

Who Is Missing? A New Pattern Recognition Puzzle

Ludmila I. Kuncheva and Aaron S. Jackson

School of Computer Science,
Bangor University, United Kingdom

Abstract. Consider a multi-class classification problem. Given is a set of objects, for which it is known that there is at most one object from each class. The problem is to identify the missing classes. We propose to apply the Hungarian assignment algorithm to the logarithms of the estimated posterior probabilities for the given objects. Each object is thereby assigned to a class. The unassigned classes are returned as the solution. Its quality is measured by a consistency index between the solution and the set of known missing classes. The Hungarian algorithm was found to be better than the rival greedy algorithm on two data sets: the UCI letter data set and a bespoke image data set for recognising scenes with LEGO parts. Both algorithms outperformed a classifier which treats the objects as iid.

Keywords: Pattern recognition, set classification, Bayes-optimal classifier, Hungarian algorithm.

1 Introduction

Who is missing? At your lecture, you are taking a class register from a single snapshot of the audience. If the number of students in the audience is the same as the size of your class, and you are satisfied that there are no impostors, then all your program needs to do is count the faces in the snapshot. However, if the number of attendees is smaller than the class list, you will need to find out *who* the missing students are.

Suppose that you have a trained classifier to recognise the students' faces. If the face detection program and the classifier were ideal, all faces would be correctly detected and recognised, hence the missing students will be identified instantly. However, if the classifier is imperfect, classifying each face individually may not be the optimal strategy. First, individual labelling will not prevent assigning the same label to several objects. Second, individual labelling cannot take into account any class dependencies. For example, suppose that students X and Y are friends, and are always present or absent together. Individual labelling will not be able to take advantage of this piece of knowledge. Therefore, some form of set classification would be a more prudent strategy.

One of the standard assumptions in classical pattern recognition is that the data points to be classified come as an independent identically distributed (iid)

sequence. In many problems this assumption does not hold. For examples of non-iid classification paradigms are listed below.

1. *The multiple-instance Problem.* This problem arises in complex machine learning applications where the information about the instances is incomplete or ambiguous [4, 7, 13, 19], e.g., in drug activity prediction [4]. The training examples come in “bags” labelled either positive or negative. For a positive bag, it is known that at least one instance in the bag has true positive label. For a bag labelled negative, all instances are known to be negative. The problem is to design a classifier that can label as accurately as possible an unseen bag of instances.
2. *Set Classification.* In this problem, all the instances in a set are assumed to have come from the same unknown class [16]. This problem may arise in face recognition where multiple images of the same person’s face are submitted as a set.
3. *Collective Recognition.* In this scenario, a set of instances are labelled together [14, 18]. The crucial assumption is that the instances within the set are related, so that the dependencies can be used to improve the classification accuracy. For examples, in classifying web pages into topic categories, hyperlinked web pages are more likely to share common class labels than non-linked pages [18].
4. *Full-Class Set Classification* [11]. Here a set of instances has to be classified together, knowing that the set contains at most one instance from each class. In other words, the c objects must be matched one-to-one to the c classes. We can call this problem ‘who-is-who’ to distinguish it from the ‘who-is-missing’ problem. Simultaneous classification of a set of instances has been used in tracking. For example, a moving object can be regarded as a patchwork of parts [1] or a set of tracklets [10], which are matched from one image frame to the next. This fits within the framework considered here because each part/tracklet on the object can be referred to as a class label, and the segmented pieces in the image have to be distributed to the different class labels. However, the instances within the set are not iid, as the parts are spatially linked within the object, and also follow physical motion laws. Other potential applications include karyotyping (labelling the chromosomes in a cell) [9, 15] and identifying footballers on the pitch in a live-play video [3].

The who-is-missing problem is a variant of paradigm #4, the full set classification [11]. In this study we formulate the *who-is-missing* problem and propose a solution based on the Hungarian assignment algorithm.

2 The Who-Is-Missing Problem

2.1 Problem Description

Let $\Omega = \{\omega_1, \dots, \omega_c\}$ be the set of class labels. The set of objects presented for classification is $\mathbf{Z} = \{\mathbf{z}_1, \dots, \mathbf{z}_k\}$, where $k < c$, and the true labels of the objects

in \mathbf{Z} , denoted $\{y_1, \dots, y_k\}$ are all different. The task is to find the set of missing classes, that is the set

$$\Omega_{(-)} = \Omega \setminus \Omega_{(+)}, \quad (1)$$

where

$$\Omega_{(+)} = \bigcup_{i=1}^k y_i. \quad (2)$$

Because of the strict inequality $k < c$, $\Omega_{(-)}$ is a non-empty set.

Denote by $P(\omega_j|\mathbf{x})$ the probability that the true class label for an observed \mathbf{x} is $\omega_j \in \Omega$. We can arrange the posterior probabilities in a $k \times c$ matrix

$$P = \begin{bmatrix} P(\omega_1|\mathbf{z}_1) & \dots & P(\omega_c|\mathbf{z}_1) \\ \vdots & & \vdots \\ P(\omega_1|\mathbf{z}_k) & \dots & P(\omega_c|\mathbf{z}_k) \end{bmatrix}. \quad (3)$$

The task is to determine the $c-k$ missing classes based on P and the knowledge that labels y_1, \dots, y_k are different. The probability that the class label for a given \mathbf{x} is *not* ω_j is $1 - P(\omega_j|\mathbf{x})$. If \mathbf{Z} contained k iid objects, the probability that class ω_i is not represented in \mathbf{Z} would be

$$P_{\text{iid}}(\sim \omega_i|\mathbf{Z}) = \prod_{j=1}^k (1 - P(\omega_i|\mathbf{z}_j)). \quad (4)$$

The $c-k$ classes with the largest P_{iid} should be returned as $\Omega_{(-)}$. This approach, however, is based on the false iid assumption and hence does not take advantage of the fact that the elements of \mathbf{Z} have different class labels.

Accurate identification of the missing classes is equivalent to accurate assignment of the present classes. Therefore the solution can be found using the Hungarian assignment algorithm¹. Proposed originally for $c \times c$ matrices, the algorithm was extended for rectangular matrices [2].

It has been shown [11] that the Bayes-optimal solution of the who-is-who problem ($k = c$) is the permutation of labels $\langle s_1, s_2, \dots, s_c \rangle$ which maximises the criterion

$$\sum_{i=1}^c \log P(\omega_{s_i}|\mathbf{z}_i). \quad (5)$$

The underlying assumption is that the object from each class is picked independently of the objects from the other classes.

2.2 An Example

As an example, consider three objects, \mathbf{z}_1 , \mathbf{z}_2 and \mathbf{z}_3 , coming from three classes, ω_1 , ω_2 and ω_3 . The class assignment of the three objects is not known, apart from the fact that there is one object from each class. Let P be

¹ Further developed by Kuhn and Munkres, also known as Kuhn-Munkres algorithm.

$$P = \begin{bmatrix} 0.65 & 0.07 & 0.28 \\ 0.43 & 0.50 & 0.07 \\ 0.24 & 0.57 & 0.19 \end{bmatrix}. \tag{6}$$

Table 1 shows all possible label permutations, the corresponding posterior probabilities and the sum-log criterion value (5) for each permutation. In addition, the sum of the posterior probabilities is also shown.

Table 1. An example of the class assignment problem

Class			Posteriors			$\sum \log()$	sum
			\mathbf{z}_1	\mathbf{z}_2	\mathbf{z}_3		
3	2	1	0.28	0.50	0.24	-3.3932	1.02
3	1	2	0.28	0.43	0.57	-2.6791	1.28
2	3	1	0.07	0.07	0.24	-6.7456	0.38
2	1	3	0.07	0.43	0.19	-5.1640	0.69
1	2	3	0.65	0.50	0.19	-2.7847	<u>1.34</u>
1	3	2	0.65	0.07	0.57	-3.6522	1.29

According to the table, the best solution is $\mathbf{z}_1 \in \omega_3$, $\mathbf{z}_2 \in \omega_1$ and $\mathbf{z}_3 \in \omega_2$. Interestingly, this is not the solution which maximises the sum of the posterior probabilities.

A greedy approach would assign class ω_1 to \mathbf{z}_1 (0.65), next assign class ω_2 to \mathbf{z}_3 (0.57), and finally assign class ω_3 to the remaining object \mathbf{z}_1 (0.07). This permutation, (1,3,2), is ranked 4th of 6 on the sum-log criterion.

2.3 Proposed Solution

We propose to use the Hungarian assignment algorithm to a full $c-by-c$ matrix where $c-k$ objects will be “dummy” objects. Their respective rows with posterior probabilities are filled with values $\frac{1}{c}$, indicating a complete lack of preference of a class label. The class labels assigned by the algorithm to the dummy objects will be the missing classes.

The hypothesis is that the Hungarian algorithm will provide better solution to the who-is-missing problem compared to a greedy algorithm or independent classification that assumes iid data.

2.4 Evaluation of the Solution

To find out how successful the proposed strategy is, we need a measure of match between the true missing classes and the obtained missing classes. Simple measures based on the intersection between the two sets will not be adequate because such measures will depend on the number of the missing classes and are not corrected for chance. Therefore, we propose to use a *consistency index* [12].

The Consistency Index $I_C(A, B)$ for two subsets $A \subset X$ and $B \subset X$, such that $|A| = |B| = k$, where $0 < k < |X| = n$, is defined as

$$I_C(A, B) = \frac{r - \frac{k^2}{n}}{k - \frac{k^2}{n}} = \frac{rn - k^2}{k(n - k)}. \quad (7)$$

where $r = |A \cap B|$. The maximum value of the index, $I_C(A, B) = 1$, is achieved when $r = k$. The minimum value of the index is bound from below by -1 . The limit value is attained for $k = \frac{n}{2}$ and $r = 0$. Note that $I_C(A, B)$ is not defined for the trivial cases $k = 0$ and $k = n$. They are not interesting from the point of view of comparing subsets, so the lack of values for $I_C(A, B)$ in these cases is not important. Finally, $I_C(A, B)$ will assume values close to zero for independently drawn A and B .

The hypothesis will be supported if the consistency index for the results from the Hungarian algorithm is higher than the indices for the rival methods.

3 Experiments

The purpose of this experiment is rather a proof of concept than comparison of possible alternatives. Both the Hungarian and the greedy algorithms were applied with criterion (5).²

Suitable data sets for the who-is-missing problem should have a large number of classes.

3.1 Letter Data Set

We chose the Letter data set from the UCI Machine Learning Repository [5]. The number of classes is 26 (letters from the Latin alphabet), and the number of objects is 20,000. The experimental protocol was as follows.

1. The data set was first standardised, and subsequently divided into a training part (the first 10,000 objects) and a testing part (the latter 10,000 objects).
2. A linear discriminant classifier was trained on the training part.³ This classifier was chosen on purpose so that there is sufficient scope for improvement. Both the training and the testing errors are approximately 30%.
3. A level of noise η was chosen from the set $\{0.0, 0.1, 0.2, \dots, 0.8\}$. Gaussian noise with mean zero and standard deviation η was added independently to each value in the testing data set. The perturbed testing data was classified using the classifier trained on the original training data. The posterior probabilities for all objects were stored.
4. The number of present classes k was chosen from the set $\{2, 3, \dots, 25\}$.

² Since the logarithm is a monotonic transformation, the greedy algorithm would give exactly the same result if applied straight on the posterior probabilities.

³ We used the `classify` function from the Statistics Toolbox of MATLAB.

5. 1000 runs were carried out with the chosen η and k . In each run, a random subset of k classes was sampled. One object was picked randomly from each of the present classes. The posterior probabilities for the selected objects (given by the classifier in Step 3) were retrieved and collated in matrix P . The matrix was augmented to 26×26 by adding $26 - k$ dummy rows with values $\frac{1}{26}$. The Hungarian and the Greedy algorithms were applied and the respective sets of missing classes (assigned to the dummy rows) were recorded. Let HM be the set of missing classes according to the Hungarian algorithm, GM , for the Greedy algorithm, and TM be the TRUE set of missing classes. We applied the original classifier assuming iid data. Let CM be the set of non-assigned classes. The respective values of the consistency index were calculated as

$$I_{\text{Hungarian}}(k, \eta) = I_C(TM, HM), \quad I_{\text{Greedy}}(k, \eta) = I_C(TM, GM)$$

and

$$I_{\text{Classifier}}(k, \eta) = \begin{cases} I_C(TM, CM), & \text{if } |CM| = |TM|, \\ 0, & \text{otherwise.} \end{cases}$$

The values of the consistency indices, averaged across the 1000 runs, for noise levels $\eta = 0$ and $\eta = 0.8$, are shown in Figure 1. The graphs for the remaining noise levels followed similar patterns. Plotted are also error bars spanning the 95% confidence intervals calculated from the 1000 values of the respective run.

The upward trend of the curves can be explained with the following argument. When only a few objects are missing, their correct labelling depends on the correct labelling of all the remaining objects. The scope for error is high. On the other hand, when only a few classes are present, there is less room for error in classifying all these objects correctly. Finally, when there is only one object present ($c - 1$ absent), all methods converge to the original classifier.

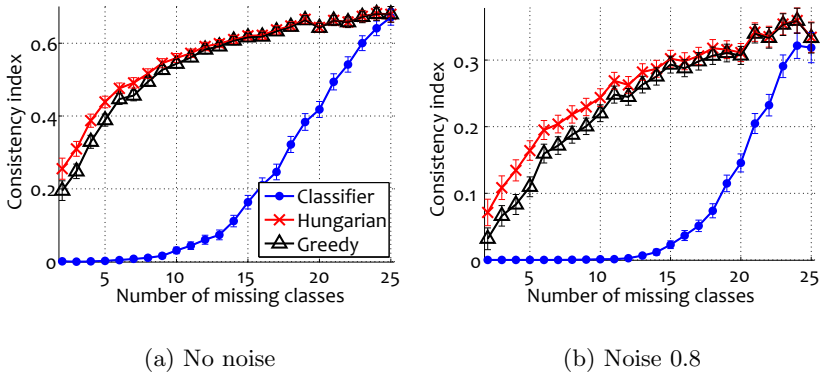


Fig. 1. Consistency indices for the sets of missing classes using the three algorithms on the Letter data set. The error bars indicate the 95% confidence intervals.

Clearly, the classifier alone is a poor choice according to the chosen measure. The Hungarian and the Greedy algorithms behave similarly but the Hungarian algorithm had an edge, giving support to our hypothesis. To demonstrate this finding, Figure 2 shows the results from a statistical test between the results of the two methods. We carried out a paired, two-sided test of the hypothesis that the difference between the matched 1000 samples of consistency indices comes from a distribution whose median is zero (Wilcoxon signed rank test). We chose to show the results as a heat map. The grey level intensity corresponds to the p -value. White indicates $p = 1$ and black, $p = 0$. Each square is the result from one comparison across the 1000 iterations for the respective number of missing classes $c - k$ and noise level η . The comparisons where there was significant difference at $\alpha = 0.05$ are marked with dots. As the Hungarian method was always superior or equivalent to the Greedy method, the dots mark the combinations of parameters where the Hungarian method wins.

As expected, larger noise level showcase the proposed method. This is shown by the larger number of dots in the top rows. For noise-free data (bottom row), the algorithms tie for a smaller number of missing classes. For large number of missing classes (rightmost column), the algorithms are similar.

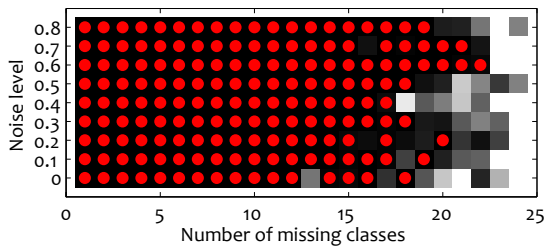


Fig. 2. Heat map of the p -value of the Wilcoxon signed rank test for equal medians of the consistency indices for the Hungarian and the Greedy algorithms. Dots signify statistically significant difference at $\alpha = 0.05$.

3.2 Who-Is-Missing: Objects in an Image

To illustrate the proposed solution in a real-life scenario, we took images of a set of 22 parts from a LEGO Mindstorms NXT kit (Figure 3). Each of the 28 images contained all 22 LEGO parts. After segmentation, 706 objects were detected, labelled, and saved as the training data. Five position-invariant and rotation-invariant features were extracted: eccentricity, solidity,⁴ and the RGB colour of the object.

Each feature was standardised to mean 0 and standard deviation 1. To evaluate the potential of the data set, we applied Principal Component Analysis (PCA) to the data set and plotted the 23 classes in the space of the first two

⁴ The `regionprops` MATLAB function was used.

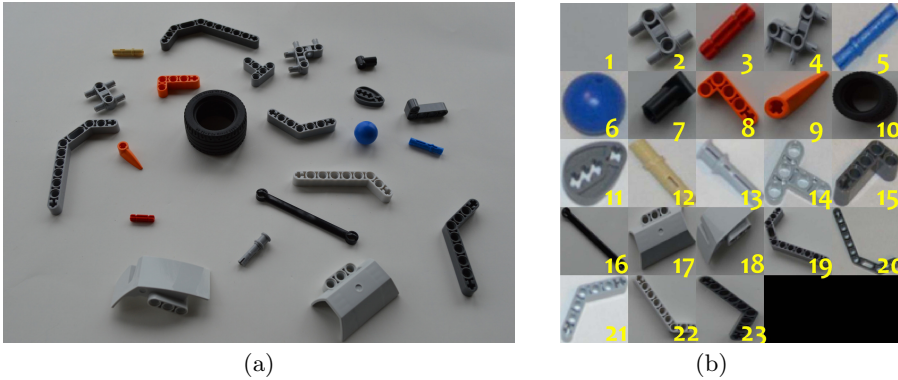


Fig. 3. (a) An image with all 22 LEGO pieces; (b) types of LEGO pieces with their class labels. Class 1 corresponds to “other”.

principal components (Figure 4). The plot indicates that the classes are highly overlapping, which will prevent the a classifier of iid objects from achieving a high consistency for the who-is-missing problem.

Next we ran 10-fold cross-validation experiments with a small selection of classifiers from WEKA [8], using the default parameter settings. The classification accuracy ranged from 16.28% for AdaBoost.M1 [6] to 82.72% for Rotation Forest [17], revealing that the data set is not too easy and, at the same time, high classification accuracy is possible. This suits our purposes, as an ideal classifier will not need an assignment algorithm to solve the who-is-missing problem.

We trained and tested the nearest mean classifier for the set, obtaining a rather mediocre testing classification accuracy of 38.63%.

A new set of 100 images was collected, 25 with two random missing class, 25 with three random missing classes, 25 with four missing classes and the last 25 with five missing classes. Each image was segmented and the features of the objects were extracted. To eliminate the effect of inaccurate segmentation on the comparison of the Hungarian and the greedy algorithms, we accepted for this analysis only images where the number of segmented objects tied with the true number of non-missing classes. The averaged consistency indices for the different number of classes are shown in Table 2.

Table 2. Average values of the consistency indices for the LEGO data and the p -value of the Wilcoxon signed rank test

	Number of missing classes				Mean	p -value
	2	3	4	5		
# images	15	22	10	11		
$I_{\text{Hungarian}}$	0.2667	0.3333	0.1444	0.2706	0.2716	0.0452
I_{Greedy}	0.1200	0.2632	0.1750	0.1529	0.1900	

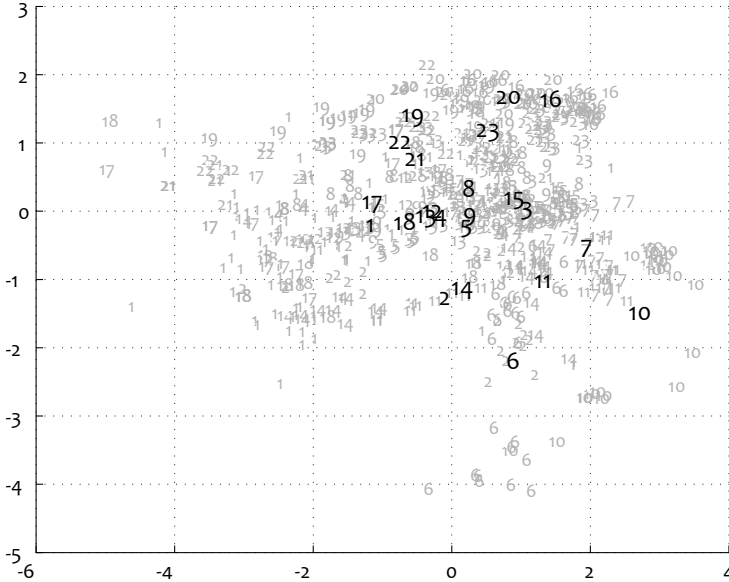


Fig. 4. A scatterplot of the 23 classes in the space of the first two principal components of the training data

While the recognition of the missing classes has not been very accurate, the results with the Hungarian Algorithm are markedly better for 2, 3 and 5 classes. We ran the Wilcoxon signed rank test for the zero median of the pairwise differences of the two consistency indices for all numbers of missing classes. The p -value, also shown in the table, indicates that the Hungarian algorithm outperforms significantly ($\alpha = 0.5$) the greedy algorithm for the who-is-missing problem.

4 Conclusions

This study formulates the who-is-missing problem and proposes a solution, completed with a measure of its quality. The Hungarian assignment algorithm, applied on the logarithms of the posterior probabilities of the objects in the set was found to dominate the intuitive alternative, called here the Greedy algorithm.

Many extensions and variants of the who-is-missing problem are yet to be formulated, for example, recognising one or more impostors in the given set, dealing with known, larger than 1, numbers of objects from each class classes, taking into the solution possible dependencies between the classes. Last but not least, important application niches for these new pattern recognition puzzles are yet to be discovered.

References

1. Amit, Y., Trouvé, A.: POP: patchwork of parts models for object recognition. *International Journal of Computer Vision* 75, 267–282 (2007)
2. Bourgeois, F., Lassalle, J.C.: An extension of the Munkres algorithm for the assignment problem to rectangular matrices. *Communications ACM* 14(12), 802–804 (1971)
3. Dearden, A., Demiris, Y., Grau, O.: Tracking football player movement from a single moving camera using particle filters. In: *Proceedings of CVMP-2006*, pp. 29–37. IET Press (2006)
4. Dietterich, T.G., Lathrop, R.H., Lozano-Perez, T.: Solving the multiple-instance problem with axis-parallel rectangles. *Artificial Intelligence* 89, 31–71 (1997)
5. Frank, A., Asuncion, A.: UCI machine learning repository (2010), <http://archive.ics.uci.edu/ml>
6. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences* 55(1), 119–139 (1997)
7. Fu, Z., Robles-Kelly, A., Zhou, J.: MILIS: multiple instance learning with instance selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33(5), 958–977 (2011)
8. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: An update. *SIGKDD Explorations* 11 (2009)
9. Kamisugi, Y., Furuya, N., Iijima, K., Fukui, K.: Computer-aided automatic identification of rice chromosomes by image parameters 1(3), 189–196 (1993)
10. Kaucic, R., Perera, A.G.A.: J., G.B., Kaufhold, Hoogs, A.: A unified framework for tracking through occlusions and across sensor gaps. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR*, vol. 1, pp. 1063–1069 (2005)
11. Kuncheva, L.I.: Full-class set classification using the Hungarian algorithm. *International Journal of Machine Learning and Cybernetics* 1(1), 53–61 (2010)
12. Kuncheva, L.: A stability index for feature selection. In: *Proc. IASTED, Artificial Intelligence and Applications*, Innsbruck, Austria, pp. 390–395 (2007)
13. Mangasarian, O.L., Wild, E.W.: Multiple instance classification via successive linear programming. *Journal of Optimization Theory and Applications* 137, 555–568 (2008)
14. McDowell, L.K., Gupta, K.M., Aha, D.W.: Cautious inference in collective classification. In: *Proceedings of AAAI*, pp. 596–601 (2007)
15. Ming, D., Tian, J.: Automatic pattern extraction and classification for chromosome images. *Journal of Infrared Milli Terahz Waves* 31, 866–877 (2010)
16. Ning, X., Karypis, G.: The set classification problem and solution methods. In: *Proceedings of SIAM Data Mining*, pp. 847–858 (2009)
17. Rodríguez, J.J., Kuncheva, L.I., Alonso, C.J.: Rotation forest: A new classifier ensemble method. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28(10), 1619–1630 (2006)
18. Sen, P., Namata, G.M., Bilgic, M., Getoor, L., Gallagher, B., Eliassi-Rad, T.: Collective classification in network data. *AI Magazine* 29, 93–106 (2008)
19. Wang, J., Zucker, J.D.: Solving the multiple-instance problem: A lazy learning approach. In: *Proceedings 17th International Conference on Machine Learning*, pp. 1119–1125 (2000)