FINDING DISTANCE-BASED OUTLIERS IN SUBSPACES THROUGH BOTH POSITIVE AND NEGATIVE EXAMPLES

Fabio Fassetti and Fabrizio Angiulli
DEIS, University of Calabria, Italy

Keywords: Data mining, Example-based outlier detection, Genetic algorithms.

Abstract: In this work an *example-based* outlier detection method exploiting both positive (that is, outlier) and negative (that is, inlier) examples in order to guide the search for anomalies in an unlabelled data set, is introduced.
The key idea of the method is to find the subspace where positive examples mostly exhibit their outlierness while at the same time negative examples mostly exhibit their inlierness. The degree to which an example is an outlier is measured by means of well-known unsupervised outlier scores evaluated on the collection of unlabelled data.
A subspace discovery algorithm is designed, which searches for the most discriminating subspace. Experimental results show that the method is able to detect a near optimal solution, and that the method is promising from the point of view of the knowledge mined.

1 INTRODUCTION

Unsupervised outlier detection techniques search for the objects most deviating from the data population they belong to. These techniques are employed on unlabelled data sets, that is when no a priori information about what should be considered normal and what should be considered exceptional is available, and outliers are singled out on the basis of certain *outlier scores* that can be assigned to each single object.

However, in addition to the unlabelled data set, very often also *examples of normality* and *examples of abnormality* are available. In this scenario it is then of interest to modify the mining technique in order to take advantage of these examples.

In this work an *example-based* outlier detection method exploiting both positive (that is, outlier) and negative (that is, inlier) examples in order to guide the search for anomalies in an unlabelled data set, is introduced. The task here introduced is novel, in that previous methods are able to exploit only positive examples. The key idea of the method is to find the subspace where positive examples mostly exhibit their outlierness while at the same time negative examples mostly exhibit their inlierness.

The method can be useful when a small amount of labelled data is available, e.g. a few patients for which an ascertained diagnosis is known, and the individuals to be single out are anomalous, that is their occurrence frequency is very low, e.g. consider people affected by a rare disease.

The degree to which an example is an outlier is measured by means of well-known unsupervised outlier scores evaluated on the collection of unlabelled data. A distance-based unsupervised outlier scores is employed, that is the mean distance of the object from its k nearest neighbors (Angiulli and Pizzuti, 2002). A subspace is then deemed to comply with the provided examples if a separation criterium between outlier scores associated with positive examples and outlier scores associated with negative examples is satisfied, and, moreover, the difference between the former and the latter ones is positive.

The most discriminating subspace is that which maximizes the above difference. Note that this measure is not monotonic with respect to subspace containment. While from a semantic point of view this property can be considered a desideratum, from the algorithmic point of view the above property makes very difficult to guide search towards the right subspace.

A subspace discovery algorithm is designed, which searches for the most discriminating subspace. As already noted, finding this subspace is a formidable problem due to the huge search space, while the non-monotonicity of the measure to op-

Fassetti F. and Angiulli F. (2010).
FINDING DISTANCE-BASED OUTLIERS IN SUBSPACES THROUGH BOTH POSITIVE AND NEGATIVE EXAMPLES.
In *Proceedings of the 2nd International Conference on Agents and Artificial Intelligence - Artificial Intelligence*, pages 5-10
DOI: 10.5220/0002699600050010

timize makes difficult to alleviate the cost of the search. The introduced mining technique is based on the paradigm of genetic algorithms, which are able to provide good approximate solutions to the problem of optimizing a multidimensional objective function.

The rest of the work is organized as follows. In the rest of this section, work related to the one here presented is briefly surveyed and major differences are pointed out. In Section 2, the novel task tackled with in this work is formally defined. Subsequent Section 3 presents the *ExampleBasedOutlierDetection* algorithm. Section 4 describes experiments on both synthetic and real data sets. Finally, Section 5 draws conclusions and future work.

1.1 Related Work

Next some outlier detection methods working on subspaces and/or exploiting examples are briefly recalled. Contributions of this work are clarified by pointing out differences with related methods while discussing them.

The work (Aggarwal and Yu, 2001) detects anomalies searching for subspaces in which the data density is exceptionally lower than the mean density of the whole data. Promising subspaces are detected by employing a technique based on genetic algorithms. Although this method works on the subspaces, it does not contemplate the presence of examples.

In (Zhang and Wang, 2006) the interest is on searching for the subspaces in which the sum of the distances between a fixed object and its *nearest neighbors* exceeds a given threshold. A dynamic subspace search exploiting sampling is presented and compared with top-down and bottom-up like techniques. This work exploits only one positive example and it has no negative ones. Furthermore, subspaces in which the example is exceptional are searched for, while discovery of additional outliers is not accomplished.

The work (Wei et al., 2003) focuses on discovering sets of categorical attributes, called *common attributes*, being able to single out a portion of the data base in which the value assumed by an object on a single additional attribute, called *exceptional attribute*, becomes infrequent with respect to the mean of the frequencies of the values assumed by the same attribute. Common attributes are determined by selecting the sets of frequent attributes of the data base.

In (Zhu et al., 2005) the *Outlier by Example* method is introduced. Given a data set and user-provided outlier examples, the goal of the method is to find the other objects of the data set exhibiting the same kind of exceptionality. Data set objects are mapped into the MDEF feature space (Papadimitriou et al., 2003), and both user-provided examples and outstanding outliers, i.e. those that can be regarded as outliers at some granularity level, are collected to form the positive training data. Then the SVM algorithm is employed in order to build a classifier separating the normal data from the positive training data. This technique employs only positive examples, is based on the MDEF measure, and does not work on subspaces, but instead searches for anomalies in the full feature space.

In (Zhu et al., 2005), given an input set of example outliers, i.e. of objects known to be outliers, the authors search for the objects of the data set which mostly exhibit the same exceptional characteristics. In order to single out these objects, they search for the subspace maximizing the average value of sparsity coefficients, that is the measure introduced in (Aggarwal and Yu, 2001), of cubes containing user examples. This method is suited only for numerical attributes, it is based on the notion of sparsity coefficient, which is different from the notion of distance-based score, and it can take advantage only of positive examples, while negative ones are not considered. Moreover, it must be noted that the sparsity coefficient is biased towards small subspaces. Indeed, in order to prefer larger ones it should take place that the number of objects is exponentially related to the number of attributes, a very unlikely situation.

2 PROBLEM STATEMENT

First some preliminary definitions are provided, and then the example-based outlier score is introduced.

A *feature* is an identifier with an associated domain. A *space* F is a set of features. An *object* of the space F is a mapping among features $A \in F$ and values in the domain of A. The value of the object o on the feature $A \in F$ is denoted by o_A. A *subspace* S of F is any subset of F. The *projection* of the object o in the subspace S, denoted by o^S, is an object of the space S such that $o^S_A = o_A$, for each $A \in S$. Note that $o^F = o$. The projection of a set of objects O in the subspace S, denoted by O^S, is $\{o^S \mid o \in O\}$.

A *distance* dist on the space F is a semimetric defined on each pair of objects of each subspace of F, that is a real-valued function which satisfies the non-negativity, identity of indiscernibles and symmetry axioms.

Let a set of objects DS of the space F, called *data set* in the following, be available. Let $K \geq 1$ be an integer. The *K-th nearest neighbor* of o^S (in the data set DS), denoted by $nn_K(o^S)$, is the object p of DS

such that there exist exactly $K-1$ objects q of DS with $\text{dist}(o^S, q^S) \leq \text{dist}(o^S, p^S)$.

Outlier Score. In this work, we employ a well-established distance-based measures of outlierness, also said *outlier score* in the following.

The *outlier score* $os(o)$ of o is defined as follows (Angiulli and Pizzuti, 2002):

$$os(o) = \frac{1}{K}\sum_{i=1}^{K} \text{dist}(o, nn_i(o)).$$

The outlier score is given by the sum of the distances between o and its K nearest neighbors in the data set. Its value provides an estimate of the data set density in the neighborhood of the object o. The objects o scoring the greatest values of outlier score $os(o)$ are also called *outliers*, since they be considered anomalous with respect to the population under consideration.

Let E be a set of objects. The *outlier score* $sc(E)$ of E is defined as the mean of the outlier scores associated with the elements of E:

$$sc(E) = \frac{1}{|E|}\sum_{e \in E} os(e).$$

Subspace Score. Assume a set O of *outlier examples* (or *positive examples*) and a set I of *inlier examples* (or *negative examples*) are available.

We are interested in finding subspaces where the outlier examples deviate from the data set population, the inlier examples comply with the data set population, and the separation between these examples is large.

In order to formalize the above intuition, the following definition of consistent (with respect to a set of positive and negative examples) subspace is needed.

We say that a subspace S is ρ-*consistent*, or simply *consistent*, where $\rho \in [0,1]$ is a user-provided parameter, with respect to a set O of positive examples and a set I of negative examples, if the ρ percent of the objects in O^S, that are the positive examples O projected in the subspace S, is globally more outlying than the set of objects in I^S, that are the negative examples I projected in the subspace S, while the remaining $1-\rho$ percent of the objects in O^S is individually more outlying than all the objects in I^S, that is to say,

1. $sc(O_b^S) > sc(I^S)$, where O_b is the set of the $\lceil \rho|O| \rceil$ objects o of O having the smallest outlier scores $os(o^S)$, and
2. $os(o^S) > \max_{i \in I} os(i^S)$, for each $o \in (O - O_b)$.

where the first condition does not apply if $\rho = 0$ or O_b is empty, and, dually, the second condition does not apply if $\rho = 1$ or $O - O_b$ is empty.

In order to measure the relevance of the subspace S with respect to the above criterium, next the concept of subspace score is introduced. The *subspace score* $ss(S)$ of the space S with respect to set of positive examples O and set of negative examples I is

$$ss(S) = \begin{cases} sc(O^S) - sc(I^S) & \text{, if } S \text{ is } \rho\text{-consistent w.r.t. } O \text{ and } I \\ 0 & \text{, otherwise} \end{cases}$$

Note that for a consistent subspace S, the corresponding subspace score $ss(S)$ is always positive.

Moreover, it is worth to point out that the subspace score is not monotonic with respect to subspace containment.

Outliers by Example Problem. We are now in the position of defining the main task we are interested in.

Given an integer $n \geq 1$, and a subspace S, the top-n outliers of DS in S are the n objects o of DS with maximum value of outlier score $os(o^S)$.

The *outlying subspace* S^{ss} is defined as

$$\arg\max_S ss(S).$$

Given a data set DS, a set of positive examples O, a set of negative examples I, and a positive integer number n, the *Distance-Based Outlier Detection by Example Problem* is defined as follows: find the top-n outliers in the outlying subspace S^{ss}.

3 ALGORITHM

Finding the outlying subspace is in general a formidable problem. We decided to face it by exploiting the paradigm of *genetic algorithms* (Holland et al., 1986; Holland, 1992), a methodology also pursued by other subspace finding methods for outlier detection (Aggarwal and Yu, 2001; Zhu et al., 2005). Genetic algorithms are based on the theory of evolution and they are probabilistic optimization methods based on the principles of evolution. These algorithms have been successfully applied to different optimization tasks. In the optimization of non-differentiable or even discontinuous functions and discrete optimization they outperform traditional methods since derivatives provide misleading information to conventional optimization methods.

Genetic algorithms maintain a population of potential solutions. In our context, a potential solution is a subspace and it is encoded by means of a binary string, also said a *chromosome*, of length $|F|$. The ith bit of the binary string being 1 (0, resp.) means that the ith feature of F is (is not, resp.) in the subspace encoded by the chromosome. At each iteration a *fitness* value is associated with each chromosome, representing a measure of the goodness of the potential

Algorithm *ExampleBasedOutlierDetection*
Input: data set DS on the set of features F, set O of positive examples, set I of negative examples, number K of nearest neighbors to consider, number n of top outliers to return, parameter ρ
Output: the example-based outliers of DS

1. Let P the initial population of subspaces having size M, obtained by selecting at random M subsets of the overall set of features F
2. While the convergence criterion is not meet do
   (a) For each subspace S in P, determine if S is already stored in the hash table *SSTable* and, in the positive case, retrieve its fitness value
   (b) Let $P_{new} = \{S_1, \ldots, S_m\}$ be the subset of P composed of the subspaces which are not stored in *SSTable*
   (c) For each negative example i in $I = \{i_1, \ldots, i_{N_I}\}$, determine simultaneously the outlier scores $\{os(i^{S_1}), \ldots, os(i^{S_m})\}$
   (d) Let B denote the number $\lceil \rho |O| \rceil$, and let $\alpha_1, \ldots, \alpha_m$ ($\beta_1, \ldots, \beta_m$, resp.) denote the maximum (mean, resp.) outlier scores associated with the negative examples in the subspaces $S_1, \ldots, S_m$, respectively, that is $\alpha_j = \max_{i \in I} os(i^{S_j})$ ($\beta_j = sc(I^{S_j})$, resp.), for $j = 1, \ldots, m$
   (e) For each positive example o_k in $O = \{o_1, \ldots, o_{N_O}\}$ do
      i. Determine simultaneously the outlier scores $\{os(o_k^S) \mid S \in P_{new}\}$
      ii. For each subspace S_j in P_{new} do
         A. Let $O_{k,j}$ be the set composed of precisely the B objects o of $\{o_1, \ldots, o_k\}$ having the smallest outlier scores $os(o^{S_j})$, and let $o_{k,j}$ be the object having the $(B+1)$–th smallest outlier score $os({o_{k,j}}^{S_j})$
         B. If either (1) $\alpha_j \geq os({o_{k,j}}^{S_j})$ or (2) $\beta_j \geq sc(O_{k,j}^{S_j})$, then set $P_{new} = P_{new} - \{P_j\}$, and set the fitness of the subspace P_j to zero and store it in the hash table *SSTable*
   (f) For each subspace S remained in P_{new}, compute its fitness as $sc(O^S) - s(I^S)$ and store it in the hash table *SSTable*
   (g) From the set P, select M pairs $\langle S_1^1, S_1^2 \rangle, \ldots \langle S_M^1, S_M^2 \rangle$ of parent subspaces for the next generation (*selection* step)
   (h) Compute the set of subspaces $P_{next} = \{S'_1, \ldots, S'_M\}$, where each subspace S'_j is obtained by crossover of the parent subspaces S_j^1 and S_j^2, for $i = 1, \ldots, M$ (*crossover* step)
   (i) Mutate some of the subspaces in the set P_{next} (*mutation* step)
   (j) Set the current population P to the next generation P_{next}
3. Select the subspace S^{ss} in P scoring the maximum fitness value
4. Determine the top-n outliers in the subspace S^{ss} and return them as the set of the example-based outliers

Figure 1: The *ExampleBasedOutlierDetection* algorithm.

solution. The current population is iteratively updated by means of the selection, crossover, and mutation mechanisms till a convergence is meet. *Selection* is a mechanism for selecting chromosomes for reproduction according to their fitness. *Crossover* denotes a method of merging the genetic information of two individuals; if the coding is chosen properly, two good parents produce good children. In genetic algorithms, *mutation* can be realized as a random deformation of the strings with a certain probability. The positive effect is preservation of genetic diversity and, as an effect, that local maxima can be avoided.

Figure 1 shows the algorithm *ExampleBasedOutlierDetection* which solves the *Outliers by Example Problem*. We employed the subspace score as fitness function for the genetic algorithm. Since computing the subspace score is expensive, some optimizations are accomplished in order to practically alleviate its cost, which are explained next.

First of all, an hash table *SSTable* of size T maintains the latest T subspaces visited by the algorithm, together with their fitness, and with a timestamp which is exploited to implement the insertion policy. This table is used as follows. Before computing the fitness associated to a subspace, it is searched for in the hash table. If the subspace is found, then its timestamp is updated and then the fitness stored in the table is employed. Vice versa, when a novel subspace has to be stored in the hash table, but no more space is available in the selected entry, the timestamps are exploited in order to determine the subspace (that is, the oldest one) that will be replaced with the latest subspace.

In this work we employed the Euclidean distance as distance function. Let $S_1, \ldots, S_m$ the subspaces of the current population which are not already stored

in *SSTable*. In order to save distance computations, the outlier scores $os(e^{S_1}), \ldots, os(e^{S_m})$ associated with a positive or negative example e are computed simultaneously as follows: first the set U is computed as $S_1 \cup \ldots \cup S_m$ and, for each $A \in U$, the values $d_A = (x_A - y_A)^2$ are obtained, and then the distances $\text{dist}(x^{S_j}, y^{S_j})$ are computed as $\sqrt{\sum_{A \in S_j} d_A}$.

As a further optimization, the outlier scores associated with the negative examples are computed first (see steps 2(c) and 2(d)). Then, while computing outlier scores associated with positive examples (see step 2(e)), the outlier scores of the negative ones are immediately exploited in order to filter out subspaces which are not ρ-consistent (see step 2(e)ii) and, hence, avoiding useless distance computations.

As selection-crossover-mutation strategies we used proportional selection, one-point crossover, and mutation by inversion of a single bit, while as convergence criterion was used an a-priori fixed number of iterations, also said *generations* (Holland, 1992).

As far as the temporal complexity of the algorithm is concerned, say N the number of data set objects, N_E the total number of examples, d the number of features in the space F, and g the number of generations. In the worst case, for each generation in order to determine outlier scores the distances among all the examples and all the data set objects are computed, with a total cost $O(g * N_E * N * d)$. After having determined the outlying subspace S^{ss}, in order to compute the top-n outliers in that subspace, all the pairwise distances among data set objects are to be computed, and, then, the top-n outliers are to be singled out, with a total cost $O(N^2 * d)$. Summarizing, the temporal cost of the algorithm *ExampleBasedOutlierDetection* is $O(g * N_E * N * d + N^2 * d)$.

4 EXPERIMENTAL RESULTS

In the experiments reported in the following, if not otherwise specified, the crossover probability was set to 0.9 and the mutation probability was set to 0.01. Moreover, the parameter ρ, determining the "degree" of consistency of the subspace, was set to 0.1.

First of all, we tested the ability of the algorithm to compute the optimal solution (that is the outlying subspace). With this aim, we considered a family of synthetic data sets, called *Synth* in the following.

Each data set of the family is characterized by the size D of its feature space. Each data set consists of 1,000 real dimensional vectors in the D-dimensional Euclidean space, and is associated with about D positive examples and D negative examples. Examples are placed so that the outlying subspace coincides with a randomly selected subspace having dimensionality $\lceil \frac{D}{5} \rceil$.

We varied the dimensionality D from 10 to 20 and run our algorithm three times on each data set. We recall that the size of the search space exponentially increases with the number of dimensions D. We set the population size to 50 and the number of generations to 50 in all the experiments. The parameter K was set to 10.

Table 1 reports the results of these experiments. Interestingly, the algorithm always found the optimal solution in at least one of the runs. Up to 15 dimensions it always terminated with the right outlying subspace. For higher dimensions it reported also some different subspaces, but in all cases the solution returned is a suboptimal one. Indeed, the second and third solutions concerning the data set *Synth18D* are subsets of the optimal solution both having only a single missing feature, while the second solution concerning the data set *Synth20D* is a superset of the optimal one having two extra features. By these experiments it is clear that the method is able to return the optimal solution or a suboptimal one.

The subsequent experiment was designed to validate the quality of the solution returned by the proposed method. In this experiment we considered the Wisconsin Diagnostic Breast Cancer data set from the UCI Machine Learning Repository. This data set is composed of 569 instances, each consisting in 30 real-valued attributes, grouped in two classes, that are *benign* (357 instances) and *malignant* (212 instances). The thirty attributes represent mean, standard error, and largest value associated with the following ten cell nucleus features: radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, and fractal dimension.

We normalized the values of each attribute in the range $[0,1]$. Moreover, we randomly selected ten benign instances as the set of negative examples I_{wdbc} and twenty malignant instances as the set of positive examples O_{wbdc}. Moreover, we built a data set DS_{wdbc} of 357 objects by merging together all the remaining benign instances (that are 347) with other ten randomly selected malignant examples, say them DS^O_{wdbc}.

We set the number of neighbors K to 50, and the number of top outliers n to 20. First of all, we computed the distance-based outliers in the full feature space. We found that among the top twenty outliers, six of them belong to the set DS^O_{wdbc} (corresponding to the 60% of DS^O_{wdbc}). Next, we run the *ExampleBasedOutlierDetection* algorithm. The outlying subspace S^{ss}_{wdbc} found was composed of seventeen features. In

Table 1: Experimental results on the synthetic data set family.

Dataset	Outlying subspace	Outlier score	Algorithm output	Outlier score
Synth10D	0000100001	1.121307	0000100001	1.121307
			0000100001	1.121307
			0000100001	1.121307
Synth12D	101000010000	1.428615	101000010000	1.428615
			101000010000	1.428615
			101000010000	1.428615
Synth15D	000010011000000	1.522407	000010011000000	1.522407
			000010011000000	1.522407
			000010011000000	1.522407
Synth18D	000100000010001100	1.667848	000100000010001100	1.667848
			000100000010001000	1.424176
			000100000010001000	1.424176
Synth20D	00011000000001000010	1.701322	00011000000001000010	1.701322
			00011000100001000011	0.995888
			00011000000001000010	1.701322

this subspace, nine objects of the set DS^{O}_{wdbc} belong to the top twenty distance-based outliers of DS (that is the 90%).

Thus, by exploiting our method we singled out a subspace in which the anomalies detected by using the distance-based definition are of better quality with respect to those detected in the full feature space by using the same definition.

5 CONCLUSIONS

We presented an example-based outlier detection method exploiting both positive and negative examples in order to search for anomalies in an input data set. The task here introduced is novel, in that previous methods are able to exploit only positive examples, and, moreover, are based on different outlier definitions. We presented a subspace discovery algorithm designed to search for the optimal subspace, and experiments showed that the method is able to detect a suboptimal solution, and that the method is promising from the point of view of the knowledge mined.

As a future work, it is of interest to investigate the inclusion in our framework of other outlier definitions, and the design of policies for selecting outliers in the outlying subspace guided by the examples. Finally, we plan to execute a more extensive experimental campaign concerning both from the computational and the semantic point of view.

REFERENCES

Aggarwal, C. C. and Yu, P. (2001). Outlier detection for high dimensional data. In *Proc. Int. Conference on Managment of Data*.

Angiulli, F. and Pizzuti, C. (2002). Fast outlier detection in large high-dimensional data sets. In *Proc. Int. Conf. on Principles of Data Mining and Knowledge Discovery*, pages 15–26.

Holland, J. (1992). *Adaptation in Natural and Artificial Systems*. The MIT Press, Cambridge, MA.

Holland, J., Holyoak, K., Nisbett, R., and Thagard, P. (1986). *Computational Models of Cognition and Perception*, chapter Induction: Processes of Inference, Learning, and Discovery. The MIT Press, Cambridge, MA.

Papadimitriou, S., Kitagawa, H., Gibbons, P. B., and Faloutsos, C. (2003). Loci: Fast outlier detection using the local correlation integral. In *ICDE*, pages 315–326.

Wei, L., Qian, W., Zhou, A., Jin, W., and Yu, J. (2003). Hot: Hypergraph-based outlier test for categorical data. In *Proc. of the Pacific-Asia Conf. on Knowledge Discovery and Data Mining*, pages 399–410.

Zhang, J. and Wang, H. (2006). Detecting outlying subspaces for high-dimensional data: the new task, algorithms, and performance. *Knowledge and Information Systems*, to appear.

Zhu, C., Kitagawa, H., and Faloutsos, C. (2005). Example-based robust outlier detection in high dimensional datasets. In *Proc. Fifth IEEE International Conference on Data Mining*, pages 829–832.