthy that none of the reports written by CoIL contest participants mention using any part of the CRISP-DM European standard methodology for data mining [9].

8. CHOICE OF LEARNING METHOD

Although it is difficult to say with certainty that one learning method gives more accurate classifiers than another, it is possible to say with certainty that for practical reasons, some learning methods are less suitable than others for the CoIL contest task and all similar tasks.

Some data mining software is too slow for real applications. For example, concerning a commercial tool one participant wrote “an analysis of the whole training data set took about 2 hours on a Pentium III with 500 MHz. Changing that [search depth] value to 3 made the analysis last for about 28 hours.” Interestingly, a web page for this tool said after the contest that its “high performance enables you to quickly and reliably analyze small to extremely large quantities of data.”

Many data mining methods are not flexible enough to cope with the variety of data types found in real commercial data. Most methods can handle numeric features and discrete-valued features. However, commercial data often contains features that are mixed: some training examples have numeric values for a feature, and other examples have symbolic values for the same feature. The symbolic values may have common meanings, for example “missing” or “not applicable,” or they may be domain-specific, for example “account frozen.”

Even if all the values of a feature appear to be numeric, it is often the case that some specific values are really discrete special cases. Occasionally, some values such as 999 are error codes. For almost all account balance or payment attributes, zero is a value that is common and that has a meaning that is quite different from that of any non-zero amount. Data mining methods that discretize numerical values can handle mixed attributes and zero or other specific codes as special cases easily. Other data mining methods often lose a lot of useful information by treating zero and other special codes as ordinary numeric values.

Some participants appeared to be unaware that software was available that is much more capable than what they used. For example, one team wrote “the number of features in target selection is typically large (50 to 250 features are common), and hence clustering in such a high dimensional space is computationally prohibitive. Moreover the data then tends to be sparse (there is always a feature where two records differ), and clustering algorithms fail in dealing with such data.” These statements about clustering methods are false. The *k*-means algorithm, for example, can handle datasets with millions of records and hundreds of dimensions, where no two records are identical, in effectively linear time [4]. Of course the *k*-means algorithm is not a panacea: it assumes that all features are numerical and a Euclidean distance metric, and no universally good method is known for relaxing these assumptions.

In general, a learning method is not useful for a data mining problem similar to the CoIL contest task unless the method gives numerical scores to examples, such that the scores are monotonically correlated with the probability of being positive. In a scenario where less than 6% of examples are positive, as here, while some examples are much more likely to be positive than others, it is unusual for any example to have a true probability of being positive that is over 50%. Therefore, it is rational for an accuracy-maximizing classifier that only makes yes/no predictions to label all test examples negative, a behavior that is useless.

As mentioned in Section 7, to make optimal decisions one needs in general to know the actual probability that a given test example is positive. However, if the cost/benefit matrix is the same for all test examples, then one only needs to know whether a given test example falls above or below a fixed probability threshold, which corresponds to a fixed score threshold, if scores and probabilities are monotonically related. If the task is simply to maximize lift, then one only has to rank test examples according to their scores. No threshold or actual probabilities are needed.

While a wide variety of learning methods can provide scores that are useful for ranking examples, decision tree and rule induction methods tend to be less suitable than other methods, because they suffer from fragmentation: the phenomenon that each rule or decision tree leaf is based on a small subset of the training data. Fragmentation typically causes three difficulties: lack of statistical significance, lack of discrimination, and lack of interpretability,

Lack of statistical significance is the same issue discussed at the end of Section 2, since each rule or leaf is a pattern based on a small subset of training examples. Lack of discrimination is the problem that if examples are scored using rules or using empirical probabilities derived from decision tree leaves, then many examples are assigned exactly the same score. Identical scores make it impossible to order these examples intelligently. For example, one participant wrote “To get our selected population down to 800 we randomly deleted 78 individuals from the set of 424 who were selected by only 7 rules.”

Lack of interpretability is the problem that a rule or decision tree leaf cannot be understood in isolation. Most rule induction methods produce an ordered collection of rules where each rule applies to an example only if all its predecessors in the ordering do not apply. Therefore, each rule is not a logical implication that is valid in isolation. Rules, and similarly decision tree leaves, can only be understood in combination.

It is not the case that all decision tree or rule induction methods are unsuitable for marketing tasks like the CoIL contest task. If the pruning algorithm used is sensitive to unbalanced data, and probability estimates at leaves are smoothed, then decision trees can be fully competitive with naive Bayesian classifiers on commercial response prediction tasks [11].

9. CONCLUSIONS

In summary, there are three main lessons to be learned from the CoIL data mining contest. The first two lessons are technical, one positive and one negative. The positive lesson is that good methods are available for classifier learning tasks similar to the CoIL contest task. In particular, naive Bayesian learning gives both good predictive accuracy and interpretable models in many commercial applications. If necessary, new interaction features and boosting can improve accuracy without impairing comprehensibility. The negative technical lesson is that reliable algorithms are still not available for doing feature subset selection and detecting feature interactions at the same time.

The third and most important lesson is more sociological than technical. There is a clear lack of awareness and understanding among some researchers and many practitioners in data mining of the issue of statistical significance. In many applications, the categories that we particularly want to model are rare. Given a training set with a small number of members of rare categories, it is pointless to apply excessively complicated learning methods, or to use an excessively time-consuming model search method. In either case, one is likely to overfit the data and to fool oneself. Fooling oneself happens both in estimating predictive accuracy and also in interpreting models to discover actionable insights. Given a small dataset such as the one for the CoIL challenge, only a few predictive relationships are statistically reliable. Other apparent relationships are likely to be spurious.

Acknowledgments: The author is grateful to Gregory Piatetsky-Shapiro, Steve Gallant, Paul Kube, and Peter van der Putten for careful and useful comments on drafts of this paper.

10. REFERENCES

[1] G. Bateson. The logical categories of learning and communication. In *Steps to an Ecology of Mind*, pages 279–308. Ballantine Books, 1972.

[2] T. G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1924, 1998.

[3] C. Elkan. Boosting and naive Bayesian learning. Technical Report CS97-557, Department of Computer Science and Engineering, University of California, San Diego, 1997.

[4] F. Farnstrom, J. Lewis, and C. Elkan. Scalability for clustering algorithms revisited. *ACM SIGKDD Explorations*, 2(1):51–57, 2000.

[5] G. Gigerenzer. *Adaptive Thinking: Rationality in the Real World*. Oxford University Press, 2000.

[6] T. S. Kuhn. *The Structure of Scientific Revolutions*. University of Chicago Press, 1962.

[7] S. D. Silvey. *Statistical Inference*. John Wiley & Sons, Inc., 1975. Reprinted with corrections.

[8] CoIL challenge 2000: The insurance company case. Technical Report 2000-09, Leiden Institute of Advanced Computer Science, Netherlands, 2000. Available at `www.wi.leidenuniv.nl/~putten/library/cc2000/report.html`.

[9] R. Wirth and J. Hipp. CRISP-DM: Towards a standard process model for data mining. In *Proceedings of the Fourth International Conference on the Practical Application of Knowledge Discovery and Data Mining (PADD’00)*, pages 29–39, 2000.

[10] B. Zadrozny and C. Elkan. Learning and making decisions when costs and probabilities are both unknown. In *Proceedings of the Seventh International Conference on Knowledge Discovery and Data Mining*. AAAI Press (distributed by MIT Press), Aug. 2001.

[11] B. Zadrozny and C. Elkan. Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In *Proceedings of the Eighteenth International Conference on Machine Learning*, June 2001.