Int. J. Mach. Learn. & Cyber. (2010) 1:53–61
DOI 10.1007/s13042-010-0002-z

ORIGINAL ARTICLE

Full-class set classification using the Hungarian algorithm

Ludmila I. Kuncheva

Received: 23 March 2010 / Accepted: 23 June 2010 / Published online: 16 September 2010

Abstract Consider a set-classification task where c objects must be labelled simultaneously in c classes, knowing that there is only one object coming from each class (full-class set). Such problems may occur in automatic attendance registration systems, simultaneous tracking of fast moving objects and more. A Bayes-optimal solution to the full-class set classification problem is proposed using a single classifier and the Hungarian assignment algorithm. The advantage of set classification over individually based classification is demonstrated both theoretically and experimentally, using simulated, benchmark and real data.

Keywords Full-class set classification · Bayes-optimal classifier · Label assignment problem

1 Introduction

For many years now, pattern recognition and machine learning have devoted major efforts to improving classification accuracy, and have allegedly cast aside a number of challenges arising from real-life problems [7]. One of the standard assumptions in classical pattern recognition is that the data comes as an independent identically distributed (i.i.d) sequence of instances. Here we abandon this assumption and consider *dependent* data where a set of instances has to be classified together, knowing that the set contains at most one instance from each class (or exactly one instance from each class, if the cardinality of the set equals the number of classes). Consider an automatic system that uses face recognition to record students' attendance of a lesson against a predefined list of identities. Without enforcing the one-to-one correspondence, one identity may be assigned to two or more students. If the individual face classifier is reasonably correct, then some mistakes can be remedied. In this context, a classifier is informally described as "reasonably correct" if the true class is ranked high among all classes, even when the top ranked class is incorrect.

An example of non-i.i.d classification, called the *multiple-instance problem*, arises in complex machine learning applications where the information about the instances is incomplete or ambiguous [4, 11, 17], 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.

Set classification is considered by Ning and Karypis [13], where all the instances in the set to be classified are known to have come from the same class. This problem may arise in face recognition where multiple images of the same person's face are submitted as a set. *Collective recognition* is another scenario where a set of instances are labelled together [12, 16]. The crucial assumption there 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 [16].

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 [8],

L. I. Kuncheva (✉)
School of Computer Science, Bangor University,
Bangor, Gwynedd LL57 1UT, UK
e-mail: l.i.kuncheva@bangor.ac.uk

which are matched from one image frame to the next. Although often mentioned en passant, this fits within the 'full-class' set classification 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 class labels. However, the instances within the set are not i.i.d, as the parts are spatially linked within the object, and also follow physical motion laws.

Our full-class set classification problem is, in a way, the opposite to multiple-instance classification or Ning and Karypis' set-classification. We assume that a classifier is available, and is trained following a standard training procedure. A *set* of instances needs to be classified, knowing that there is at most one instance from each class; termed the "full-class classification" problem. Intuitively, this set-up seems to be favourable compared to the classical i.i.d set-up, as it contains information about the dependency between the data to be classified. To the best of the author's knowledge, this problem has not been treated on a systematic level before. Here we view it from a Bayesian perspective. To find a guaranteed optimal solution we cast the full-class classification as a linear programming (LP) problem and apply the Hungarian assignment algorithm [9].

The rest of the paper is organised as follows. Section 2 introduces the details of the set classification problem and explains the proposed method. Experimental results are shown in Sect. 3, and conclusions are given in Sect. 4.

2 Full-class set classification

Consider a classification problem where an instance $\mathbf{x}$ may come from one of the c classes in the set $\Omega = \{\omega_1,\ldots, \omega_c\}$. Let $X = \{\mathbf{x}_1,\ldots, \mathbf{x}_c\}$ be a set containing exactly one instance from each class. A set-classifier, D_{set}, will label any set X with a permutation of the class indices.

The accuracy of D_{set} is the probability that the *whole* set is labelled correctly. D_{set} can be constructed using the output of a base individual classifier D_{ind}. We assume that D_{ind} outputs estimates of $P(\omega_i|\mathbf{x}_j)$, the posterior probability that instance $\mathbf{x}_j$ belongs to class ω_i. Then D_{set} is defined as

$$D_{\text{set}} : \mathcal{M} \rightarrow \mathcal{I}, \tag{1}$$

where $\mathcal{I}$ is the set of all permutations of the class indices, and $\mathcal{M}$ is the set of all square matrices of size $(c \times c)$ with entries $m_{(i,j)} \in [0, 1]$, such that $\sum_{j=1}^{c} m_{(i,j)} = 1$.

Let p be the probability that D_{ind} will label correctly a randomly chosen instance $\mathbf{x}$. Suppose that D_{set} takes the labels suggested by D_{ind} without any modification. Assuming that all instances are labelled independently, the accuracy of D_{set} will be $p_{\text{ind}} = p^c$. There is a possibility to improve upon this rate, if D_{set} takes into account the fact that all class labels must be present. To illustrate this, consider the example below.

2.1 An example

Let $c = 2$, $X = \{\mathbf{x}_1, \mathbf{x}_2\}$, and let the true posterior probabilities be $P(\omega_1|\mathbf{x}_1) > 0.5$ and $P(\omega_2|\mathbf{x}_2) > 0.5$. Ideally, D_{ind} will label $\mathbf{x}_1$ in class ω_1, and $\mathbf{x}_2$ in class ω_2. However, the estimates of the posterior probabilities obtained from D_{ind} may deviate from the true values, and corrupt the labelling. Denote these estimates by $P_1 = \hat{P}(\omega_1|\mathbf{x}_1)$ and $P_2 = \hat{P}(\omega_2|\mathbf{x}_2)$. Suppose that, due to the imperfection of the classifier, it labels $\mathbf{x}_2$ incorrectly, i.e. $P_2 < 0.5$. Then both points are labelled in ω_1, and, if these labels are taken forward, D_{set} will label the set incorrectly. However, if D_{set} is constructed so that it maximises the sum of the logarithms of the posterior probabilities with the constraint that there should be one label from each class, the mistake can be remedied.[1] The sum of the estimates for the two possible labellings are $\log(P_1) + \log(P_2)$ for permutation $\langle\omega_1, \omega_2\rangle$, and $\log(1 - P_1) + \log(1 - P_2)$ for permutation $\langle\omega_2, \omega_1\rangle$. Thus if

$$P_1P_2 > (1 - P_1)(1 - P_2), \text{ or equivalently, } P_1 + P_2 > 1, \tag{2}$$

then D_{set} will assign the correct labels to X. Equation 2 will hold not only when both $P_1 > 0.5$ and $P_2 > 0.5$, but also when one is <0.5 but $P_1 + P_2 > 1$. As a numerical example consider $P_1 = 0.8$, $P_2 = 0.4$. The score for labelling $\langle\omega_1, \omega_2\rangle$ is $\log(0.8) + \log(0.4) \approx -1.1$, and for $\langle\omega_2, \omega_1\rangle$, $\log(0.2) + \log(0.6) \approx -2.1$. Then D_{set} will label both objects correctly while D_{ind} will fail on $\mathbf{x}_2$.

2.2 The Bayes full-class set classifier

Let $X = \{\mathbf{x}_1,\ldots, \mathbf{x}_c\}$ be the set of c instances. The super-label that is assigned to X is taken from the set $\mathcal{C}$ of all possible sequences of c label outputs. Figure 1 shows the relationship between $\mathcal{C}$ and $\mathcal{I}$ (the set of all label permutations). It is convenient to call the elements of $\mathcal{C}$ 'super-labels'.

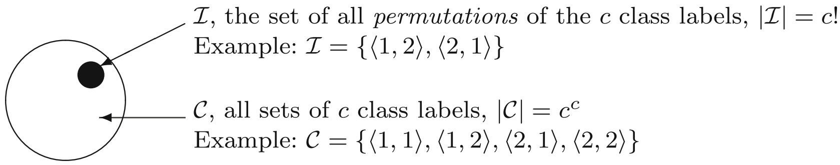
We propose to construct D_{set} as follows. Find a super-label, $S^* \in \mathcal{I}$, such that

$$S^* = \arg\max_{S} \sum_{i=1}^{c} \log \hat{P}(\omega_{s_i}|\mathbf{x}_i), \qquad S = \langle s_1, s_2, \ldots, s_c\rangle,\ S \in \mathcal{I}. \tag{3}$$

Below we demonstrate the Bayes-optimality of this criterion. Define a prior distribution over $\mathcal{C}$.

[1] The relevance of the logarithm will transpire later in relation to the Bayes optimality of the set classifier. The base of the logarithm can be any.

Fig. 1 Diagrammatic representation of the set $\mathcal{C}$ of all possible sequences of length c and set $\mathcal{I}$ of all permutations of the c class labels


$\mathcal{I}$, the set of all *permutations* of the c class labels, $|\mathcal{I}| = c!$
Example: $\mathcal{I} = \{\langle 1, 2\rangle, \langle 2, 1\rangle\}$
$\mathcal{C}$, all sets of c class labels, $|\mathcal{C}| = c^c$
Example: $\mathcal{C} = \{\langle 1, 1\rangle, \langle 1, 2\rangle, \langle 2, 1\rangle, \langle 2, 2\rangle\}$

$$P(S) = \begin{cases} \frac{1}{c!}, & \text{if } S \in \mathcal{I} \\ 0, & \text{otherwise,} \end{cases} \tag{4}$$

The posterior probability of super-label $S \in \mathcal{C}$, given the set X, is

$$P(S|X) = \frac{P(S)}{P(X)}P(X|S) = \frac{1}{c!P(X)}P(X|S). \tag{5}$$

We assume that the elements of X are conditionally independent. In other words, the super-label is fixed, and the elements of X are drawn independently from the respective classes. Then for $S = \langle s_1,\ldots, s_c\rangle$,

$$P(X|S) = \prod_{i=1}^{c} p(\mathbf{x}_i|s_i). \tag{6}$$

Fro any super-label $S \in \mathcal{I}$, the product $\prod_{i=1}^{c} P(s_i)$ is a constant because it does not depend on the order of the labels in the specific permutation. Also, $\prod_{i=1}^{c} p(\mathbf{x}_i)$ is a constant with respect to the super-label A. Define a constant $\beta = \prod_{i=1}^{c} P(s_i) / \prod_{i=1}^{c} p(\mathbf{x}_i)$. Assuming non-zero priors for all classes and non-vanishing probability density functions, multiply and divide (5) by β

$$\begin{aligned} P(S|X) &= \frac{1}{c!P(X)\beta}\prod_{i=1}^{c} p(\mathbf{x}_i|s_i)\beta \\ &= \frac{1}{c!P(X)\beta}\prod_{i=1}^{c} \frac{P(s_i)p(\mathbf{x}_i|s_i)}{p(\mathbf{x}_i)}. \end{aligned} \tag{7}$$

Noticing that the terms of the product are the posterior probabilities $P(s_i|\mathbf{x}_i)$, and absorbing the constant in front of the product into a single constant α, we obtain

$$P(S|X) = \alpha \prod_{i=1}^{c} P(s_i|\mathbf{x}_i). \tag{8}$$

Equation (8) shows the relationship between the posterior probability of the super-label and the posterior probabilities of the elements of X for the permutation of individual labels. Taking logarithm, and maximising the posterior probability on $S \in \mathcal{C}$, the optimal S^* is derived as

$$S^* = \arg\max_S \log P(S|X) = \arg\max_S \sum_{i=1}^{c} \log P(s_i|\mathbf{x}_i). \tag{9}$$

The term $\log(\alpha)$ is a constant for any S, and hence has been dropped from the criterion. Note that the maximisation of the sum of the logarithms is not equivalent to the maximisation of the sum of the posterior probability. The equivalence holds only for two classes, as shown by the example above. It is easy to construct an example for $c > 2$ classes, where the set labelling chosen by the sum of the posteriors will be different to that chosen by the sum of the logarithms.

Constructing D_{set} amounts to finding the optimal label permutation. This problem is not trivial. To look for a solution, we cast it as a LP problem. Let $M \in \mathcal{M}$ be a reward matrix with entries $m_{i,j} = \hat{P}(\omega_i|\mathbf{x}_j)$ (the probability estimate for class ω_i that D_{ind} outputs for instance $\mathbf{x}_j$). Introducing the unknowns $r_{(i,j)} \in \{0, 1\}$, $i, j = 1,\ldots, c$, the LP is

$$\max \sum_{i=1}^{c}\sum_{j=1}^{c} r_{i,j} \log(m_{i,j}),$$

subject to

$$\sum_{i=1}^{c} r_{i,j} = 1, \quad j = 1, \ldots, c,$$

$$\sum_{j=1}^{c} r_{i,j} = 1, \quad i = 1, \ldots, c.$$

Hungarian assignment algorithm [9] has been used in image processing to resolve the correspondence between tracklets [8] or shape parts [3]. We propose to use it here for constructing D_{set}. The input to the algorithm is M and the output is the optimal permutation S^*. The set classification proceeds by first labelling all instances through D_{ind} to obtain M, and then deriving the class labels through the Hungarian algorithm.

The set classifier will always be no worse than the c consecutive labellings by D_{ind}. If all instances are correctly labelled by D_{ind}, then the sum of the estimates of the posterior probabilities will maximise the criterion (3)), and D_{set} will also assign the correct labels. In addition, correct labelling may be achieved when D_{ind} is wrong, and the mistakes are remedied by the process of constructing D_{set}. Therefore

$$\begin{aligned} P_{\text{set}} &= \Pr(\text{all correct}) \\ &\quad + \Pr(\text{some incorrect})\Pr(\text{mistake corrected}) \geq p^c. \end{aligned} \tag{10}$$

2.3 A guaranteed improvement for two classes

Consider a two-class problem. The ROC curve of a classifier is constructed by nominating any of the two classes to be the ‘positive’ class and the other to be the ‘negative’ class. The area under the ROC curve, AUC, gives the probability that the classifier will rank a randomly chosen positive instance higher than randomly chosen negative instance [6]. Phrased differently, this is the probability that the classifier will make errors with less certainty compared to the certainty when assigning a correct label. Formally, using again the notation $P_1 = \hat{P}(\omega_1|\mathbf{x}_1)$ and $P_2 = \hat{P}(\omega_2|\mathbf{x}_2)$,

$$\begin{aligned}\text{AUC} &= \Pr(P_1 > 1 - P_2 \mid \omega_1 \text{ correct})\\ &= \Pr(P_2 > 1 - P_1 \mid \omega_2 \text{ correct}).\end{aligned}$$

Definition 1 We will call a two-class classifier reasonable if its AUC is strictly >0.5.

Proposition 1 *For $c = 2$ classes and a reasonable base classifier D_{ind} with accuracy p, the accuracy of D_{set} is strictly greater than p.*

Proof Without loss of generality, assume that the correct super-label is $\langle\omega_1, \omega_2\rangle$. The set-classifier D_{set} will label the set correctly in the following three mutually exclusive cases: (1) both labels are correct; (2) $\mathbf{x}_1$ is labelled correctly ($P_1 > 0.5$), $\mathbf{x}_2$ is labelled incorrectly ($P_2 < 0.5$), and $P_1 > 1 - P_2$; (3) $\mathbf{x}_1$ is labelled incorrectly ($P_1 < 0.5$), $\mathbf{x}_2$ is labelled correctly ($P_2 > 0.5$), and $P_2 > 1 - P_1$. Then the probability that D_{set} labels the set correctly is

$$\begin{aligned}P_{\text{set}} &= \Pr(\text{both correct})\\ &\quad + \Pr(\text{correct/wrong})\Pr(P_1 > 1 - P_2|\ \omega_1 \text{ correct})\\ &\quad + \Pr(\text{wrong/correct})\Pr(P_2 > 1 - P_1|\ \omega_2 \text{ correct})\\ &= p^2 + p(1-p)\text{AUC} + p(1-p)\text{AUC} \qquad (11)\end{aligned}$$

$$P_{\text{set}} = p^2 + 2p(1-p)\text{AUC}, \qquad (12)$$

Since AUC >0.5,

$$P_{\text{set}} = p^2 + 2p(1-p)\text{AUC} > p^2 + 2p(1-p) \times 0.5 = p. \qquad (13)$$

□

2.4 Simulation results

Simulations were carried out with number of classes $c = \{2, 4, 8, 10, 20, 35, 50\}$ and individual classification accuracy $p = \{0.70, 0.75, 0.80, 0.90, 0.95\}$. For each pair (c, p), 1,000 random matrices M of size $c \times c$ were sampled. To ensure that each row of M summed up to one, we selected $c-1$ random numbers to split the [0, 1] interval into c parts, and arranged them in ascending order, e.g., $n_1, n_2,..., n_{c-1}$. The posterior probabilities forming a row of M were calculated as the differences $P_i = n_i - n_{i-1}$, where $n_0 = 0$ and $i = 1,..., c$. With probability p, the entry (i, i) was swapped with the largest entry in the row, making the true label the most probable one. With probability $1 - p$, entry (i, i) would contain a random value, smaller than the highest in the row (not necessarily the second highest). Thus we are subjecting the set classifier to an unfavourable scenario, where the true class has the highest rank with probability p but is ranked randomly among the other classes with probability $1 - p$.

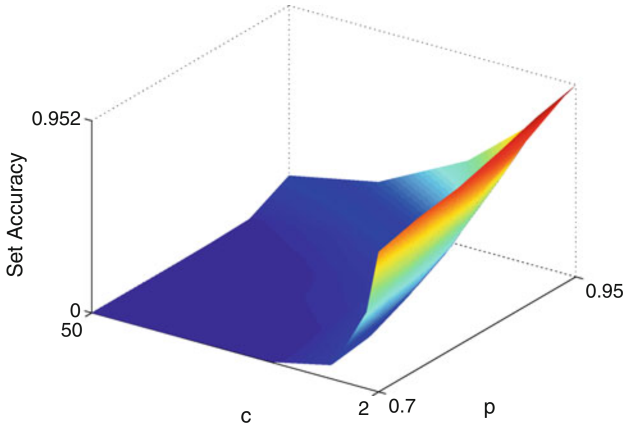


Fig. 2 Surface plot of the set accuracy p_{set} as a function of the individual accuracy p of the base classifier, and the number of classes c

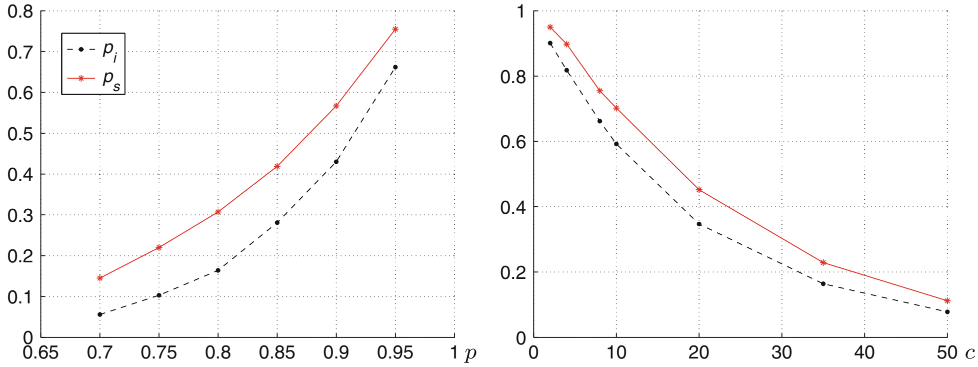
The surface of the set accuracy p_{set} as a function of p and c is shown in Fig. 2. As expected, the set accuracy for this scenario drops quite abruptly with increasing number of classes c. For more inaccurate individual classifiers D_{ind} (smaller p) the drop is steeper than for accurate classifiers. The surface also confirms that higher p leads to higher set accuracy. Recall the notation p_{ind}, the accuracy of labelling the whole set when D_{ind} is used c times. The surface for p_{ind} runs just underneath that for p_{set}. Note that even for moderate number of classes c, the set accuracy drops to zero even for quite accurate classifiers. This is a consequence of our choice of unfavourable scenario where the true class does not necessarily rate high if D_{ind} labels the object incorrectly. In real problems, it may be expected that if D_{ind} is wrong, the true class will be ranked second or third. This will increase the chance of repairing the wrong classification and will lead to a better set accuracy, compared to the simulation scenario.

To compare p_{ind} and p_{set}, Fig. 3 plots the two accuracies as functions of c and p, where one of the parameters is fixed. Both plots indicate, as expected, that $p_{\text{set}} > p_{\text{ind}}$.

3 Experiments with real data

3.1 Benchmark data

Experiments were carried out with thirty data sets from UCI [2] and from a private repository. The data sets are



(a) Fixed $c = 8$

(b) Fixed $p = 0.95$

Fig. 3 Plots of the set accuracy p_{set} and the individual-set accuracy p_{ind} as a function of the individual accuracy p of the base classifier, and the number of classes c

listed in alphabetical order in Table 1, and the source is indicated on the side. The entries in the table are the averages of the testing accuracies of a stratified tenfold cross-validation. For each fold, a linear discriminant classifier (LDC) [5] was trained on the training data. The testing data was constructed by sampling 500 sets from the testing part of the fold. Every set contained one randomly chosen instance from each class.[2] Five accuracies were estimated:

1. The individual accuracy of D_{ind} (p).
2. The accuracy of D_{ind} applied c times to label the set (p_{ind}).
3. The set-accuracy (p_{set}) of the following *Greedy* procedure. The largest posterior probability in the matrix is identified and the respective class label is assigned to the object. The object and the class label are eliminated, and the next largest posterior probability is identified and used to label the next object. The result is a set-labelling with a permutation of the class labels.
4. The set-accuracy (p_{set}) of the following *Sampled* procedure. A class label is sampled for the first object in the set using the distribution defined by the posterior probabilities for this object. The class is recorded and dropped from further consideration. For the next object, a class label is sampled from the reduction of the posterior probability distribution over the remaining set of class labels. Again, the result is a set-labelling with a permutation of the class labels.
5. The set-accuracy (p_{set}) of the proposed set-classification solution using the Hungarian algorithm (shown in the table as **H**).

The accuracy for a set was recorded as 1 if all elements of the set were correctly classified, and 0 otherwise. Superscript '●' in Table 1 indicates that **H** is significantly better, and superscript '○' indicates that **H** is significantly worse than the respective classifier (paired t test, $\alpha = 0.05$).

As expected, the set-classification approach was not even once worse than taking the c individual decisions of D_{ind}. The improvement was statistically significant on 26 out of the 30 data sets ($\alpha = 0.05$). Even better, the set accuracy happened to be significantly better than the individual accuracy p on 14 out of the 30 data sets. These are mainly two-class data sets (see Proposition 1) except for 'iris' and 'wine' data sets, both with three classes. Note that Proposition 1 is valid for any classifier, not just the LDC. The Greedy algorithm looks as a good proxy for the optimal assignment. Greedy and **H** are identical for two classes (15 data sets). For the remaining 15 datasets, **H** was better than Greedy on 8 data sets, and in 8 of the cases the difference was found to be significant at 0.05. The difference was in favour of Greedy only for the ecoli data set. The sampled algorithm was generally worse than **H**.

To gain further insight about the success of D_{set} for two classes, we plotted P_1 versus P_2 for all 15 data sets with $c = 2$ classes. For each data set and for each cross-validation fold we sampled 100 sets of 2 instances, the first from class ω_1 and the second from ω_2. Figure 4 shows a scatterplot of the 15,000 points (P_1,P_2). As argued in the proof of the Proposition, D_{set} will assign correct labels to both instances in the set if $P_1 + P_2 > 1$. The area corresponding to this case is shaded in light grey. The dark grey is where $P_1 > 0.5$ and $P_2 > 0.5$, i.e., where D_{ind} will be correct for both instances, and hence for the set. The improvement upon p^2 is due to the points that lie in the light shaded area, which, for our example, is a substantial amount: 59% of the points lie in the dark grey area and additional 27% lie in the light grey area.

There is another interesting phenomenon in Fig. 4. The density of points increases toward the right vertical edge of the unit square, where P_1 values are high regardless of the value of P_2. A detailed check revealed that this tendency was common to many of the data sets. One possible reason

[2] The Matlab code for the Hungarian algorithm was written by Alex Melin, University of Tennessee, 2006, available through Matlab Central.

Table 1 Individual accuracy (p) and set accuracies for the four set methods

Data set	N	n	c	p	p_{ind}	p_{set}		
						Greedy	Sampled	**H**
Breast	277	9	2	72.2 ●	25.9 ●	74.8 –	75.3 –	74.8
Crabs[a]	200	6	2	100.0 –	100.0 –	100.0 –	100.0 –	100.0
Ecoli	336	7	8	79.5 ○	0.0 –	31.2 ○	5.9 ○	0.0
German	1,000	24	2	77.3 ●	44.9 ●	79.9 –	77.6 ●	79.9
Glass	214	9	6	62.1 ○	0.0 ●	7.3 ●	20.1 ●	25.9
Heart	297	17	2	83.1 ●	68.5 ●	91.2 –	84.6 ●	91.2
Image	2,310	18	7	84.8 ○	29.5 ●	64.0 ●	24.6 ●	71.9
Ionosphere	351	33	2	85.8 ●	60.9 ●	89.9 –	95.1 ○	89.9
Iris	150	4	3	98.0 ●	93.7 ●	100.0 –	95.0 ●	100.0
Laryngeal1[b]	213	16	2	84.0 ●	70.3 ●	90.2 –	72.4 ●	90.2
Laryngeal2[b]	692	16	2	95.4 ●	70.8 ●	97.3 –	71.4 ●	97.3
Laryngeal3[b]	353	16	3	73.1 ○	22.7 ●	62.2 ●	59.8 ●	66.0
Letters	20,000	16	26	70.2 ○	0.0 ●	2.3 ●	0.1 ●	4.0
Liver	345	6	2	68.1 –	39.3 ●	68.7 –	47.6 ●	68.7
Pendigits	10,992	16	10	87.5 ○	25.0 ●	75.5 ●	35.7 ●	83.6
Phoneme	5,404	5	2	75.8 ●	44.0 ●	80.6 –	78.3 ●	80.6
Pima	768	8	2	77.9 ●	51.7 ●	83.3 –	76.4 ●	83.3
Satimage	6,435	36	6	84.0 ○	16.8 ●	64.4 ●	29.0 ●	75.0
Sonar	208	60	2	77.4 ●	59.6 ●	83.4 –	71.4 ●	83.4
Soybean_large	266	35	15	85.7 ○	13.4 ●	63.1 –	23.7 ●	57.5
Soybean_small	47	21	4	100.0 –	100.0 –	100.0 –	97.0 ●	100.0
Spam	4,601	57	2	88.7 ●	74.7 ●	95.3 –	89.4 ●	95.3
Spect_binary	267	22	2	82.4 –	41.9 ●	82.4 –	43.7 ●	81.5
Spect_cont	349	44	2	76.8 ●	42.4 ●	80.4 –	48.4 ●	80.4
Vehicle	846	18	4	78.6 ○	35.0 ●	70.4 ●	46.6 ●	73.1
Voice_3[b]	238	10	3	73.6 ○	2.7 ●	47.9 –	8.0 ●	51.3
Voice_9[b]	428	10	9	38.8 ○	0.0 –	0.0 –	0.0 –	0.0
Votes	232	16	2	97.0 ●	94.1 ●	98.8 –	99.2 ○	98.8
Vowel	990	11	10	62.8 ○	0.5 ●	11.8 ●	2.2 ●	25.0
Wine	178	13	3	98.3 ●	94.6 ●	100.0 –	97.2 ●	100.0

If not indicated otherwise, the source for the data set is the UCI machine learning repository [2]

N number of instances, n number of features, c number of classes

[a] Used in [15]

[b] Private collection http://www.bangor.ac.uk/mas00a/activities/real_data.htm

● Indicates that **H** is significantly better ($\alpha = 0.05$)

○ Indicates that **H** is significantly worse ($\alpha = 0.05$)

is that the class usually labelled as class ω_1 is the more "standard" of the two classes, in that it is more easily recognised. Then many of the points from this class will have high P_1. On the other hand, class ω_2 is usually the class of interest, and the objects from that class are not as clearly distinguishable from the other class, hence the smaller values of P_2.

Figure 5 shows the accuracy gain upon p_{ind} by using D_{set}. The points in the figure represent the 30 data sets. The size of the blob corresponds to the number of classes—the larger the size, the larger the c. All points are above the diagonal line indicating $p_{\text{set}} \geq p_{\text{ind}}$. The largest improvement is seen for medium values of p_{ind} and for small number of classes. The figure also shows that for larger c, both p_{ind} and p_{set} are smaller, which was also observed in the simulation experiment.

3.2 Illustration

Consider a real example where the task is to recognise six pens in an image as shown in Fig. 6. To demonstrate the benefit from set classification we used deliberately a crude

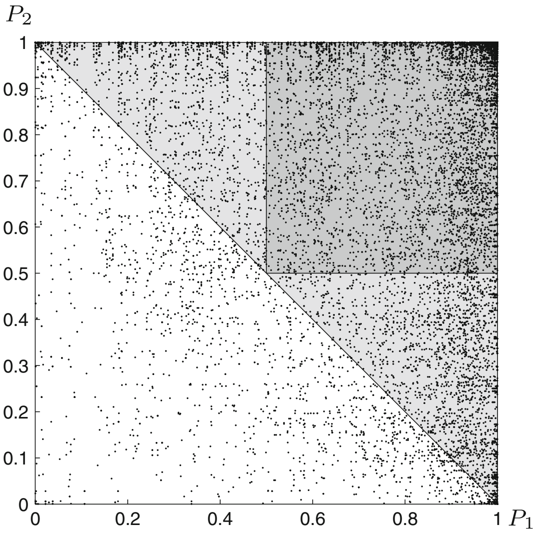


Fig. 4 Scatterplot of 15,000 points for the 15 two-class data sets. Each point corresponds to a pair of instances with true labels $\{\omega_1, \omega_2\}$. The coordinates of the point are $P_1 = \hat{P}(\omega_1|\mathbf{x}_1)$ and $P_2 = \hat{P}(\omega_2|\mathbf{x}_2)$. The *dark grey* region is where both labels are correct and the *light grey* region is where D_{set} rectifies one wrong label

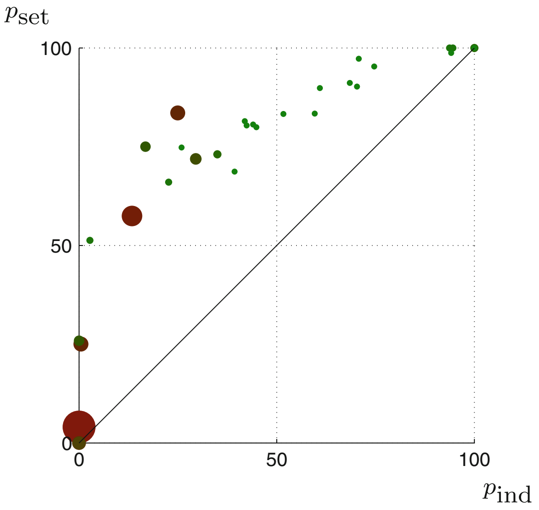


Fig. 5 Accuracy gain due to D_{set}. The points in the figure represent the 30 data sets. The size of the blob corresponds to the number of classes—the larger the size, the larger the c

segmentation method, and a primitive classifier. After segmentation, three features were extracted from the segmented rectangle: the mean read, mean green and mean blue. The nearest mean classifier was chosen for the task. A single manually cropped example of the six pens provided the "means" for the classes. Six pieces were automatically segmented from the images as shown in the figure. The segmented pieces were labelled in the six classes. The posterior probabilities for an object $\mathbf{x}$ were calculated using the soft-max equation [5]

$$\hat{P}(\omega_i|\mathbf{x}) = \frac{\exp\{g_i(\mathbf{x})\}}{\sum_{j=1}^{c} \exp\{g_j(\mathbf{x})\}},$$

where $g_i(\mathbf{x})$ is the discriminant function for class ω_i. For the nearest mean classifier used here, $g_i(\mathbf{x})$ is the negative Euclidean distance from $\mathbf{x}$ to the centre of class ω_i in the space of the three features.

The six segmented pens are shown next to the images in the order of how they were classified by D_{set}. The only incorrect classification occurred in the image in subplot (d) where the first two classes were swapped. All the other sets were labelled correctly. On the other hand, applying D_{ind} six times and taking the labels forward, none of the four sets was labelled correctly. What is more, in each of the four cases, D_{ind} labels only one of the six pens correctly while D_{set} manages to correct all five errors in cases (a), (b) and (c) and four errors in case (d).

4 Conclusions

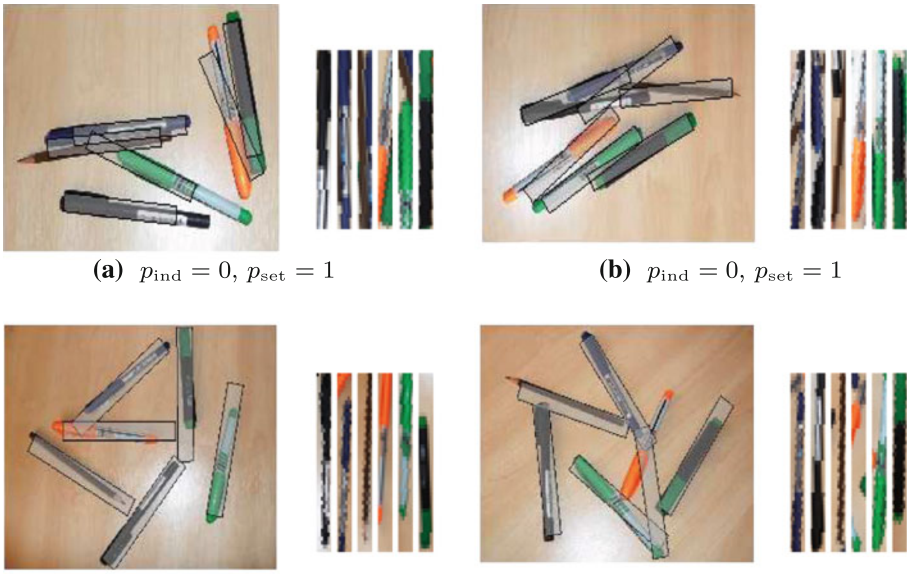
This study considers simultaneous classification of a set of c objects in c classes, where the set contains exactly one object from each class, called full-class set classification. The proposed Bayes-optimal solution is based upon the Hungarian assignment algorithm. It maximises the sum of the logarithms of the posterior probability estimates for the chosen classes under the one-to-one constraint. Simulation results and experiments with benchmark data support, from a practical perspective, the intuition and the theory about the set-classification method. In the illustration given at the end, we deliberately chose imperfect automatic segmentation and a practically untrained classifier in order to showcase the set-classification approach.

What makes this approach appealing is that the recognition is not tied up with the previous stages of the scenario. For example, no information about the previous positions of the objects that are being tracked is used in the classification. This makes the approach suitable when the objects move quickly and randomly or when the recognition times are spaced apart. An example of the latter can be wild life observation and day-to-day following of a group of animals.

The strength of the set-classification approach is when the classes are difficult to distinguish with high certainty or when there is no time for applying a sophisticated and accurate classifier. The pen-image illustration shows that high accuracy can be achieved even with a very primitive classifier and noisy input data.

One difficulty in applying the set-classification approach is that posterior probabilities are not readily produced by all classifier models. Probabilistic calibration of the outputs of the most popular and accurate classifiers such as

Fig. 6 Segmentation of six objects in a scene for subsequent classification by the nearest mean classifier



(a) $p_{\mathrm{ind}} = 0,\ p_{\mathrm{set}} = 1$ **(b)** $p_{\mathrm{ind}} = 0,\ p_{\mathrm{set}} = 1$

(c) $p_{\mathrm{ind}} = 0,\ p_{\mathrm{set}} = 1$ **(d)** $p_{\mathrm{ind}} = 0,\ p_{\mathrm{set}} = 0$

decision trees, SVM, and the nearest neighbour are not straightforward [14, 18, 19]. The classifier that lends itself naturally to this task is the Naïve Bayes classifier but it may not be very accurate when the features are dependent.

This paper only lays the ground of the full-class set classification scenario. There are several interesting further directions.

- Sensitivity to noise. A theoretical evaluation of the sensitivity of the approach to noise in the estimates of the posterior probabilities can be carried out, similar to the one in [10], but within the current context.
- Investigating the accuracy when the number of objects to classify is strictly smaller than the number of classes. This situation will arise more often than the full-class set in automatic systems for classroom attendance, where some students will be missing.
- Identifying impostors. It is interesting to investigate set-classification when some of the objects are "impostors", i.e., they do not belong to any of the classes which the base classifier is trained to recognise. The task would be to gauge the recognition success as a function of the proportion from unknown classes.
- Estimating the correct labelling within the set. Here we assumed that if at least two labels are swapped, the set is completely misclassified. For practical applications this may be too harsh, especially when there is a way to remedy a small number of mistakes to achieve correct set-classification. For example, in a face recognition system for classroom attendance, the lecturer may be able to correct swapped identities while still having the rest of the class successfully registered. Thus it will be useful to measure the correct classification within the set as well as the total set-accuracy.
- The most important, and at the same time the most challenging future direction is taking into account possible relationships within the set to be classified. For example, when a lecturer learns the names of the students in the class, a strong reference pointers are where the students sit in the lecture theatre and which students sit in a group close to one another. In labelling moving objects, the full-class set classification may be facilitated by modelling laws of physics. Tracking the parts of a composite object will be aided by incorporating knowledge of the spatial structure of the object. It will be interesting to look for a principled approach starting from the problem set-up proposed in this paper.

References

1. Amit Y, Trouvé A (2007) POP: patchwork of parts models for object recognition. Int J Comput Vis 75:267–282
2. Asuncion A, Newman DJ (2007) UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences. http://www.ics.uci.edu/~mlearn/MLRepository.html
3. Belongie S, Malik J, Puzicha J (2002) Shape matching and object recognition using shape contexts. IEEE Trans Pattern Analy Mach Intell 24:509–522
4. Dietterich TG, Lathrop RH, Lozano-Perez T (1997) Solving the multiple-instance problem with axis-parallel rectangles. Artif Intell 89:31–71
5. Duda RO, Hart PE, Stork DG (2001) Pattern classification, 2nd edn. Wiley, NY
6. Fawcett T (2003) ROC graphs: notes and practical considerations for researchers. Technical Report HPL-2003-4, HP Labs, Palo Alto. http://www.hpl.hp.com/techreports/2003/HPL-2003-4.pdf
7. Hand DJ (2006) Classifier technology and the illusion of progress (with discussion). Stat Sci 21:1–34

8. Kaucic R, Perera AGA, Brooksby G, Kaufhold J, Hoogs A (2005) 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
9. Kuhn HW (1955) The Hungarian method for the assignment problem. Nav Res Logist Q 2:83–97
10. Kuncheva LI (2002) A theoretical study on expert fusion strategies. IEEE Trans Pattern Anal Mach Intell 24(2):281–286
11. Mangasarian OL, Wild EW (2008) Multiple instance classification via successive linear programming. J Optim Theory Appl 137:555–568
12. McDowell LK, Gupta KM, Aha DW (2007) Cautious inference in collective classification. In: Processdings of AAAI, pp 596–601
13. Ning X, Karypis G (2009) The set classification problem and solution methods. In: Proceedings of SIAM data mining, pp 847–858
14. Provost F, Domingos P (2003) Tree induction for probability-based ranking. Mach Learn 52(3):199–215
15. Ripley BD (1996) Pattern recognition and neural networks. University Press, Cambridge
16. Sen P, Namata G, Bilgic M, Getoor L, Gallagher B, Eliassi-Rad T (2008) Collective classification in network data. AI Magazine 29:93–106
17. Wang J, Zucker J-Dl (2000) Solving the multiple-instance problem: a lazy learning approach. In: Proceedings 17th international conference on machine learning, pp 1119–1125
18. Zadrozny B, Elkan C (2001) Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In: Proceedings of the eighteenth international conference on machine learning (ICML'01), pp 609–616
19. Zadrozny B, Elkan C (2002) Transforming classifier scores into accurate multiclass probability estimates. In: Proceedings of the 8th international conference on knowledge discovery and data mining (KDD'02)