Evidential Logistic Regression for Binary SVM Classifier Calibration

Philippe Xu[1], Franck Davoine[1,2], and Thierry Denœux[1]

[1] UMR CNRS 7253, Heudiasyc, Université de Technologie de Compiègne, France
{philippe.xu,franck.davoine,thierry.denoeux}@hds.utc.fr
https://www.hds.utc.fr/~xuphilip
[2] CNRS, LIAMA, Beijing, P. R. China

Abstract. The theory of belief functions has been successfully used in many classification tasks. It is especially useful when combining multiple classifiers and when dealing with high uncertainty. Many classification approaches such as k-nearest neighbors, neural network or decision trees have been formulated with belief functions. In this paper, we propose an evidential calibration method that transforms the output of a classifier into a belief function. The calibration, which is based on logistic regression, is computed from a likelihood-based belief function. The uncertainty of the calibration step depends on the number of training samples and is encoded within a belief function. We apply our method to the calibration and combination of several SVM classifiers trained with different amounts of data.

Keywords: Classifier calibration, theory of belief functions, Dempster-Shafer theory, support vector machines, logistic regression.

1 Introduction

The combination of pattern classifiers is an important issue in machine learning. In many practical situations, different kinds of classifiers have to be combined. If the outputs of the classifiers are of the same nature, such as probability measures or belief functions, they can be combined directly. Evidential versions of several classification methods such as the k-nearest neighbor rule [2], neural network [3] or decision trees [11] can be found in the literature. Otherwise, if their outputs are of different type, they have to be made comparable.

The transformation of the score returned by a classifier into a posterior class probability is called calibration. Several methods can be found in the literature [8,13,14]. The quality of the calibration highly depends on the amount of training data available. The use of belief functions is often more appropriate when dealing with few training data. It becomes especially critical when the classifiers to combine are trained with different amounts of training data. In this paper, we introduce an evidential calibration method that transforms the outputs of a binary classifier into belief functions. It is then applied to the calibration of SVM classifiers.

F. Cuzzolin (Ed.): BELIEF 2014, LNAI 8764, pp. 49–57, 2014.

The rest of this paper is organized as follows. In Section 2, we present likelihood-based belief functions for both statistical inference and forecasting. In particular, the case of a Bernoulli distribution is detailed. Its application to a logistic regression based calibration method is then introduced in Section 3. Experimental results on the calibration and combination of SVM classifiers are then presented in Section 4.

2 Likelihood-Based Belief Function

In this section, we present the formulation of likelihood-based belief functions. Our presentation follows the work of Denœux [4] for statistical inference and the work of Kanjanatarakul et al. for its application to forecasting [6].

2.1 Statistical Inference

Let $X \in \mathbb{X}$ be some observable data and $\theta \in \Theta$ the unknown parameter of the density function $f_\theta(x)$ generating the data. Information about θ can be inferred given the outcome x of a random experiment. Shafer [10] proposed to build a belief function Bel_x^Θ on Θ from the likelihood function. Denœux further justified this approach in [4]. After observing $X = x$, the likelihood function $L_x : \theta \mapsto f_\theta(x)$ is normalized to yield the following contour function:

$$pl_x^\Theta(\theta) = \frac{L_x(\theta)}{\sup_{\theta' \in \Theta} L_x(\theta')}, \quad \forall \theta \in \Theta, \tag{1}$$

where sup denotes the supremum operator. The consonant plausibility function associated to this contour function is

$$Pl_x^\Theta(A) = \sup_{\theta \in A} pl_x^\Theta(\theta), \quad \forall A \subseteq \Omega. \tag{2}$$

The focal sets of Bel_x^Θ are defined as

$$\Gamma_x(\gamma) = \{\theta \in \Theta \mid pl_x^\Theta(\theta) \geq \gamma\}, \quad \forall \gamma \in [0,1]. \tag{3}$$

The random set formalism can be used to represent the belief and plausibility functions on Θ. Given the Lebesgue measure λ on $[0,1]$ and the multi-valued mapping $\Gamma_x : [0,1] \to 2^\Theta$, we have

$$\begin{array}{r} Bel_x^\Theta(A) = \lambda\left(\{\gamma \in [0,1] \mid \Gamma_x(\gamma) \subseteq A\}\right) \\ Pl_x^\Theta(A) = \lambda\left(\{\gamma \in [0,1] \mid \Gamma_x(\gamma) \cap A \neq \emptyset\}\right) \end{array}, \quad \forall A \subseteq \Theta. \tag{4}$$

2.2 Forecasting

Suppose that we now have some knowledge about θ after observing some training data x. The *forecasting* problem consists in making some predictions about some random quantity $Y \in \mathbb{Y}$ whose conditional distribution $g_{x,\theta}(y)$ given $X = x$

depends on θ. A belief function on $\mathbb{Y}$ can be derived from the sampling model proposed by Dempster [1]. For some unobserved auxiliary variable $Z \in \mathbb{Z}$ with known probability distribution μ independent of θ, we define a function φ so that

$$Y = \varphi(\theta, Z). \tag{5}$$

A multi-valued mapping $\Gamma'_x : [0,1] \times \mathbb{Z} \to 2^{\mathbb{Y}}$ is defined by composing Γ_x with φ

$$\begin{array}{rccc} \Gamma'_x : & [0,1] \times \mathbb{Z} & \to & 2^{\mathbb{Y}} \\ & (\gamma, z) & \mapsto & \varphi(\Gamma_x(\gamma), z). \end{array} \tag{6}$$

A belief function on $\mathbb{Y}$ can then be derived from the product measure $\lambda \otimes \mu$ on $[0,1] \times \mathbb{Z}$ and the multi-valued mapping Γ'_x

$$\begin{array}{r} Bel_x^{\mathbb{Y}}(A) = (\lambda \otimes \mu)\left(\{(\gamma, z) \ \mid \varphi\left(\Gamma_x\left(\gamma\right), z\right) \subseteq A\}\right) \\ Pl_x^{\mathbb{Y}}(A) = (\lambda \otimes \mu)\left(\{(\gamma, z) \ \mid \varphi\left(\Gamma_x\left(\gamma\right), z\right) \cap A \neq \emptyset\}\right) \end{array}, \quad \forall A \subseteq \Omega. \tag{7}$$

2.3 Binary Case Example

In the particular case where Y is a random variable with a Bernoulli distribution $\mathcal{B}(\omega)$, it can be generated by a function φ defined as

$$Y = \varphi(\omega, Z) = \begin{cases} 1 \text{ if } Z \leq \omega, \\ 0 \text{ otherwise,} \end{cases} \tag{8}$$

where Z has a uniform distribution on $[0,1]$. Assume that the belief function Bel_x^{Ω} on Ω is induced by a random closed interval $\Gamma_x(\gamma) = [U(\gamma), V(\gamma)]$. In particular, it is the case if it is the consonant belief function associated to a unimodal contour function. We get

$$\Gamma'_x(\gamma, z) = \varphi\left([U(\gamma), V(\gamma)], z\right) = \begin{cases} 1 & \text{if } Z \leq U(\gamma), \\ 0 & \text{if } Z > V(\gamma), \\ \{0,1\} & \text{otherwise.} \end{cases} \tag{9}$$

The *predictive* belief function $Bel_x^{\mathbb{Y}}$ can then be computed as

$$\begin{aligned} Bel_x^{\mathbb{Y}}(\{1\}) &= (\lambda \otimes \mu)(\{(\gamma, z) \mid Z \leq U(\gamma)\}) && \text{(10a)} \\ &= \int_0^1 \mu(\{z \mid z \leq U(\gamma)\}) f(\gamma) d\gamma && \text{(10b)} \\ &= \int_0^1 U(\gamma) f(\gamma) d\gamma = \mathbb{E}(U) && \text{(10c)} \end{aligned}$$

and

$$\begin{aligned} Bel_x^{\mathbb{Y}}(\{0\}) &= (\lambda \otimes \mu)(\{(\gamma, z) \mid Z > V(\gamma)\}) && \text{(11a)} \\ &= 1 - (\lambda \otimes \mu)(\{(\gamma, z) \mid Z \leq V(\gamma)\}) && \text{(11b)} \\ &= 1 - \mathbb{E}(V). && \text{(11c)} \end{aligned}$$

As U and V take only non-negative values, these quantities have the following expressions:

$$Bel_x^{\mathbb{Y}}(\{1\}) = \int_0^{+\infty} (1 - F_U(u))du = \int_0^{\hat{\omega}} (1 - pl_x^{\Omega}(u))du \tag{12a}$$

$$= \hat{\omega} - \int_0^{\hat{\omega}} pl_x^{\Omega}(u)du \tag{12b}$$

and

$$Pl_x^{\mathbb{Y}}(\{1\}) = 1 - Bel_x^{\mathbb{Y}}(\{0\}) = \int_0^{+\infty} (1 - F_V(v))dv \tag{13a}$$

$$= \hat{\omega} + \int_{\hat{\omega}}^{1} pl_x^{\Omega}(v)dv, \tag{13b}$$

where $\hat{\omega}$ is the value maximizing pl_x^{Ω}. In many practical situations, the belief function $Bel_x^{\mathbb{Y}}$ cannot be expressed analytically. However, they can be approximated either by Monte Carlo simulation using Equations (10) and (11) or by numerically estimating the integrals of Equations (12) and (13).

3 Classifier Calibration

Let us consider a binary classification problem. Let $x = \{(x_1, y_1), \dots, (x_n, y_n)\}$ be some training data, where $x_i \in \mathbb{R}$ is the score returned by a pre-trained classifier for the i-th training sample which label is $y_i \in \{0, 1\}$. Given a test sample of score $s \in \mathbb{R}$ and unknown label $y \in \{0, 1\}$, the aim of calibration is to estimate the posterior class probability $P(y = 1|s)$. Several calibration methods can be found in the literature. Binning [13], isotonic regression [14] and logistic regression [8] are the most commonly used ones. Niculescu-Mizil and Caruana [7] showed that logistic regression is well-adapted for calibrating maximum margin methods like SVM. Moreover, it is less prone to over-fitting as compared to binning and isotonic regression, especially when relatively few training data are available. Thus, logistic regression will be considered in this paper.

3.1 Logistic Regression-Based Calibration

Platt [8] proposed to use a logistic regression approach to transform the scores of an SVM classifier into posterior class probabilities. He proposed to fit a sigmoid function

$$P\left(y = 1|s\right) \approx h_s(\theta) = \frac{1}{1 + \exp\left(a + bs\right)}. \tag{14}$$

The parameter $\theta = (a, b) \in \mathbb{R}^2$ of the sigmoid function is determined by maximizing the likelihood function on the training data,

$$L_x(\theta) = \prod_{k=1}^{n} p_k^{y_k}(1 - p_k)^{1-y_k} \quad \text{with} \quad p_k = \frac{1}{1 + \exp(a + bx_k)}. \tag{15}$$

To reduce over-fitting and prevent a from becoming infinite when the training examples are perfectly separable, Platt proposed to use an out-of-sample data model by replacing y_k and $1 - y_k$ by t_+ and t_- defined as

$$t_+ = \frac{n_+ + 1}{n_+ + 2} \quad \text{and} \quad t_- = \frac{1}{n_- + 2}, \tag{16}$$

where n_+ and n_- are respectively the number of positive and negative training samples. This ensures L_x to have a unique supremum $\hat{\theta} = (\hat{a}, \hat{b})$.

3.2 Evidential Extension

After observing the score s of a test sample, its label $y \in \{0, 1\}$ can be seen a the realisation of a random variable Y with a Bernoulli distribution $\mathcal{B}(\omega)$, where $\omega = h_s(\theta) \in [0, 1]$. A belief function $Bel^{\mathbb{Y}}_{x,s}$ can thus be derived from the contour function $pl^{\Omega}_{x,s}$ as described in Section 2.3. Function $pl^{\Omega}_{x,s}$ can be computed from Pl^{Θ}_x as

$$pl^{\Omega}_{x,s}(\omega) = \begin{cases} 0 & \text{if } \omega \in \{0, 1\} \\ Pl^{\Theta}_x \left(h_s^{-1}(\omega)\right) & \text{otherwise,} \end{cases} \tag{17}$$

where

$$h_s^{-1}(\omega) = \left\{ (a, b) \in \Theta \;\middle|\; \frac{1}{1 + \exp(a + bs)} = \omega \right\} \tag{18}$$

$$= \left\{ (a, b) \in \Theta \;\middle|\; a = \ln\left(\omega^{-1} - 1\right) - bs \right\}, \tag{19}$$

which finally yields

$$pl^{\Omega}_{x,s}(\omega) = \sup_{b \in \mathbb{R}} pl^{\Theta}_x \left(\ln\left(\omega^{-1} - 1\right) - bs, b \right), \quad \forall \omega \in (0, 1). \tag{20}$$

Figure 1 illustrates the computation of the predictive belief function $Bel^{\mathbb{Y}}_{x,s}$. Fig. 1 (a) shows level sets of the contour function pl^{Θ}_x computed from the scores of an SVM classifier trained on the UCI[1] Australian dataset. The value of $pl^{\Omega}_{x,s}(\omega)$ is defined as the maximum value of pl^{Θ}_x along the line $a = \ln(\gamma^{-1} - 1) - bs$ represented by the doted lines. It can be approximated by a gradient descent algorithm. Fig. 1 (b) shows the contour function $pl^{\Omega}_{x,s}$ from which $Bel^{\mathbb{Y}}_{x,s}$ can be computed using Equations (12) and (13). Fig. 1 (c-d) displays the calibration results for $n = 20$ and $n = 200$, respectively.

4 Experimental Evaluation

Experiments were conducted using three binary classification problems from the UCI dataset: Adult, Australian and Diabetes. For each dataset, 10 non-linear

[1] http://archive.ics.uci.edu/ml

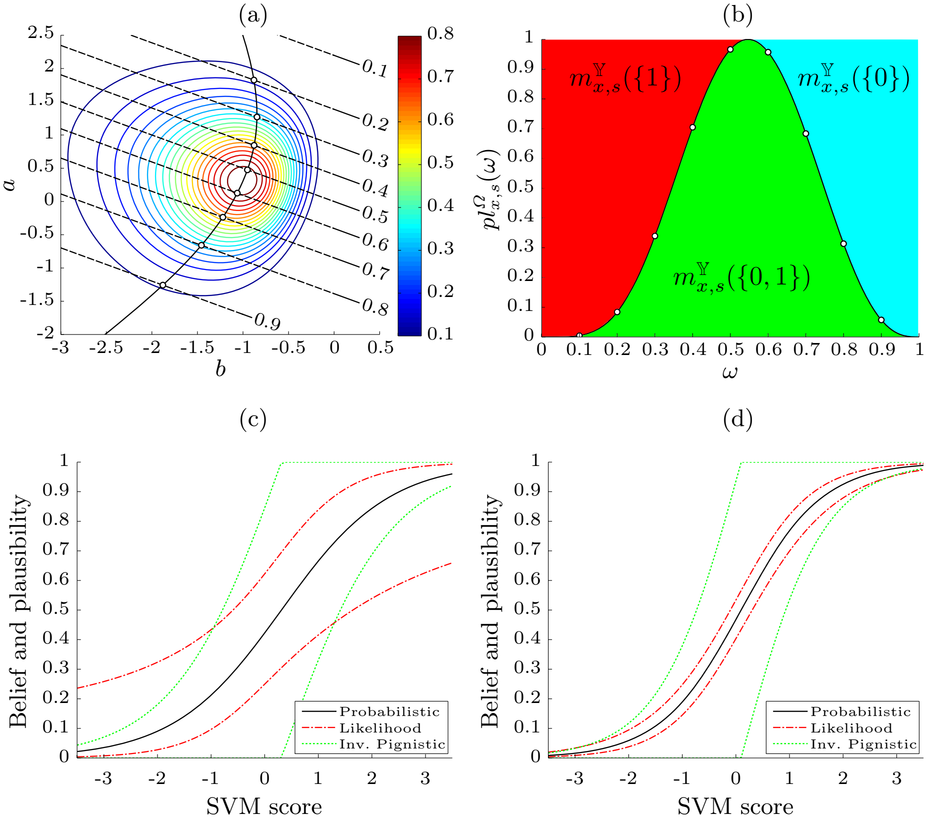

Fig. 1. Calibration results on the Australian dataset. (a) Level sets of the contour function pl_x^{Θ}. (b) Contour function $pl_{x,s}^{\Omega}$ with $s = 0.5$. The three coloured areas correspond to the predictive mass function $m_{x,s}^{\mathbb{Y}}$. (c) Calibration results with $n = 20$. (d) Calibration results with $n = 200$.

SVM classifiers with RBF kernel were trained using non-overlapping training sets of different sizes. Three scenarios were considered, as illustrated in Fig. 2. In the first scenario (a), all 10 classifiers were trained using the same amount of training data. In the second one (b), one half of the classifiers were trained with five times more data than the other half. Finally, in (c), one classifier was trained with $2/3^{\mathrm{rd}}$ of the data, a second one used $1/5^{\mathrm{th}}$ and the eight other ones shared the rest uniformly. The total amounts of training and testing data are detailed in Table 1.

The LibSVM[2] library was used to train the classifiers. For each experiment, 5-fold cross validation was conducted on the training data to get both the SVM parameters and the scores for calibration. As each classifier was trained with different training data, they were assumed to be independent. After calibration, the classifier outputs were thus combined with Dempster's rule. The class with

[2] `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

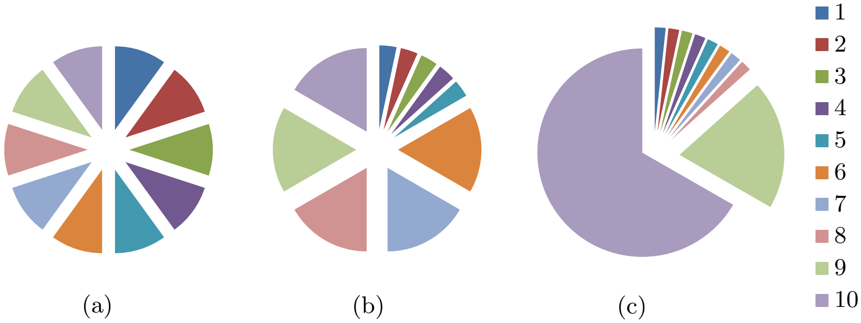

Fig. 2. Proportions of data used to train each of the 10 classifier. (a) All classifiers use 10% of the training data. (b) One half the classifiers use 1/6th of the data and the other half the rest. (c) One classifier uses 2/3rd of the data, a second one uses 1/5th and the eight other classifiers use the rest.

Table 1. Classification accuracy for several datasets and different scenarios. The best results are underlined and those that are not significantly different are in bold.

	Adult #train=600, #test=16,281			Australian #train=300, #test=390		
Scenario	(a)	(b)	(c)	(a)	(b)	(c)
Probabilistic	83.24%	82.70%	80.90%	**85.13**%	**85.90**%	85.90%
Inv. Pign.	**83.32**%	82.79%	81.02%	**85.13**%	**85.90**%	**86.41**%
Likelihood	**83.29**%	**83.03**%	**81.65**%	**85.13**%	**86.67**%	**88.46**%

	Diabetes #train=300, #test=468		
Scenario	(a)	(b)	(c)
Probabilistic	**78.42**%	**77.14**%	53.42%
Inv. Pign.	**78.63**%	**77.14**%	54.70%
Likelihood	**79.06**%	**77.35**%	**68.16**%

maximum plausibility was selected for decision. The probabilistic calibration served as baseline. We compared it the likelihood-based evidential approach and the inverse pignistic transformation. The classification accuracies on the testing data are shown in Table 1.

To compare the performances of the different calibration approaches, the significance of the results was evaluated from a McNemar test [5] at the 5% level. The best results were always obtained by the likelihood-based approach except for Adult (a). In particular, expect for the inverse pignistic transformation on the Australian dataset, the results were always significantly better for scenario (c). For the Adult dataset, the likelihood-based calibration always gave significantly better results than the probabilistic approach. We can see that the likelihood-based approach is more robust when the training sets have highly unbalanced sizes.

5 Conclusion

In this paper, we showed how to extend logistic regression-based calibration methods using belief functions. Belief functions can better represent the uncertainty of the calibration procedure, especially when very few training data are available. The method was used to calibrate the scores from SVM classifiers but it may also be used for other classification algorithms. Evidential formulations of other calibration methods such as binning [13] and isotonic regression [14] will be considered in future work. Extension to multi-class problem is also possible through the use of one-vs-one or one-vs-all binary decompositions. Comparison of probabilistic approaches [12] and evidential ones [9] will be considered in future work.

Acknowledgments. This research was carried out in the framework of the Labex MS2T, which was funded by the French Government, through the program "Investments for the future" managed by the National Agency for Research (Reference ANR-11-IDEX-0004-02). It was supported by the ANR-NSFC Sino-French PRETIV project (Reference ANR-11-IS03-0001).

References

1. Dempster, A.: The Dempster-Shafer calculus for statisticians. International Journal of Approximate Reasoning 48(2), 365–377 (2008)
2. Denœux, T.: A k-nearest neighbor classification rule based on dempster-shafer theory. IEEE Transactions on Systems, Man and Cybernetics 25(5), 804–813 (1995)
3. Denœux, T.: A neural network classifier based on dempster-shafer theory. IEEE Transactions on Systems, Man and Cybernetics 30(2), 131–150 (2000)
4. Denœux, T.: Likelihood-based belief function: justification and some extensions to low-quality data. International Journal of Approximate Reasoning (in Press, 2014), http://dx.doi.org/10.1016/j.ijar.2013.06.007, doi:10.1016/j.ijar.2013.06.007
5. Dietterich, T.G.: Approximate statistical tests for comparing supervised classification algorithms. Neural Computation 10(7), 1895–1923 (1998)
6. Kanjanatarakul, O., Sriboonchitta, S., Denœux, T.: Forecasting using belief functions: an application to marketing econometrics. International Journal of Approximate Reasoning 55(5), 1113–1128 (2014)
7. Niculescu-Mizil, A., Caruana, R.: Predicting good probabilities with supervised learning. In: Proceedings of the 22nd International Conference on Machine Learning, Bonn, Germany, pp. 625–632 (2005)
8. Platt, J.C.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: Smola, A.J., Bartlett, P., Schölkopf, B., Schuurmans, D. (eds.) Advances in Large-Margin Classifiers, pp. 61–74. MIT Press (1999)
9. Quost, B., Denœux, T., Masson, M.H.: Pairwise classifier combination using belief functions. Pattern Recognition Letters 28(5), 644–653 (2007)
10. Shafer, G.: A mathematical theory of evidence. Princeton University Press, Princeton (1976)

11. Sutton-Charani, N., Destercke, S., Denœux, T.: Classification trees based on belief functions. In: Denoeux, T., Masson, M.H. (eds.) Belief Functions: Theory and Applications. AISC, vol. 164, pp. 77–84. Springer, Heidelberg (2012)
12. Wu, T.F., Lin, C.J., Weng, R.C.: Probability estimates for multi-class classification by pairwise coupling. Journal of Machine Learning Research 5, 975–1005 (2004)
13. Zadrozny, B., Elkan, C.: Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In: Proceedings of the 18th International Conference on Machine Learning, Williamstown, USA, pp. 609–616 (2001)
14. Zadrozny, B., Elkan, C.: Transforming classifier scores into accurate multiclass probability estimates. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, USA, pp. 694–699 (2002)