ESANN 2011 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2011, i6doc.com publ., ISBN 978-2-87419-044-5.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100817300.

Locating Anomalies Using Bayesian Factorizations and Masks

Li Yao Amaury Lendasse Francesco Corona

Department of Information and Computer Science
Aalto University School of Science and Technology
Finland

Abstract. A plethora of methods have been developed to handle anomaly detection in various application domains. This work focuses on locating anomalies inside a categorical data set without assuming any specific domain knowledge. By exploiting the conditional dependence and independence relationships among data attributes, not only can data analysts recognize the anomaly, but also locate the potentially anomalous attributes inside an anomalous instance following its masks. Masks are geometrically generated based on the factorization of the joint probability from a Bayesian network automatically learnt from the given data set.

1 Introduction

In the problem of anomaly detection, unusual patterns inside a given data set are of special interests to data analysts. A comprehensive review of anomaly detection techniques and applications can be found in [2] in which a typical method, either supervised or unsupervised, reports an anomalous score for each test instance. However, the discussion of anomaly detection in categorical data sets is relatively limited, especially in the unsupervised detection where no training labels are available. The most recent work dealing with categorical data sets includes [6], [7] and [8]. In [8], the authors introduce a rule-based algorithm called LERAD in which a minimal set of conditional probabilities are estimated from the data set and then used to compute the anomaly score for each of the instances. In [7], the method learns a Bayesian network by using an Optimal Reinsertion procedure and then uses the learnt Bayesian network to estimate the conditional probability (anomaly score) of an instance given some disease-related environmental variables. The authors in [6] propose Conditional and Marginal Anomaly Tests both of which are defined on the conditional probabilities for a carefully selected set of variables. Even though successful in their application domains, all three methods are not easily generalized into applications where not only labels are unavailable but also the conditional dependencies among variables are hardly known. Exploiting dependencies among variables provides invaluable information for anomaly detection.

This work focuses on anomaly detection in an unlabeled categorical data set without assuming any *a priori* knowledge of it. In particular, the method proposed in this work performs *anomaly location* by which each instance receives a "mask" indicating both its anomaly score and the locations of its potentially anomalous attributes. The fundamental assumption of this work is that in the

ESANN 2011 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2011, i6doc.com publ., ISBN 978-2-87419-044-5.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100817300.

given categorical data set, anomalies take uncommon attributes appearing with low frequencies. In order to locate the anomalies and their anomalous attributes, the suggested approach follows two stages. In the factorization stage, the joint probability of an instance is factorized into a set of conditional and marginal probabilities that best represent the conditional independencies among all attributes. This is achieved by learning a Bayesian network with the maximal Bayesian score from the data set. The second stage generates a mask for each instance using the factorization learnt in the first stage. This is illustrated from a geometrical point of view. The rest of this paper is organized as following: in the theory part, Section 2 illustrates the factorization by performing structural learning on Bayesian networks. Section 3 explains three steps to produce masks. In the experiment part, Section 4 applies the methodology to a real world data set that exemplifies the advantages and limitations of our method.

2 Factorization

Denote with $\mathbf{X}$ a categorical data set with N rows (instances) and K columns (attributes, variables). Denote with $X_k(k = 1, \ldots, K)$ the k-th variable representing the k-th column in $\mathbf{X}$. Denote with $\mathbf{x}_n = \{x_{n1}, \ldots, x_{nK}\}(n = 1, \ldots, N)$ the n-th instance in $\mathbf{X}$. Given $\mathbf{x}_n$, denote its joint probability with $P(\mathbf{x}_n) = P(x_{n1}, x_{n2}, \ldots, x_{nK})$. Denote factorization of $P(\mathbf{x}_n)$ with an operation $\mathcal{F}(P(\mathbf{x}_n)) = \prod_{k=1}^{K} f_k(x_{n,k})$. The factorization defines a multiplication of K factors $f_k(x_{n,k})$ each of which is a function (conditional or marginal probability) of $x_{n,k}$.

In order to factorize $P(\mathbf{x}_n)$, given no knowledge of dependencies and independencies between variables, the intuitive method is to use the probability chain rule. The problems are that there are $K!$ different factorizations by using the chain rule and the selection of factors does not take into account the true dependencies and independencies between groups of attributes. Therefore, we prefer $\mathcal{F}(P(\mathbf{x}_n))$ with factors fully exploring the conditional independencies among variables and meanwhile with the multiplication of factors approximating $P(\mathbf{x}_n)$ as closely as possible. In the extreme case, such a factorization can be achieved by using the assumption that X_k are statistically independent of each other. That is: $\mathcal{F}_1(P(\mathbf{x}_n)) = \prod_{k=1}^{K} f_k(x_{n,k}) = \prod_{k=1}^{K} P(x_{n,k})$. Thus, we obtain an alternative way to compute the joint probability $P(\mathbf{x}_n)$ by using only K marginals $P(x_{n,k})$. The limitation of using the independent assumption is obvious: the statistical independence among variables is not likely to be true. Thus, it is critical to explore automatically the conditional dependencies and independencies among variables and encode this information into the factorization.

2.1 Factorization with Bayesian networks

Bayesian networks exploit the conditional independence within a joint distribution and the use of a Directed Acyclic Graph (DAG) allows a compact representation of those independencies. Given a categorical data set $\mathbf{X}$, a Bayesian network $\mathbf{B}$ for a set of variables X_k consists of a network structure g that encodes

ESANN 2011 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2011, i6doc.com publ., ISBN 978-2-87419-044-5.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100817300.

a set of conditional independence assertions about variables X_k and g indicates a factorization $P(X_1, \ldots, X_K) = \prod_{k=1}^{K} P(X_k|Pa_k)$ where Pa_k denotes the parents of X_k in g.

According to [1], learning the structure of $\mathbf{B}$ is NP-hard. Following the work in [4], the authors in [3] introduce a score-based Markov chain Monte Carlo (MCMC) procedure for structural learning. MCMC converges to $P(g|\mathbf{X}) = \frac{P(\mathbf{X}|g)P(g)}{P(\mathbf{X})}$, the posterior distribution of g given $\mathbf{X}$. Ignoring the normalizing denominator (same for all candidates g), the Bayesian score of g is $score(g) = log[P(\mathbf{X}|g)] + log[P(g)]$. MCMC is guided by both the generation of the next candidate graph and the computation of $score(g)$. We use *potential scale reduction factor* (PSRF) discussed in [5] to evaluate the convergence of MCMC. PSRF monitors the convergence of a directed link between each pair of nodes in g. MCMC converges if and only if all the links converge.

Table 1: The toy example.

instance	X_1	X_2	X_3	$P(\mathbf{x}_n)$	instance	X_1	X_2	X_3	$P(\mathbf{x}_n)$
$\mathbf{x}_1$	1	2	2	0.125	$\mathbf{x}_5$	2	2	3	0.125
$\mathbf{x}_2$	1	1	2	0.125	$\mathbf{x}_6$	2	3	3	0.125
$\mathbf{x}_3$	1	2	3	0.125	$\mathbf{x}_7$	3	2	2	0.125
$\mathbf{x}_4$	1	1	3	0.125	$\mathbf{x}_8$	4	3	1	0.125

For the toy example in Table 1, the structural learning returns two optimal DAGs g_1^* and g_2^* with the same maximal Bayesian score. Two optimal DAGs suggest two factorizations $\mathcal{F}_2(P(\mathbf{x}_n)) = P(x_{n1})P(x_{n2})P(x_{n3}|x_{n1})$ and $\mathcal{F}_3(P(\mathbf{x}_n)) = P(x_{n1}|x_{n3})P(x_{n2})P(x_{n3})$. Comparing the goodness of $\mathcal{F}_1$, $\mathcal{F}_2$ and $\mathcal{F}_3$ on their closeness to $P(\mathbf{x}_n)$ indicates $\mathcal{F}_3$ is the best factorization.

3 Generating masks to locate anomalous attributes

Given an instance $\mathbf{x}_n$, the factorization produces a group of factors $f_k(x_{n,k})$ each of which represents one attribute $x_{n,k}$. Therefore, the task of anomaly location naturally moves on to the factors. We would prefer a clear label for each $x_{n,k}$ indicating whether it is anomalous or not. The mask fulfills this requirement. We define a *mask* for $\mathbf{x}_n$ as a binary vector $\mathbf{m}_n = \{m_{n,1}, \ldots, m_{n,K}\}$ having the same dimension K as $\mathbf{x}_n$. Each $x_{n,k}$ of $\mathbf{x}_n$receives a mask $m_{n,k}$ such that $x_{n,k}$ is "masked" by $m_{n,k}$. As a binary vector, $m_{n,k}$ of $\mathbf{m}_n$ may recieve either 0 or 1 according to the value of the factor $f_k(x_{n,k})$ from the factorization $\mathcal{F}(P(\mathbf{x}_n))$. $m_{n,k} = 0$ indicates the corresponding attribute $x_{n,k}$ it "masks" in $\mathbf{x}_n$ is anomalous while $m_{n,k} = 1$ indicates $x_{n,k}$ is normal. It is likely that a given instance is associated with a mask with $m_{n,k} = 1$ for all $k = \{1, \ldots, K\}$, indicating a perfectly normal instance. Otherwise, all the attributes of $\mathbf{x}_n$ taking the mask 0 indicate the potentially anomalous positions.

ESANN 2011 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2011, i6doc.com publ., ISBN 978-2-87419-044-5.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100817300.

3.1 Mask generation

Step 1: construct a hypercube in the continuous Euclidean space. The process is summarized by $\mathbf{x}_n \rightarrow \mathbf{r}_n \rightarrow \mathbf{p}_n$. Given $\mathbf{x}_n$, it is mapped into $\mathbf{r}_n$ by using the factorization from g^*. So for each $\mathbf{r}_n$, $P(\mathbf{r}_n) = \prod_{k=1}^{K} f_k(x_{n,k}) = \prod_{k=1}^{K}(r_{n,k})$. In order to construct a Euclidean space containing all instances, we use logarithm to transform the product into summation so that $\mathbf{r}_n$ is mapped into a point $\mathbf{p}_n$ in the log-likelihood space of $\mathbf{x}_n$. $\mathbf{p}_n$ has the coordinates of $\{log_e(r_{n,1}), \ldots, log_e(r_{n,K})\}$. A hypercube is constructed in K dimensional Euclidean space so that there are no points falling outside the hypercube. To fully specify the hypercube, one needs only two such points $\mathbf{p}_{max}$ and $\mathbf{p}_{min}$ having the maximal and minimal $log_e(r_{n,k})$ across all n in each direction k. Given $\mathbf{p}_{max}$ and $\mathbf{p}_{min}$, the coordinates of all 2^K vertexes of the hypercube can be derived from those two points. Note that we use a relatively loose definition of "hypercube" since its edges in our case are not necessarily equal. The bounding cube for the toy example is shown in Figure 1.

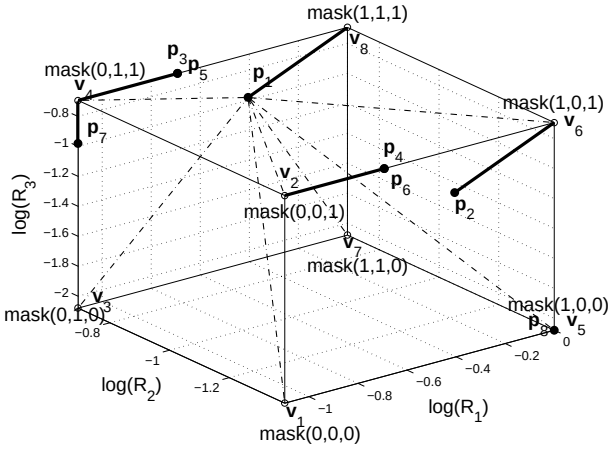

Fig. 1: A geometrical perspective of mask generation using the toy example.

Step 2: assign masks to vertexes of the hypercube. After the hypercube is constructed, we assign a unique mask to each vertex of it. The point $\mathbf{p}_{max}$, the vertex closest to the origin in the space, is assigned a perfect mask $\mathbf{m}_n = \{m_{n,1} = 1, \ldots, m_{n,K} = 1\}$. The point $\mathbf{p}_{min}$, farthest to the origin, is assigned the worst mask $\mathbf{m}_n = \{m_{n,1} = 0, \ldots, m_{n,K} = 0\}$. Following the same pattern of generating the coordinates of the vertexes, each of the other vertexes receives a unique mask, as shown in Figure 1. Since the whole procedure is unsupervised, it is reasonable to consider the extreme instances defined on the vertexes and use them as prototypes to perform step 3.

Step 3: assign masks to instances. Given a point $\mathbf{p}_n$ representing an in-

ESANN 2011 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2011, i6doc.com publ., ISBN 978-2-87419-044-5. Available from http://www.i6doc.com/en/livre/?GCOI=28001100817300.

stance $\mathbf{x}_n$, $\mathbf{p}_n$ finds its nearest vertex by comparing the distances between each vertex of the hypercube and itself. Ultimately, the point gets exactly the same mask as its nearest vertex. $\mathbf{p}_n$ in the toy example are associated with its corresponding vertexes by the bold solid line (shortest distance) in Figure 1. This step resembles nearest neighbor classification such that all vertexes are in the neighborhood of $\mathbf{p}_n$ and the classification label $\mathbf{p}_n$ receives from its nearest neighbor is a mask. There are situations where a point gets more than one nearest neighbor among vertexes, leading to multiple masks.

4 Experiment

The Breast Cancer Wisconsin data set is originally used in the supervised classification of the cancer diagnosis results into benign and malign. We treat it, however, from an unsupervised perspective. The data set consists of 699 instances (samples) and 11 attributes including a patient ID and a class label. 16 instances with missing values are not used in this work. In order to fit into the assumption that benign instances dominate the categorical data set, we reconstruct three data set: full data set $\mathbf{X}_1$ having 444 samples for benign and 239 for malign, $\mathbf{X}_2$ having 444 samples for benign and 160 for malign, and $\mathbf{X}_3$ having 444 samples for benign and 81 for malign. Note that there are duplicated instances in all three data sets. Each unique sample receives a mask and we conclude that samples receiving the perfect mask are benign, otherwise, they are malign with 0s indicating the anomalous attributes.

The classification results are summarized in the confusion matrices in Table 2, 3 and 4 in which α^+ and β^+ denote real and predicted malign, α^- and β^- denote real and predicted benign. The results demonstrate both the merits and

Table 2: $\mathbf{X}_1$

	α^+	α^-
β^+	207	126
β^-	29	87

Table 3: $\mathbf{X}_2$

	α^+	α^-
β^+	152	124
β^-	5	89

Table 4: $\mathbf{X}_3$

	α^+	α^-
β^+	81	137
β^-	0	76

drawbacks of our method. It performs well in detecting maligns with the accuracy of 85.17%, 96.82% and 100% individually as the proportion of maligns drops from 35.0% for $\mathbf{X}_1$ to 26.5% for $\mathbf{X}_2$ to 15.4% for $\mathbf{X}_3$. The decreasing error rate with the increasing dominance of normal instances consolidates the applicability of the method. Most attractively, the mask provides each resulted malign sample with anomalous attributes potentially leading to cancer. However, the method suffers high false alarm rate for benign instances. One way to mollify this is to relax the classification criterion on the number of "0" each instance receives from its mask. It is clear from Figure 2 that more maligns tend to have more "0" than benigns. If, for instance, 2 is chosen as the new classification criterion, the accuracy becomes 58.5% for malign and 80.3% for benign.

ESANN 2011 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2011, i6doc.com publ., ISBN 978-2-87419-044-5.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100817300.

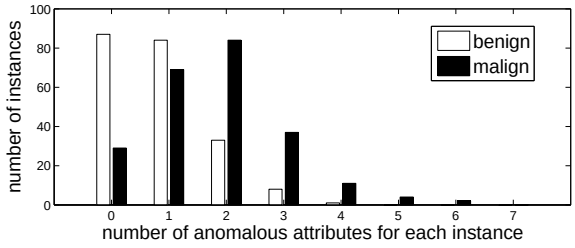

Fig. 2: Comparison of two classes on the number of anomalous attributes.

5 Conclusion

When dealing with anomaly detection in categorical data sets, most methods, including ours, are based on threshold setting in conditional, marginal and joint probabilities. Instead of manually defining those probabilities, our method expresses them by using a joint probability factorization according to a Bayesian network automatically learnt from data. Mask generation shows one of many possible approaches to set up a threshold on probabilities to differentiate normal attributes from anomalous ones.

Our method does not need any *a priori* knowledge on the relationships among categorical attributes. It might be an interesting direction in the future to enhance the method if *a priori* knowledge is available. In addition, the bounding hypercube may be generalized into other geometric shapes such as a convex hull that allows more flexibility in detecting anomalous attributes.

References

[1] D. Koller and N. Friedman, Probabilistic Graphical Models: Principles and Techniques, MIT Press, 2009.

[2] Chandola Varun and Banerjee Arindam and Kumar Vipin, Anomaly detection: A survey, *ACM Comput. Surv.*, 41:1–41:58, ACM, 2009.

[3] Giudici Paolo and Castelo Robert, Improving Markov Chain Monte Carlo Model Search for Data Mining, *Machine Learning*, 50:127–158, Springer Netherlands, 2003.

[4] David Madigan and Jeremy York, Bayesian Graphical Models for Discrete Data, *International Statistical Review*, 63:216–232, International Statistical Institute, 1995.

[5] Stephen P. Brooks and Andrew Gelman, General methods for monitoring convergence of iterative simulations, *Journal of Computational and Graphical Statistics*, 7:434–455,1998.

[6] Kaustav Das, Detecting anomalous records in categorical datasets, *Proceedings of the* 13^{th} *ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 220–229, ACM, 2007.

[7] Weng-Keen Wong and Andrew Moore and Gregory Cooper and Michael Wagner, Bayesian network anomaly pattern detection for disease outbreaks, *Proceedings of the* 20^{th} *International Conference on Machine Learning*, page 808–815, 2003.

[8] Philip K. Chan and Matthew V. Mahoney and Muhammad H. Arshad, A machine learning approach to anomaly detection, Technical Report, Department of Computer Sciences, Florida Institute of Technology, FL 32901, March, 2003.