Analyzing Data Through Data Fusion Using Classification Techniques

Elizabeth Shanthi and D. Sangeetha

Abstract Knowledge is the ultimate output of decisions on a dataset. Applying classification rules is one of the vital methods to extract knowledge from dataset. Knowledge in a very distributed approach is derived by combining or fusing these rules. In a very standard approach this may generally be done either by combining the classifiers outputs or by combining the sets of classification rules. In this paper, we tend to do a new approach of fusing classifiers at the extent of parameters using classification rules. This approach relies on the fused probabilistic generative classifiers using multinomial distributions for categorical input dimensions and multivariable normal distributions for the continual ones. These distributions are used to produce results like valid/invalid data, error rate etc. Fusing two (or more) classifiers may be done by multiplying the hyper-distributions of the parameters. The main advantage of this fusion approach is that it requires less time to classify the data and is easily extensible for large dataset.

Keywords Data fusion · Classification · Multinomial distribution · Hyper-distribution

1 Introduction

In most of the data mining applications, the task of extracting knowledge (e.g., classification rules) from sample data is divided into a number of subtasks. Typical examples are smart sensor networks, robot teams or software agents that learn locally in their environment. At some point, there is the necessity to fuse or to

E. Shanthi (✉) · D. Sangeetha
Department of Computer Science, Avinashilingam Deemed University,
Coimbatore 641043, India
e-mail: elizabeth_cs@avinuty.ac.in

D. Sangeetha
e-mail: sangeethaprabha@ymail.com

L.C. Jain et al. (eds.), *Computational Intelligence in Data Mining - Volume 2*,
Smart Innovation, Systems and Technologies 32, DOI 10.1007/978-81-322-2208-8_16

combine the knowledge that is now *contained* in a number of *classifiers* in order to apply it to new data. *Probabilistic classifiers* provide outputs that can be interpreted as conditional probabilities as they model the conditional distribution of classes given an input sample. *Generative classifiers* aim at modeling the processes from which the sample data are assumed to originate. *Probabilistic generative classifiers* are usually based on Bayes' theorem [1] given by

$$P(c|\mathrm{x}) = \frac{p(\mathrm{x}|c) \cdot p(c)}{P(\mathrm{x})}. \tag{1}$$

where x is a (multivariate) random variable which models the input space of the classifier (e.g., $\mathrm{x} \in \mathrm{IR}^D$ where $D \in \mathrm{IN}$ is the input space dimension) and c is a random variable representing a class (e.g., $c \in \{1 \ldots C\}$). In contrast to generative classifiers, *discriminative classifiers* such as support vector machines, for instance, are *only* expected to have an optimal classification performance on new data (generalization). Compared to the probabilistic generative classifiers the discriminative classifiers take several advantages as well as drawbacks [2, 3]. There are also many application areas where both approaches can successfully be used in combination [4, 5]. *Advantages* are, the class posterior probabilities $p(c|\mathrm{x})$ are very useful to weigh single decisions when several classifiers are combined, e.g., in form of ensembles. A rejection criterion could easily be defined which allows to refuse a decision if none of the class posteriors reaches a pre-specified threshold. Possible *drawbacks* are that, these classifiers are more likely to over-fit to sample data as the (effective) number of parameters is typically quite high. The classification performance is sometimes worse if the data do not (at least nearly) meet the distribution assumptions. Altogether, it depends on the type of application whether such classifiers can successfully be applied [6].

2 Related Work

Knowledge fusion means the knowledge represented by components of classifiers fused at a parameter level. Fusion can take place at various levels or categories viz, data (e.g., sensor measurements or observations) or information extracted from databases can be fused to come to more certain conclusions. Models or parts of models trained from sample data or information can be fused if the models were constructed in a distributed fashion. The outputs of models can be fused to get more certain decisions or as in the case of temporal and spatial data mining to derive conclusions for certain points in space and time.

Here, two main fields can be identified: On one hand, knowledge is often equated with constraints and there is some work focusing on fusion of constraints discussed in [7–9]. On the other hand, knowledge is often represented by graphical models that are subject to fusion, e.g., Bayesian networks, (intelligent) topic maps, or the like as indicated in [10–13]. Paper [14] describes a Bayesian fusion approach

based on hyper-parameters and it also exploits the concept of conjugate priors. Parallelization approaches can be found in [15, 16] for instance. It could even be shown that exact approaches are feasible in the sense that they give the same results as if the data were not processed in distributed chunks [17]. These techniques typically assume some shared resources and allow for an exchange of intermediate results with the corresponding communication overhead.

3 Methodology

This paper describes the way of fusing data using one or more classifier and/or combination steps. A classification is a task that begins with a given dataset to do probabilistic generative classifier (CMM), generating classification rule, knowledge fusion and classification, probabilistic classification, fusion techniques for classification, a similarity measure for hyper-distributions and fusion training and analysis.

3.1 Probabilistic Generative Classifier and Generating Classification Rule

A CMM classifier consists of several components each of which represents the knowledge of the classifier about one process 'generating' data in the input space. Here a new fusion classifier at the level of parameters of classification rule generates rules namely RULE 1 and RULE 2 for grouping of real values in that dataset to find a positioner value. In general, a classifier is a function mapping an input value x to an output class c ϵ {1, C} of C possible classes. A probabilistic classifier takes the form p(c|x) which denotes the probability for class c given an input sample x.

According to [17],

$$P(c|X) = \frac{P(X|c)\,p(c)}{p(X)} = \frac{p(c)\sum_{i=1}^{Jc} p(j|c)\,p(x|c,j)}{p(X)}. \tag{2}$$

The classifier is split into C parts one for each class. Here p(c) is a multinomial distribution specifying the prior probability of class c, the conditional densities p(x‖c, j) are called components of the classifier and p(j‖c) is another multinomial distribution whose parameters π c, j are called mixture coefficients and weight the components in their respective part of the mixture model. The overall classifier, which is called a classifier based on a mixture model (CMM), consists of $J = \sum_{c=1}^{C} J_c$ components each of which is described by a (usually multivariate) distribution P(x|c, j). As the input data can have both, categorical and continuous dimensions, the distributions p(x|c, j) must be chosen in a way such that both cases can be handled by the classifier.

3.2 Knowledge Fusion and Classification

The fusion mechanism uses the hyper-distributions obtained in the variance inference training process. Doing so, these hyper-distributions are retained throughout the fusion process which has several advantages over a simple linear combination of CMM parameters. The classifier resulting from all fusion and combination steps is also called overall classifier. The following algorithm from [1] gives an overview of the classification and describes explicitly how two CMM classifiers can be merged (fused/combined) but, as all proposed operators are associative, multiple CMM classifiers can easily be merged by iteratively merging pairs of classifiers until only one overall classifier remains.

Algorithm1: Fusion and Combination of CMM

```
    Input: Two sets of hyper- distributions C1 and C2
    Output: Fused/combined overall classifier
1 C'<- Ø
    2 foreach c1 in C1 do
     3    found<- false
     4    foreach c2 in C2 do
     5          // similarity evalution
     6          dist<-ΔH (c1, c2)
     7          // check for consistent classes
     8          If dist<  H and class (c1) == class (c2) then
     9                // Fusion
    10        C'.add(fuse (c1,c2))
    11                C2.remove (c2)
    12                 found <- true
    13                 break
    14          if not found then
    15                // combination
    16                C'.add (c1)
    17    foreach c2 in C2 do
    18                // combination
    19              C'.add (c2)
    20   classifier < - Ø
    21    return classifier
```

3.3 Probabilistic Classification

The main work here is to divide the classifier into four divisions based on the probability of each classification. Probabilistic classifier provides output that are interrupt to an conditional probabilistic that is we are going to classify data based on the input and output of the data. This is done with different folds of probability namely (1, 1), (1, 2), (1, 3), (1, 4). Based on the likelihood and the positioner value the classifier gets fusion based on the formula.

3.4 Fusion Techniques for Classification

If two classifiers model similar processes they are likely to contain many similar components. We now want to detect such a situation in order to fuse all pairs of similar components. When two CMM are trained separately, each with a distinct part of the training data, we have two likelihood functions derived from the two sets of training data and two prior distributions. We now assume that the two priors are equal because in both cases we make use of the same prior knowledge or want to express the same amount of uncertainty about the parameters we want to estimate. Nevertheless, this leaves us with two posterior distributions.

$$\text{Posterior1} \propto \text{likelihood1.prior} \quad \text{and} \quad \text{Posterior2} \propto \text{likelihood2.prior}. \tag{3}$$

Each likelihood is itself a product over all data points in the respective training set. To fuse the posteriors they simply could be multiplied to obtain one overall posterior. This would lead to

$$\frac{\text{likelihood}_1.\text{Prior.}}{\text{posterior}_1} \frac{\text{likelihood}_2.\text{Prior}}{\text{posterior}_2} = (\text{likelihood}_1.\text{Likelihoodd}_2).\text{prior}^2. \tag{4}$$

If we had used Eq. (4) for the overall dataset with the same prior, the result would have been

$$\text{Likelihood.Prior} = (\text{likelihood}_1.\text{Likelihoodd}_2).\text{Prior}. \tag{5}$$

Comparing (4) and (5) we see that there is an additional 'prior' factor. As the prior is known, we can compensate for this fact by dividing by this prior which finally leads to our new fusion approach.

$$\text{Posterior} \propto = \frac{\dfrac{\text{likelihood}_1.\text{Prior.likelihood}_2.\text{Prior}}{\text{prior}}}{\dfrac{\text{Posterior}_1.\text{Posterior}_2}{\text{Prior}}} \tag{6}$$

We cannot simply cancel out the prior here because it is only implicitly contained in the posterior distributions which are the result of the training algorithm. Instead, we multiply the two posteriors and then divide the result by the prior. Finally, we can make the assumption that the resulting overall posterior has the same functional form as the two posteriors that are fused. This is important for two reasons: First, this allows us to easily determine the parameters of the overall posterior and second, we can derive the parameters of the CMM classifier from the fused posterior. This fusion technique is implemented for both classification1 and classification2.

3.5 A Similarity Measure for Hyper-Distributions

The similarity measure should be symmetric such that the order in which two components are compared does not matter. A simple measure ΔH that fulfills this restriction can be derived from the Hellinger distance [18]. The similarity measure ΔH directly operates on the normal and multinomial distributions of the classifier. Theoretically, it would also be possible to compute the Hellinger distance of the hyper-distributions to evaluate the similarity of components but in that case the integral in Equation could not be computed in a closed form.

3.6 Fusion Training and Analysis

In this module we fuse two classifiers based on the likelihood and positioner value. The entire data is viewed and from the fusion value generated an error rate for each classifiers with the generated value gets formulated. The conjugate prior distribution that must be used to estimate the parameters of a multinomial distribution is a Dirichlet distribution. In order to fuse two Dirichlet distributions their density functions are multiplied and then divide the result by the prior. The knowledge that we have a certain distribution type implicitly gives us a suitable normalizing factor for the fused distribution.

4 Experimental Results

4.1 Dataset Used

This data was extracted from the census bureau database found at http://www.census.gov/ftp/pub/DES/www/welcome.html. The above mentioned data was pre-processed in WEKA. Convert the dataset into .arff or .csv format and extract into WEKA. To find a classification technique such as Navie Bayes to do the followings (Correctly classified instances, Incorrectly classified instances, Kappa statistic, Mean absolute error, Root mean squared error, Relative absolute error, Coverage of cases, Mean rel. region size, Total number of instances). To get detailed accuracy by class (TP Rate, FP Rate, Precision, Recall, F-Measure, ROC Area, class) and it formed a confusion matrix. This is depicted in Tables 1, 2 and 3.

Table 1 Evaluation on training set

Correctly classified instances	1,300	81.607 %
Incorrectly classified instances	293	18.393 %
Kappa statistic	0.5299	
Mean absolute error	0.2102	
Root mean squared error	0.3665	
Relative absolute error	56.8115 %	
Root relative squared error	85.2391 %	
Coverage of cases (0.95 level)	96.108 %	
Mean rel. region size (0.95 level)	71.0923 %	
Total number of instances	1,593	

Table 2 Detailed accuracy by class

	TP rate	FP rate	Precision	Recall	F-measure	ROC area	Class
Weighted avg.	0.85	0.29	0.901	0.85	0.875	0.881	≤50K
	0.71	0.15	0.606	0.71	0.654	0.881	>50K
	0.816	0.255	0.828	0.816	0.821	0.881	

Table 3 Confusion matrix

a	b	<– Classified as
1,023	180	a = ≤50K
113	277	b = >50K

4.2 Results Observed

The basic idea of this work is how two or more classifier and thus, the represented knowledge can be combined by means of several fusion and/or combination steps.

Step 1 *Dataset extraction is done before we start our process.*

Step 2 *Generating Classification Rule*
Here Probabilistic Generative Classifier is used to generate classification rules.

Step 3 *Probabilistic Generative Classifier* (*CMM*)
For rule1 and rule2 the continuous dimensions are modeled with multivariate Gaussian distribution and results noted.

Step 4 *Knowledge fusion and classification*
Classification1 and classification2 are carried out for different dimensions and data fusion occurs.

Step 5 *Probabilistic generative classifier*
The Mahalanobis distance [1] and mixture coefficient value [1] is calculated for classification.

Step 6 *Probabilistic classification1 and classification2*
The classifier here is divided into various folds based on the probability of classification. Data gets fused based on the likelihood and positioner value.
Step 7 *Fusion techniques for classification1 and classification2*
All pairs of similar components are fused for classification1. Similar Fusion technique for classification2 may also be obtained.
Step 8 *A Similarity Measure for Hyper-Distributions*

Here two distributions namely Dirichlet distribution is applied to find parameters namely Hellinger distance, continuous dimension and hyper Distribution, whereas Normal-Wishart distribution is used for detecting error rate. The experiments have shown that this new way of fusing and combining CMM classifiers can successfully be applied in given datasets. The number of components and the classification performance of the overall classifiers obtained with the fusion/combination algorithm depend on the similarity threshold that has to be adjusted by the user depending on the application. It is influenced by parameters such as the type of the dimensions (categorical/continuous), the number of dimensions, or the number of categories in the case of categorical dimensions.

5 Conclusion and Outlook

This work discusses a new technique to fuse two probabilistic generative classifiers (CMM) into one. To identify components of two classifiers that shall be fused, a similarity measure that operates on the distributions of the classifier is suggested. The actual fusion of two components works one level higher on the hyper-distributions which are the result of the Bayesian training of a CMM. Formulas to fuse both Dirichlet and normal Wishart distributions which are the conjugate prior distributions of the multinomial and normal distributions of a CMM are used to obtain a more certain decision of a dataset. Applying data fusion approach to more than two CMM classifiers is straight forward as it is possible to apply the technique iteratively. It will certainly be possible to use the same parameter values (fusion threshold) for all single fusions. While being trivial from a technical point of view, the actual advantages for real applications have still to be pointed out in our future work. If the number of classifiers is known in advance it would also be possible to modify the fusion formulas accordingly. We can also generalize the approach to other distributions, in particular members of the exponential family of distributions and investigate how different prior distributions can be handled. We can find a more intuitive way to parameterize the fusion threshold and we will investigate the weighting of categorical and continuous dimensions in more detail. The proposed techniques could be used in the field of distributed data mining, where datasets have to be split to cope with huge amounts of data and where the communication costs have to be low. It is also possible to use fusion in distributed environments where

data are locally processed as they arise (e.g., in smart sensor networks). The work can be applied for a specific application like collaborative learning and intrusion detection.

References

1. Fisch, D., Kalkowski, E., Sick, D.: Knowledge fusion for probabilistic generative classifier with data mining application. IEEE Trans. Knowl. Data Eng. **26**, 652–666 (2014)
2. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006)
3. Fisch, D., Kühbeck, B., Sick, B., Ovaska, S.J.: So near and yet so far: new insight into properties of some well-known classifier paradigms. Inf. Sci. **180**(18), 3381–3401 (2010)
4. Bouguila, N.: Hybrid generative/discriminative approaches for proportional data modeling and classification. IEEE Trans. Knowl. Data Eng. (2011). Accepted for publication doi:10.1109/TKDE.2011.162
5. Hospedales, T.M., Gong, S., Xiang, T.: Finding rare classes: active learning with generative and discriminative models. IEEE Trans. Knowl. Data Eng. (2011). Accepted for publication. doi:10.1109/TKDE.2011.231
6. Fisch, D., Gruber, T., Sick, B.: Swiftrule: mining comprehensible classification rules for time series analysis. IEEE Trans. Knowl. Data Eng. **23**(5), 774–787 (2011)
7. Gray, P., Preece, A., Fiddian, N., Gray, W., Capon, T.B., Have, M., Azarmi, N., Wiegand, I., Ashwell, M., Beer, M. et al.: KRAFT: knowledge fusion from distributed databases and knowledge bases. In: Proceedings of the 8th International Workshop on Database and Expert Systems Applications, pp. 682–691 (1997)
8. Hui, K.Y., Gray, P.: Constraint and data fusion in a distributed information system. In: Embury S., Fiddian N., Gray W., Jones A. (eds.) Advances in Databases, Ser. Lecture Notes in Computer Science, vol. 1405, pp. 181–182. Springer, Berlin
9. Hui, K.Y.: Knowledge fusion and constraint solving in a distributed environment. Ph.D. Dissertation, Department of Computing Science, University of Aberdeen (2000)
10. Pavlin, G., De Oude, P., Maris, M., Nunnink, J., Hood, T.: A multi agent systems approach to distributed Bayesian information fusion. Inf. Fusion **11**(3), 267–282 (2010)
11. Santos Jr., E., Wilkinson, J., Santos, E.: Bayesian knowledge fusion. In: Proceedings of the 22nd International FLAIRS Conference, pp. 559–564 (2009)
12. Wang, Y., Wu, B., Hu, J.: A semantic knowledge fusion method based on topic maps. In: Workshop on Intelligent Information Technology Application, pp 74–76 (2007)
13. Smirnov, A., Pashkin, M., Chilov, N., Levashova, T.: KSNET—approach to knowledge fusion from distributed sources. Comput. Inform. **22**(2), 105–142 (2003)
14. Foina, A.G., Planas, J., Badia, R.M., Ramirez-Fernandez, F.J.: P-means, a parallel clustering algorithm for a heterogeneous multi-processor environment. In: Proceedings of the international conference on high performance computing and simulation (HPCS), pp. 239–248 (2011)
15. Li, Y., Zhao, K., Chu, X., Liu, J.: Speeding up k-means algorithm by GPUs. In: Proceedings of the 10th IEEE International Conference on Computer and Information Technology, pp. 115–122 (2010)
16. Chu, C.T., Kim, S.K., Lin, Y.A., Yu, Y., Bradski, G., Ng, A.Y., Olukotun, K.: Map-reduce for machine learning on multicore. In: Proceedings of NIPS (2006)
17. Fisch, D., Ovaska, S.J., Kalkowski, E., Sick, B.: In your interest objective interestingness measures for a generative classifier. In: Proceedings of the 3rd International Conference on Agents and Artificial Intelligence, pp. 414–423 (2011)
18. Le Cam, L., Yang, G.: Asymptotics in statistics: some basic concepts, 2nd edn. Springer, Berlin (2000)