Analyzing Attribute Dependencies

Aleks Jakulin and Ivan Bratko

Faculty of Computer and Information Science, University of Ljubljana,
Tržaška 25, Ljubljana, Slovenia

Abstract. Many effective and efficient learning algorithms assume independence of attributes. They often perform well even in domains where this assumption is not really true. However, they may fail badly when the degree of attribute dependencies becomes critical. In this paper, we examine methods for detecting deviations from independence. These dependencies give rise to "interactions" between attributes which affect the performance of learning algorithms. We first formally define the degree of interaction between attributes through the deviation of the best possible "voting" classifier from the true relation between the class and the attributes in a domain. Then we propose a practical heuristic for detecting attribute interactions, called *interaction gain*. We experimentally investigate the suitability of interaction gain for handling attribute interactions in machine learning. We also propose visualization methods for graphical exploration of interactions in a domain.

1 Introduction

Many learning algorithms assume independence of attributes, such as the naïve Bayesian classifier (NBC), logistic regression, and several others. The independence assumption licenses the classifier to collect the evidence for a class from individual attributes separately. An attribute's contribution to class evidence is thus determined independently of other attributes. The independence assumption does not merely simplify the learning algorithm; it also results in robust performance and in simplicity of the learned models.

Estimating evidence from given training data with the independence assumption is more robust than when attribute dependencies are taken into account. The evidence from individual attributes can be estimated from larger data samples, whereas the handling of attribute dependencies leads to fragmentation of available data and consequently to unreliable estimates of evidence. This increase in robustness is particularly important when data is scarce, a common problem in many applications. In practice these unreliable estimates often cause inferior performance of more sophisticated methods.

Methods like NBC that consider one attribute at a time are called "myopic." Such methods compute evidence about the class separately for each attribute (independently from other attributes), and then simply "sum up" all these pieces of evidence. This "voting" does not have to be an actual arithmetic sum (for example, it can be the product, that is the sum of logarithms, as in NBC). The aggregation of pieces of evidence coming from individual attributes does not depend on the relations among the attributes. We will refer to such methods as "voting methods;" they employ "voting classifiers."

A well-known example where the myopia of voting methods results in complete failure, is the concept of exclusive OR: $C = XOR(X, Y)$, where C is a Boolean class, and X and Y are Boolean attributes. Myopically looking at attribute X alone provides no evidence about the value of C. The reason is that the relation between X and C critically depends on Y. For $Y = 0, C = X$; for $Y = 1, C \neq X$. Similarly, Y alone fails. However, *X and Y together* perfectly determine C. We say that there is a *positive interaction* between X and Y with respect to C. In the case of a positive interaction the evidence from jointly X and Y about C is greater than the sum of the evidence from X alone and evidence from Y alone.

The opposite may also happen, namely that the evidence from X and Y jointly is worth less than the sum of the individual pieces of evidence. In such cases we say that there is a *negative interaction* between X and Y w.r.t. C. A simple example is when attribute Y is (essentially) a duplicate of X. For example, the length of the diagonal of a square duplicates the side of the square. Voting classifiers are confused by negative interactions as well by positive ones.

2 Attribute Interactions

Let us first define the concept of interaction among attributes formally. Let there be a supervised learning problem with class C and attributes $X_1, X_2, \ldots$. Under conditions of noise or incomplete information, the attributes need not determine the class values perfectly. Instead, they provide some "degree of evidence" for or against particular class values. For example, given an attribute-value vector, the degrees of evidence for all possible class values may be a probability distribution over the class values given the attribute values.

Let the *evidence function* $f(C, X_1, X_2, \ldots, X_k)$ define some chosen "true" degree of evidence for class C in the domain. The task of machine learning is to induce an approximation to function f from training data. In this sense, f is the target concept for learning. In classification, f (or its approximation) would be used as follows: if for given attribute values $x_1, x_2, \ldots, x_k : f(c_1, x_1, x_2, \ldots, x_k) > f(c_2, x_1, \ldots, x_k)$, then the class c_1 is more likely than c_2.

We define the presence, or absence, of interactions among the attributes as follows. If the evidence function can be written as a ("voting") sum:

$$f(C, X_1, X_2, \ldots, X_k) = v\left(\sum_{i=1,\ldots,k} e_i(C, X_i)\right) \tag{1}$$

for some voting function v, and myopic predictor functions $e_1, e_2, \ldots, e_k$, then there is no interaction between the attributes. Equation (1) requires that the joint evidence of all the attributes can essentially be reduced to the sum of the pieces of evidence $e_i(C, X_i)$ from individual attributes. The function e_i is a predictor that investigates the relationship between an attribute X_i and the class C.

If, on the other hand, no such functions $v, e_1, e_2, \ldots, e_k$ exist for which (1) holds, then there *are* interactions among the attributes. The strength of interactions IS can be defined as

$$IS := f(C, X_1, X_2, \ldots, X_k) - v\left(\sum_i e_i(C, X_i)\right). \tag{2}$$

IS greater than some positive threshold would indicate a positive interaction, and IS less than some negative threshold would indicate a negative interaction. Positive interactions indicate that a holistic view of the attributes unveils new evidence. Negative interactions are caused by multiple attributes providing the same evidence, while the evidence should count only once.

Many classifiers are based on the linear form of (1): naïve Bayesian classifier, logistic and linear regression, linear discriminants, support vector machines with linear kernels, and others. Hence, interaction analysis is relevant for all these methods. All we have written about relationships between attributes also carries over to relationships between predictors in an ensemble.

3 Interaction Gain: A Heuristic for Detecting Interactions

The above definition of an interaction provides a "golden standard" for deciding, in principle, whether there is interaction between two attributes. The definition is, however, hard to use as a procedure for detecting interactions in practice. Its implementation would require combinatorial optimization.

We will not refine the above definition of interactions to make it applicable in a practical learning setting. Instead, we propose a heuristic test, called *interaction gain*, for detecting positive and negative interactions in the data, in the spirit of the above definition. Our heuristic will be based on information-theoretic notion of entropy as the measure of classifier performance, joint probability distribution as the predictor, and the chain rule as the voting function. Entropy has many useful properties, such as linear additivity of entropy with independent sources. We will consider discriminative learning, where our task is to study the class probability distribution. That is why we will always investigate relationships between an attribute and the class, or between attributes with respect to the class.

Interaction gain is based on the well-known idea of information gain. *Information gain* of a single attribute X with respect to class C [1], also known as *mutual information* between X and C, measured in bits:

$$\mathrm{Gain}_C(X) = I(X; C) = \sum_x \sum_c P(x, c) \log \frac{P(x, c)}{P(x)P(c)}. \tag{3}$$

Information gain can be regarded as a measure of the strength of a 2-way interaction between an attribute X and the class C. In this spirit, we can generalize it to 3-way interactions by introducing the *interaction gain* [2]:

$$I(X;Y;C) := I(X,Y;C) - I(X;C) - I(Y;C). \tag{4}$$

Interaction gain is also measured in bits, and can be understood as the difference between the actual decrease in entropy achieved by the joint attribute XY and the expected decrease in entropy with the assumption of independence between attributes X and Y. The higher the interaction gain, the more information was gained by joining the attributes in the Cartesian product, in comparison with the information gained from single attributes. When the interaction gain is negative, both X and Y carry the same evidence, which was consequently subtracted twice.

To simplify our understanding, we can use the entropy $H(X)$ to measure the uncertainty of an information source X through the identity $I(X;Y) = H(X) + H(Y) - H(X,Y)$. It it then not difficult to show that $I(X;Y;C) = I(X;Y|C) - I(X;Y)$. Here, $I(X;Y|C) = H(X|C) + H(Y|C) - H(X,Y|C)$ is conditional mutual information, a measure of dependence of two attributes given the *context* of C. $I(X;Y)$ is an information-theoretic measure of dependence or "correlation" between the attributes X and Y regardless of the context.

Interaction gain (4) describes the change in a dependence of a pair of attributes X, Y by introducing context C. It is quite easy to see that when interaction gain is negative, context decreased the amount of dependence. When the interaction gain is positive, context increased the amount of dependence. When the interaction gain is zero, context did not affect the dependence between the two attributes. Interaction gain is identical to the notion of mutual information among three random variables [3].

4 Detecting and Resolving Interactions

A number of methods have been proposed to account for dependencies in machine learning, in particular with respect to the naïve Bayesian classification model [4–6], showing improvement in comparison with the basic model. The first two of these methods, in a sense, perform feature construction; new features are constructed from interacting attributes, by relying on detection of interactions. On the other hand, tree augmentation [6], merely makes the dependence explicit, but this is more a syntactic distinction. In this section we experimentally investigate the relevance of interaction gain as a heuristic for guiding feature construction.

The main questions addressed in this section are: Is interaction gain a good heuristic for detecting interactions? Does it correspond well to the principled definition of interactions in Section 2?

The experimental scenario is as follows:

1. We formulate an operational approximation to our definition of interaction. This is a reasonable and easy to implement special case of formula (1) as follows: the degree of evidence is a probability, and the formula (1) is instantiated to the naïve Bayesian formula. It provides an efficient test for interactions. We refer to this test as BS.

2. In each experimental data set, we select the most interacting pair of attributes according to (a) BS, (b) positive interaction gain (PIG), and (c) negative interaction gain (NIG).
3. We build naïve Bayesian classifiers (NBC) in which the selected interactions are "resolved." That is, the selected pair of most interacting attributes is replaced in NBC by its Cartesian product. This interaction resolution is done for the result of each of the three interaction detection heuristics (BS, PIG and NIG), and the performance of the three resulting classifiers is compared.

We chose to measure the performance of a classifier with Brier score (described below). We avoided classification accuracy as performance measure for the following reasons. Classification accuracy is not very sensitive in the context of probabilistic classification: it usually does not matter for classification accuracy whether a classifier predicted the true class with the probability of 1 or with the probability of, e.g., 0.51. To account for the precision of probabilistic predictions, we employed Brier score [7]. Given two probability distributions, the predicted class probability distribution $\hat{p}$, and the actual class probability distribution p, where the class can take N values, the Brier score of the prediction is:

$$b(\hat{p}, p) := \frac{1}{N} \sum_{i=1}^{N} (\hat{p}_i - p_i)^2 \tag{5}$$

The larger the Brier score, the worse a prediction. Error rate is a special case of Brier score for deterministic classifiers, while Brier score could additionally reward a probabilistic classifier for better estimating the probability. In a practical evaluation of a classifier given a particular testing instance, we approximate the actual class distribution by assigning a probability of 1 to the true class of the testing instance. For multiple testing instances, we compute the average Brier score.

We used two information-theoretic heuristics, based on interaction gain: the interaction with the maximal positive magnitude (PIG), and the interaction with the minimal negative magnitude (NIG). We also used a wrapper-like heuristic: the interaction with the maximum improvement in the naïve Bayesian classifier performance after merging the attribute pair, as measured with the Brier score or classification accuracy on the training set, $b(NBC(C|X)(C|Y)) - b(NBC(C|X, Y))$, the first term corresponding to the independence-assuming naïve Bayesian classifier and the second to the non-assuming Bayesian classifier. As the basic learning algorithm, we used the naïve Bayesian classifier. After the most important interaction was determined outside the context of other attributes, we modified the NBC model created with all the domain's attributes by taking the single most dependent pair of attributes and replacing them with their Cartesian product, thus eliminating that particular dependence. All the numerical attributes in the domains were discretized beforehand, and missing values represented as special values. Evaluations of the default NBC model and of its modifications with different guiding heuristics were performed with 10-fold cross-validation. For each fold, we computed the average score. For each domain, we computed the score mean and the standard error over the 10 fold experiments. We performed all our experiments with the Orange toolkit [8].

domain	NB	PIG	NIG	BS
lung	0.230√	**0.208**	0.247	0.243
soy-small	**0.016**	**0.016**	**0.016**	**0.016**
zoo	**0.018**	0.019√	**0.018**	**0.018**
lymph	0.079√	0.094	0.077√	**0.075**
wine	**0.010**	0.010√	0.015	0.014
glass	**0.070**	0.071√	0.071√	0.073√
breast	**0.212**	0.242	0.212√	0.221√
ecoli	**0.032**	0.033√	0.039	0.046
horse-col	0.108√	0.127	0.106√	**0.104**
voting	0.089	0.098	0.089	**0.063**
monk3†	0.042	**0.027**	0.042	**0.027**
monk1†	0.175	**0.012**	0.176	**0.012**
monk2†	0.226√	**0.223**	0.224√	0.226√
soy-large	0.008√	**0.007**	0.008√	0.008√
wisc-canc	0.024√	**0.023**	0.024√	0.026√
austral	0.120√	0.127	**0.114**	0.116√
credit	0.116√	0.122	**0.111**	0.115√
pima	0.159√	0.159√	**0.158**	0.159√
vehicle	0.142	0.136	0.138	**0.127**
heart	**0.095**	0.098	0.095√	0.095√
german	**0.173**	0.175√	0.174√	0.175√
cmc	0.199√	**0.194**	0.195√	0.198√
segment	0.016	0.017	0.017	**0.015**
krkp†	0.092	0.077√	0.088	**0.076**
mushroom	0.002	0.006	**0.002**	**0.002**
adult	0.119	0.120	**0.115**	0.119

Table 1. The table lists Brier scores obtained with 10-fold cross validation after resolving the most important interaction, as assessed with different methods. A result is set in bold face if it is the best for the domain, and checked if it is within the standard error of the best result for the domain. We marked the artificial† domains.

In Table 1 we sorted 26 of the UCI KDD archive [9] domains according to the number of instances in the domain, from the smallest on the top left to the largest on the bottom right, along with the results obtained in the above manner. We can observe that in two domains resolution methods matched the original result. In 6 domains resolution methods worsened the results, in 10 domains the original performance was within a standard error of the best, and in 8 domains, the improvement was significant beyond a standard error. We can thus confirm that accounting for interactions in this primitive way did help in $\sim$70% of the domains.

If comparing different resolution algorithms, the Brier-score driven interaction detection was superior to either of the information-based heuristics PIG or NIG, achieving the best result in 11 domains. However, in only two domains, 'voting' and 'segment,' neither of PIG and NIG was able to improve the result while BS did. Thus, information-theoretic heuristics are a reasonable and effective choice for interaction detection, providing competitive results even if the BS heuristic had the advantage of using the same evaluation function as the final classifier evaluation. PIG improved the results in 5 natural domains, and in 4 artificial domains. This confirms earlier intuitions that the XOR-type phenomena occur more often in synthetic domains. NIG provided an improvement in 7 domains, all of them natural. Negative interactions are generally very frequent, but probabilistic overfitting cannot always be resolved to a satisfactory extent by merely resolving the strongest interaction because the result is dependent on the balance between multiple negatively interacting attributes.

times	NB	PIG	NIG	BS	AIG
best	8	8	7	11	10
good $\surd$	10	7	10	11	7
bad	8	11	9	4	9

Table 2. A summary of results shows that BS-driven interaction resolution provides the most robust approach, while AIG follows closely behind.

In Table 2, we summarize the performance of different methods, including AIG as an approach deciding between the application of NIG and PIG in a given domain. AIG suggests resolving the interaction with the largest absolute interaction gain. We can also observe that success is more likely when the domain contains a large number of instances. It is a known result from statistical literature that a lot of evidence is needed to show the significance of higher-order interactions [10].

5 Visualization of Interactions

The analysis of attribute relationships can be facilitated by methods of information visualization. We present two methods for visualization of attribute interactions. Interaction dendrogram illustrates groups of mutually interacting attributes. Interaction graph provides a detailed insight in the nature of attribute relationships in a given domain.

5.1 Interaction Dendrograms

Interaction dendrogram illustrates the change in dependence between pairs of attributes after introducing the context. The direction of change is not important: we will distinguish this later. If we bind proximity in our presentation to the change in level of dependence, either positive or negative, we can define the distance d_m between two attributes X, Y as:

$$d_m(X,Y) := \begin{cases} |I(X;Y;C)|^{-1} & \text{if } |I(X;Y;C)|^{-1} < 1000, \\ 1000 & \text{otherwise.} \end{cases} \tag{6}$$

Here, 1000 is a chosen upper bound as to prevent attribute independence from disproportionately affecting the graphical representation. To present the function d_m to a human analyst, we tabulate it in a dissimilarity matrix and apply the techniques of hierarchical clustering or multi-dimensional scaling. Dependent attributes will hence appear close to one another; independent attributes

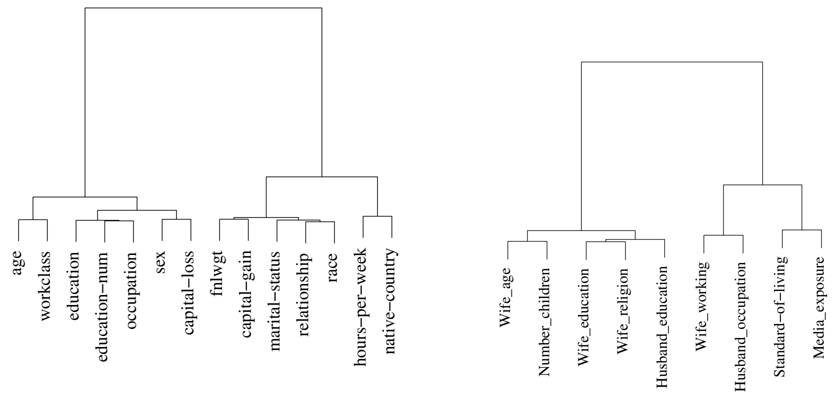

Fig. 1. An interaction dendrogram illustrates which attributes interact, positively or negatively, in the 'census/adult' (left) and 'cmc' (right) data sets. We used the Ward's method for agglomerative hierarchical clustering [11].

will appear far from one another. This visualization is an approach to variable clustering, which is normally applied to numerical variables outside the context of supervised learning, e.g. [12]. Diagrams, such as those in Fig. 1, may be directly useful for feature selection: the search for the best model starts by only picking the individually best attribute from each cluster. We must note, however, that an attribute's membership in a cluster merely indicates its average relationship with other cluster members.

5.2 Interaction Graphs

The analysis described in the previous section was limited to rendering the magnitude of interaction gains between attributes. We cannot use the dendrogram to identify whether an interaction is positive or negative, nor can we see the importance of each attribute. An interaction graph presents the proximity matrix better. To reduce clutter, only the strongest N interactions are shown, usually $5 \leq N \leq 20$. With an interactive method for graph exploration, this trick would not be necessary. We also noticed that the distribution of interaction gains usually follows a Gaussian-like distribution, with only a few interactions standing out from the crowd, either on the positive or on the negative side.

Each node in the interaction graph corresponds to an attribute. The information gain of each attribute is expressed as a percentage of the class entropy (although some other uncertainty measure, such as the error rate or Brier score, could be used in the place of class entropy), and written below the attribute name. There are two kinds of edges, bidirectional arrows and undirected dashed arcs. Arcs indicate negative interactions, implying that the two attributes provide partly the same information. The amount of shared information, as a percentage of the class entropy, labels the arc. Analogously, the amount of novel information labels the arrow, indicating a positive interaction between a pair of attributes. Figure 2 explains the interpretation of the interaction graph, while Figs. 3 and 4 illustrate two domains. We used the 'dot' utility [13] for generating the graph.

6 Implications for Classification

In discussing implications of interaction analysis to classification, there are two relevant questions. The first is the question of significance: when is a particular interaction worth considering. The second is the question of how to treat negative and positive interactions between attributes in the data.

In theory, we should react whenever the conditional mutual information $I(X;Y|C)$ deviates sufficiently from zero: it is a test of conditional dependence. In practice, using a joint probability distribution for XY would increase the complexity of the classifier, and this is often not justified when the training data is scarce. Namely, introducing the joint conditional probability distribution $P(X,Y|C)$ in place of two marginal probability distributions $P(X|C)P(Y|C)$ increases the degrees of freedom of the model, thus increasing the likelihood that the fit was accidental. In the spirit of Occam's razor, we should increase the complexity of a classifier only to obtain a significant improvement in classification performance. Hence, the true test in practical applications is improvement in generalization performance, measured with devices such as the training/testing set separation and cross-validation. When the improvement after accounting for an interaction is significant, the interaction itself is significant.

6.1 Negative Interactions

If X and Y are interacting negatively, they both provide the same information. If we disregard a negative interaction, we are modifying the class probability distribution with the same information twice. The estimated class probabilities become excessively confident, another case of *overfitting*. Unbalanced class probability estimates by themselves do not necessarily bother several classification performance measures, such as the classification accuracy and the ROC, because they do not always change the classifications. Also, mutual attribute dependence is *not* a problem when the interaction gain is zero. For example, a duplicate random attribute unrelated to class is equally dependent inside and outside the context of the class, and the pair's interaction gain is zero. This kind of dependence does not affect classification performance whatsoever.

The most frequent method of resolving negative interactions is feature selection. However, we observe that two noisy measurements of the same quantity are better than a single measurement, so other approaches may be preferable. The simplest way lies in weighing attributes, such as feature weighting or least-squares regression. Alternatively, a latent attribute L can be inferred, to provide evidence for all three attributes: X, Y and C. The trivial approach to latent attribute inference is the introduction of the Cartesian product between attributes, the technique we applied in our experiments, but methods like factor analysis,

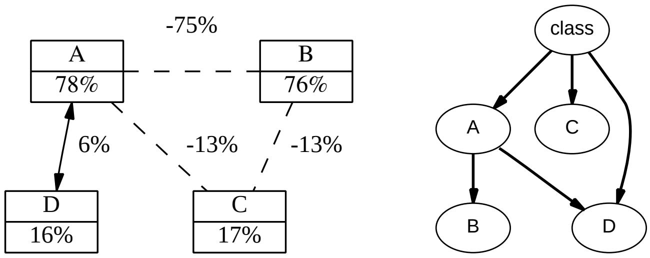

Fig. 2. The four most informative attributes were selected from a real medical domain. In the interaction graph (left), the most important attribute A alone eliminates 78% of class entropy. The second most important attribute B alone eliminates 76% of class entropy, but A and B interact negatively (dashed arc), and share 75% of class entropy. So B reduces class entropy by only 76-75=1% of its truly own once we have accounted for A: but if we leave B out in feature subset selection, we are giving this information up. Similarly, C provides 4% of its own information, while the remaining 13% is contained also in A and in B. Attribute D provides 'only' 16% of information, but if we account for the positive interaction between A and D (solid bidirectional arrow), we provide for 78+16+6=100% of class entropy. Consequently, only attributes A and D are needed, and they should be treated as dependent. A Bayesian network [14] learned from the domain data (right) is arguably less informative.

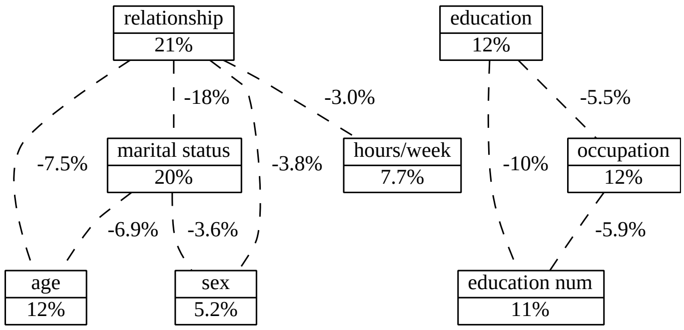

Fig. 3. An interaction graph for the 'census/adult' domain confirms our intuitions about natural relationships between the attributes. All interactions in this graph are negative, but there are two clusters of them.

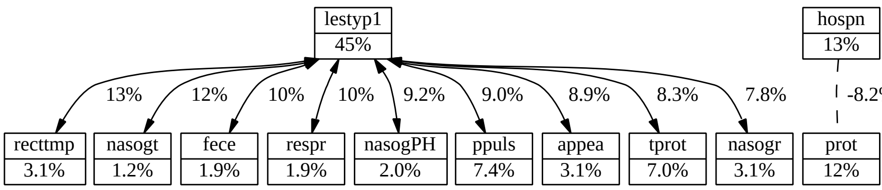

Fig. 4. In this illustration of the 'horse colic' domain, one attribute appears to moderate a number of other attributes' relationships with the class. There is a separate and independent negative interaction on the right.

independent component analysis are also applicable here. This kind of attribute dependence is a simple and obvious explanation for conditional dependence. Negative interactions allow us to simplify the model in the sense of reducing the quantity of evidence being dealt with.

6.2 Positive Interactions

The second cause of independence assumption violation is more interesting: two attributes together explain more than what we estimated from each attribute individually. There could be some unexplained moderating effect of the first attribute onto the second attribute's evidence for C. There could be a functional dependence involving X, Y and C, possibly resolved by feature construction. Such positive interactions can be inferred from a positive value of interaction gain. Positive interactions are interesting subjects for additional study of the domain, indicating complex regularities. They too can be handled with latent attribute inference. Positive interactions indicate a possible benefit of complicating the model.

If we disregard a positive interaction by assuming attribute independence, we are not taking advantage of all the information available: we are *underfitting*. Of course, one should note that the probabilities, on the basis of which entropy is calculated, might not be realistic. Since the probabilities of a joint probability distribution of two values of two attributes and the class are computed with fewer supporting examples than those computed with only one attribute value and the class, the 3-way interaction gains are less trustworthy than 2-way interaction gains. Consequently, positive interaction gains may in small domains indicate only a coincidental regularity. Taking accidental dependencies into consideration is a well-known cause of *overfitting*, but there are several ways of remedying this probability estimation problem, e.g. [15].

7 Conclusion

In this paper we studied the detection and resolution of dependencies between attributes in machine learning. First we formally defined the degree of interaction between attributes through the deviation of the best possible "voting" classifier from the true relation between the class and the attributes in a domain. Then we proposed the *interaction gain* as a practical heuristic for detecting attribute interactions. We experimentally investigated the suitability of interaction gain (IG) for handling attribute dependencies in machine learning. Experimental results can be summarized as follows:

– IG as a heuristic for detecting interactions performs similarly as the BS criterion (a heuristic that was directly derived from the principled formal definition of attribute interaction), and enables the resolution of interactions in classification learning with similar performance as BS.
– IG enables the distinction between *positive* and *negative* interactions while BS does not distinguish between these two types of interaction.

- According to empirical results, in real-world domains strong interactions are rare, particularly positive interactions.
- In typical artificial domains, strong interactions are more frequent, particularly positive interactions. The IG heuristic reliably detects them.

We also presented visualization methods for graphical exploration of interactions in a domain. These are useful tools that should help expert's understanding of the domain under study, and could possibly be used in constructing a predictive model.

Problems for future work include: handling of n-way interactions where $n > 3$; building learning algorithms that will incorporate interaction detection facilities, and provide superior means of resolving these interactions when building classifiers.

References

1. Hunt, E.B., Martin, J., Stone, P.: Experiments in Induction. Academic Press, New York (1966)
2. Jakulin, A.: Attribute interactions in machine learning. Master's thesis, University of Ljubljana, Faculty of Computer and Information Science (2003)
3. Yeung, R.W.: A new outlook on Shannon's information measures. IEEE Transactions on Information Theory **37** (1991) 466–474
4. Kononenko, I.: Semi-naive Bayesian classifier. In Kodratoff, Y., ed.: European Working Session on Learning - EWSL91. Volume 482 of LNAI., Springer Verlag (1991)
5. Pazzani, M.J.: Searching for dependencies in Bayesian classifiers. In: Learning from Data: AI and Statistics V. Springer-Verlag (1996)
6. Friedman, N., Goldszmidt, M.: Building classifiers using Bayesian networks. In: Proc. National Conference on Artificial Intelligence, Menlo Park, CA, AAAI Press (1996) 1277–1284
7. Brier, G.W.: Verification of forecasts expressed in terms of probability. Weather Rev **78** (1950) 1–3
8. Demšar, J., Zupan, B.: Orange: a data mining framework. http://magix.fri.uni-lj.si/orange (2002)
9. Hettich, S., Bay, S.D.: The UCI KDD archive http://kdd.ics.uci.edu. Irvine, CA: University of California, Department of Information and Computer Science (1999)
10. McClelland, G.H., Judd, C.M.: Statistical difficulties of detecting interactions and moderator effects. Psychological Bulletin **114** (1993) 376–390
11. Struyf, A., Hubert, M., Rousseeuw, P.J.: Integrating robust clustering techniques in S-PLUS. Computational Statistics and Data Analysis **26** (1997) 17–37
12. SAS Institute Inc.: SAS/STAT User's Guide. SAS Institute Inc., Cary, NC, USA (1998)
13. Koutsofios, E., North, S.C.: Drawing Graphs with dot. (1996) Available on research.att.com in dist/drawdag/dotguide.ps.Z.
14. Myllymaki, P., Silander, T., Tirri, H., Uronen, P.: B-Course: A web-based tool for Bayesian and causal data analysis. International Journal on Artificial Intelligence Tools **11** (2002) 369–387
15. Cestnik, B.: Estimating probabilities: A crucial task in machine learning. In: Proc. 9th European Conference on Artificial Intelligence. (1990) 147–149