Boosting in Probabilistic Neural Networks

Jiří Grim, Petr Somol, Pavel Pudil

Department of Pattern Recognition
Institute of Information Theory and Automation
Academy of Sciences of the Czech Republic

Abstrakt

The basic idea of boosting is to increase the pattern recognition accuracy by combining classifiers which have been derived from differently weighted versions of the original training data. It has been verified in practical experiments that the resulting classification performance can be improved by increasing the weights of misclassified training samples. However, in statistical pattern recognition, the weighted data may influence the form of the estimated conditional distributions and therefore the theoretically achievable classification error could increase. We prove that in case of maximum-likelihood estimation the weighting of discrete data vectors is asymptotically equivalent to multiplication of the estimated discrete conditional distributions by a positive bounded function. Consequently, the Bayesian decision-making is shown to be asymptotically invariant with respect to arbitrary weighting of data provided that (a) the weighting function is defined identically for all classes and (b) the prior probabilities are properly modified.

1 Introduction

The performance of pattern recognition methods can be improved by applying some combining technique to a set of classifiers designed for a given problem. There are different possibilities to create suitable classifier ensembles [10]. Bagging constructs the classifiers by sampling the training data set, random subspace method is based on sampling the feature set and boosting derives the classifiers from differently weighted versions of the original training set.

The most widely used boosting algorithm is AdaBoost [3]. In AdaBoost the base classifier is applied iteratively to modify the weights of data vectors in the training set. At each iteration the weights of the misclassified examples are increased to design a more successful classifier. In this way AdaBoost constructs increasingly difficult learning problems and the corresponding classifiers are combined by a weighted vote into the final decision rule. It has been verified that boosting can convert a weak classifier to a strong decision function [3, 4, 8, 10, 12]. Generally the effect of boosting increases

[0]Early version of the paper: Grim J., Pudil P., Somol P.: "Boosting in probabilistic neural networks". In: Proceedings of the 16th International Conference on Pattern Recognition. (Kasturi R., Laurendeau D., Suen C. eds.). IEEE Computer Society, Los Alamitos 2002, pp. 136-139.

with the number of combined classifiers. However, in case of good classifiers, boosting may occur unproductive or counterproductive [12].

In the present paper we consider the problem of weighting discrete training data in connection with the optimization of probabilistic neural networks. We analyze first some basic theoretical aspects of weighting data in case of maximum-likelihood estimation of discrete probability distributions. In particular we show that the weighting of discrete data vectors is asymptotically equivalent to the analogous weighting of the estimated discrete conditional distributions. In view of this fact the Bayesian decision-making is shown to be asymptotically invariant with respect to arbitrary weighting provided that (a) the weighting function is defined identically for all classes and (b) the prior probabilities are properly modified. Consequently, the Bayes classification error is also asymptotically invariant to arbitrary boosting provided that the combined classifiers satisfy the above properties.

The proposed modification of boosting has been applied to statistical recognition of unconstrained hand-written numerals from the database of Concordia University, Montreal, Canada whereby the class-conditional probability distributions have been approximated by means of multivariate Bernoulli mixtures. The maximum-likelihood estimates of the included parameters have been computed by using EM algorithm [1].

2 Probabilistic neural networks

Considering a finite set of mutually exclusive classes $\Omega = \{\omega_1, \ldots, \omega_K\}$ we assume that some multivariate binary observations

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_N) \in \mathcal{X}, \quad \mathcal{X} = \{0,1\}^N \tag{1}$$

occur with the respective class-conditional probability distributions $P(\boldsymbol{x}|\omega)p(\omega)$. We approximate the unknown distributions by finite mixtures of product components:

$$P(\boldsymbol{x}|\omega) = \sum_{m \in \mathcal{M}_\omega} F(\boldsymbol{x}|m,\omega) f(m|\omega), \quad \omega \in \Omega. \tag{2}$$

Here $f(m|\omega) \geq 0$ are the conditional probabilistic weights, $F(\boldsymbol{x}|m,\omega)$ are the product distributions

$$F(\boldsymbol{x}|m,\omega) = \prod_{n \in \mathcal{N}} f_n(x_n|m,\omega), \quad \mathcal{N} = \{1,2,\ldots,N\} \tag{3}$$

and $\mathcal{M}_\omega$ the index sets. If the probabilistic description is known then any new observation $\boldsymbol{x} \in \mathcal{X}$ can be classified by means of the Bayes formula for posterior probabilities

$$p(\omega|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|\omega)p(\omega)}{P(\boldsymbol{x})}, \quad \omega \in \Omega. \tag{4}$$

where

$$P(\boldsymbol{x}) = \sum_{\omega \in \Omega} P(\boldsymbol{x}|\omega)p(\omega). \tag{5}$$

To simplify notation we assume a consecutive indexing of components throughout the classes. In this way the component index m uniquely identifies the class $\omega \in \Omega$ and therefore the parameter ω can be partly omitted in the following sections.

The basic idea of PNN is to view the component distributions in Eq. (2) as formal neurons. As the component distributions $F(\boldsymbol{x}|m,\omega)$ must be normed on the input space $\mathcal{X}$, the corresponding neurons have to be connected with all input nodes. We avoid this undesirable complete inter-connection property by using structural mixtures [5, 7]. Introducing binary structural parameters $\phi_{mn} \in \{0,1\}$ we define

$$F(\boldsymbol{x}|m,\omega) = \prod_{n\in\mathcal{N}} f_n(x_n|m,\omega)^{\phi_{mn}} f_n(x_n|0)^{1-\phi_{mn}} \tag{6}$$

where $f_n(x_n|0)$ are some univariate background distributions usually defined as (non-zero) unconditional marginals, i.e. $f_n(x_n|0) = P_n(x_n),\ n \in \mathcal{N}$. We can see that $F(\boldsymbol{x}|m,\omega)$ is always a multivariate Bernoulli distribution. Nevertheless, by setting $\phi_{mn} = 0$, any component-specific distribution $f_n(x_n|m,\omega)$ is actually replaced by the respective fixed background distribution $f_n(x_n|0)$. Assuming binary variables x_n we can write

$$\theta_{nm} = f_n(1|m), \quad n \in \mathcal{N}, \tag{7}$$

$$f_n(x_n|m,\omega) = \theta_{nm}^{x_n}(1-\theta_{nm})^{1-x_n}, \quad x_n \in \{0,1\} \tag{8}$$

and, making substitution (6) and (8), we obtain a modified distribution mixture containing structural parameters:

$$P(\boldsymbol{x}|\omega) = \sum_{m\in\mathcal{M}_\omega} F(\boldsymbol{x}|0) G(\boldsymbol{x}|m,\phi_m) f(m|\omega). \tag{9}$$

Here

$$F(\boldsymbol{x}|0) = \prod_{n\in\mathcal{N}} f_n(x_n|0) = \prod_{n\in\mathcal{N}} \theta_{n0}^{x_n}(1-\theta_{n0})^{1-x_n} \tag{10}$$

is a nonzero "background" distribution common to all classes $\omega \in \Omega$ and the component functions $G(\boldsymbol{x}|m,\phi_m)$ may be defined on arbitrary subspaces

$$G(\boldsymbol{x}|m,\phi_m) = \prod_{n\in\mathcal{N}} \left[\frac{\theta_{nm}^{x_n}(1-\theta_{nm})^{1-x_n}}{\theta_{n0}^{x_n}(1-\theta_{n0})^{1-x_n}}\right]^{\phi_{mn}} \tag{11}$$

according to the structural binary parameters ϕ_{mn}. It can be seen that the background probability distribution $F(\boldsymbol{x}|0)$ can be canceled in the Bayes formula (4) and therefore the posterior probability $p(\omega|\boldsymbol{x})$ is proportional to weighted sum of the component functions $G(\boldsymbol{x}|m,\phi_m)$ which can be defined on different subspaces. Note that (cf. Sec. 3) the optimization of the structural parameters ϕ_{nm} can be included into the EM algorithm in full generality.

3 Boosting and finite mixtures

Assume that for each class $\omega \in \Omega$ there is a training set

$$\mathcal{S}_\omega = \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(I_\omega)}\}, \quad \boldsymbol{x}^{(i)} \in \mathcal{X} \tag{12}$$

and in all classes the data vectors are weighted by a positive bounded function $\lambda(\boldsymbol{x})$ uniquely defined on $\mathcal{X}$. Let us note that, in the framework of m.-l. estimation, arbitrary

weighting of the training data is naturally realized by weighting the corresponding terms of the likelihood function:

$$L_\omega = \frac{1}{\Lambda(\mathcal{S}_\omega)} \sum_{x \in \mathcal{S}_\omega} \lambda(\boldsymbol{x}) \log P(\boldsymbol{x}|\omega), \tag{13}$$

$$\Lambda(\mathcal{S}_\omega) = \sum_{x \in \mathcal{S}_\omega} \lambda(\boldsymbol{x}). \tag{14}$$

The weighted version of the likelihood function is equivalent to repeated occurrence of data vectors in the training set $\mathcal{S}_\omega$ and therefore we can easily derive the corresponding weighted modification of the structural EM algorithm (cf. [5, 6, 7]):

$$q(m|\,\boldsymbol{x}) = \frac{G(\boldsymbol{x}|\,m, \phi_m) f(m|\omega)}{\sum_{j \in \mathcal{M}_\omega} G(\boldsymbol{x}|j, \phi_j) f(j|\omega)}, \tag{15}$$

$$f^{'}(m|\omega) = \frac{1}{\Lambda(\mathcal{S}_\omega)} \sum_{x \in \mathcal{S}_\omega} \lambda(\boldsymbol{x}) q(m|\,\boldsymbol{x}), \tag{16}$$

$$\theta^{'}_{nm} = \frac{1}{\Lambda(\mathcal{S}_\omega) f^{'}(m|\omega)} \sum_{x \in \mathcal{S}_\omega} x_n \lambda(\boldsymbol{x}) q(m|\boldsymbol{x}), \tag{17}$$

$$\gamma^{'}_{nm} = f^{'}(m|\omega)[\theta^{'}_{nm} \log \frac{\theta^{'}_{nm}}{\theta_{n0}} + (1 - \theta^{'}_{nm}) \log \frac{(1 - \theta^{'}_{nm})}{(1 - \theta_{n0})}],$$

$$\phi^{'}_{nm} = \begin{cases} 1, & \gamma^{'}_{nm} \in \Gamma^{'}_m, \\ 0, & \gamma^{'}_{nm} \notin \Gamma^{'}_m, \end{cases} \quad , \quad m \in \mathcal{M}_\omega. \tag{18}$$

Here $\Gamma^{'}_m$ is the set of r highest quantities $\gamma^{'}_{nm}$ for a fixed $m \in \mathcal{M}$ and $f^{'}(m|\omega), \theta^{'}_{nm}$, and $\phi^{'}_{nm}$ are the new values of mixture parameters.

Obviously the weighted EM algorithm (15) - (18) converges monotonically to a possibly local maximum and retains all its basic properties. We prove now the following simple Lemma:

Lemma 1. Let $\mathcal{S}_\omega$ be a sample of independent observations identically distributed according to an unknown discrete distribution $P^*(\boldsymbol{x}|\omega)$ and $\lambda(\boldsymbol{x})$ be a positive bounded function on $\mathcal{X}$. The maximum-likelihood estimation of the unknown distribution based on the weighted likelihood function L_ω (cf. (13)) is asymptotically equivalent to m.-l. estimation of the distribution

$$\tilde{P}(\boldsymbol{x}|\omega) = \frac{\lambda(\boldsymbol{x})}{\Lambda^*_\omega} P^*(\boldsymbol{x}|\omega), \quad \boldsymbol{x} \in \mathcal{X}, \tag{19}$$

$$\Lambda^*_\omega = \sum_{x \in \mathcal{X}} \lambda(\boldsymbol{x}) P^*(\boldsymbol{x}|\omega)$$

obtained by analogous weighting of the original distribution $P^*(\boldsymbol{x}|\omega)$.

Proof. Let us note that for the sample-size $|\mathcal{S}_\omega|$ approaching infinity we can write (cf. (14))

$$\Lambda^*_\omega = \lim_{|\mathcal{S}_\omega| \to \infty} \frac{\Lambda(\mathcal{S}_\omega)}{|\mathcal{S}_\omega|} = \sum_{x \in \mathcal{X}} \lambda(\boldsymbol{x}) P^*(\boldsymbol{x}|\omega) \tag{20}$$

and further (cf.(13))

$$L^*_\omega = \lim_{|\mathcal{S}_\omega|\to\infty}\{\frac{1}{\Lambda(\mathcal{S}_\omega)}\sum_{x\in\mathcal{S}_\omega}\lambda(\boldsymbol{x})\log P(\boldsymbol{x}|\omega)\} =$$

$$= \lim_{|\mathcal{S}_\omega|\to\infty}\{\frac{|\mathcal{S}_\omega|}{\Lambda(\mathcal{S}_\omega)}\sum_{x\in\mathcal{S}_\omega}\frac{\nu(\boldsymbol{x}|\omega)}{|\mathcal{S}_\omega|}\lambda(\boldsymbol{x})\log P(\boldsymbol{x}|\omega)\} = \tag{21}$$

$$= \sum_{x\in\mathcal{X}}\frac{\lambda(\boldsymbol{x})}{\Lambda^*_\omega}P^*(\boldsymbol{x}|\omega)\log P(\boldsymbol{x}|\omega). \tag{22}$$

The proof is complete since the last expression is maximized by $P(\cdot|\omega) = \tilde{P}(\cdot|\omega)$.

From the above Lemma it follows that, for the sample-size approaching infinity, the Bayesian decision-making is invariant with respect to boosting. More exactly, we prove:

Lemma 2. The posterior probabilities

$$p^*(\omega|\boldsymbol{x}) = \frac{P^*(\boldsymbol{x}|\omega)p^*(\omega)}{P^*(\boldsymbol{x})}, \quad \omega\in\Omega, \quad \boldsymbol{x}\in\mathcal{X} \tag{23}$$

are invariant with respect to weighting of the conditional distributions $P^*(\boldsymbol{x}|\omega)$ by a positive bounded function $\lambda(\boldsymbol{x})$ in the sense of the Eq.

$$\tilde{p}(\omega|\boldsymbol{x}) = \frac{\tilde{P}(\boldsymbol{x}|\omega)\tilde{p}(\omega)}{\tilde{P}(\boldsymbol{x})} = p^*(\omega|\boldsymbol{x}), \quad \omega\in\Omega \tag{24}$$

which is satisfied for the weighted distributions

$$\tilde{P}(\boldsymbol{x}|\omega) = \frac{\lambda(\boldsymbol{x})}{\Lambda^*_\omega}P^*(\boldsymbol{x}|\omega), \quad \tilde{p}(\omega) = \frac{p^*(\omega)\Lambda^*_\omega}{\Lambda^*_0}, \tag{25}$$

$$\tilde{P}(\boldsymbol{x}) = \frac{\lambda(\boldsymbol{x})}{\Lambda^*_0}P^*(\boldsymbol{x}), \quad \boldsymbol{x}\in\mathcal{X}, \tag{26}$$

$$\Lambda^*_0 = \sum_{\omega\in\Omega}\Lambda^*_\omega p^*(\omega). \tag{27}$$

Proof. The assertion of the Lemma is easily verified by making substitutions from (25) - (27) into (24).

Let us recall finally that, in practical experiments, the conditional distributions $\tilde{P}(\boldsymbol{x}|\omega)$ estimated from weighted data have to be used for classification along with the estimates of the recomputed prior probabilities $\tilde{p}(\omega)$. In particular we should use the estimates

$$\tilde{p}(\omega) \approx \frac{p^*(\omega)\Lambda(\mathcal{S}_\omega)}{|\mathcal{S}_\omega|\Lambda_0}, \quad \Lambda_0 = \sum_{\omega\in\Omega}\frac{p^*(\omega)\Lambda(\mathcal{S}_\omega)}{|\mathcal{S}_\omega|}. \tag{28}$$

in order to avoid an unnecessary increase of the classification error.

4 Computational experiments

The weighted version of EM algorithm has been applied to recognize totally unconstrained hand-written numerals from the database of Concordia University, Montreal (cf. [6, 7] for methodological details). The class-conditional distributions were approximated in the original 1024-dimensional space by the structural distribution mixtures (9). In all experiments the parameters $f(m|\omega), \theta_{mn}$ and ϕ_{mn} were estimated by means of the EM algorithm of Section 3 for all class-conditional distributions. The iterative procedure (15)-(18) was started randomly with different number of components M and component-specific parameters r. For each combination of the parameters M, r first the non-weighted EM algorithm was applied (i.e. with $\lambda(\boldsymbol{x}) = 1$ for all $\boldsymbol{x} \in \mathcal{X}$) and then the modified EM algorithm weighted with the function $\lambda(\boldsymbol{x})$. In general, 25 iterations of EM algorithm were sufficient to achieve reasonable convergence.

In all experiments the weighting function $\lambda(\boldsymbol{x})$ has been defined by means of the initially estimated parameters as the entropy of the posterior distribution $p(\omega|\boldsymbol{x})$:

$$\lambda(\boldsymbol{x}) = C_0 + \sum_{\omega \in \Omega} -p(\omega|\boldsymbol{x}) \log p(\omega|\boldsymbol{x}). \tag{29}$$

Here the entropy has been chosen as a measure of the decision complexity. A positive constant C_0, $(C_0 = 0.2 - 0.5)$ has been added because in the high-dimensional input space ($N = 1024$) the entropy is frequently zero and a large portion of training data would be suppressed completely.

Tabulka 1: The effect of data weighting on recognition accuracy in 6 independent computational experiments. Here ϵ, ϵ_w are the (independently tested) classification errors achieved with the non-weighted- and weighted training sets respectively.

Solution	$\sum M_\omega$	r	ϵ	ϵ_w
1	994	70	.3175	.2110
2	500	100	.2455	.2430
3	696	400	.0555	.0585
4	592	500	.0520	.0445
5	400	500	.0500	.0575
6	498	600	.0450	.0455

We estimated the class-conditional probability distributions in 6 randomly initialized independent computational experiments. Table 1 displays the corresponding total number of mixture components M (column 2) and the number of independent parameters of components r (column 3). Classification error ϵ as verified by the independent test set is given in column 4 and the last column shows the classification error ϵ_w obtained by using weighted data vectors. The number of nonzero structural parameters $r = \sum_n \phi_{mn}$ was identical in all components. In different experiments it has been set to different values between $r = 70$ and $r = 600$ whereby the number of components of the conditional mixtures has been chosen between $M_\omega = 40$ and $M_\omega = 200$.

In accordance with the published experience (cf. [12]) a strong improvement of the classification error by means of data weighting was observed only in case of the worst

classifier 1. In all other experiments the effect of weighting was rather moderate (2, 4) or even counterproductive (3, 5, 6). Nevertheless, as the "weighted" EM algorithm was always started independently with random initial values, the results may be influenced by some less favorable locally optimal solutions.

5 Conclusion

In practical situations the method of boosting proved to be a useful heuristical principle which can essentially improve the recognition performance of weak classifiers. However, in the more exactly defined context of statistical pattern recognition the underlying weighting of data becomes questionable as it may influence the form of the estimated conditional distributions with unhappy consequences for the asymptotic classification error. We have shown that, in case of m.-l. estimation of discrete class-conditional distributions, the Bayesian decision-making is invariant to arbitrary data weighting - provided that the prior probabilities are properly modified (cf. (25), (28)). For this reason the Bayes classification error is also asymptotically invariant with respect to arbitrary boosting provided that the combined classifiers satisfy the conditions of Lemma 1 and Lemma 2.

Reference

[1] A.P. Dempster, N.M. Laird, D.B. Rubin "Maximum likelihood from incomplete data via the EM algorithm." *Journal of the Royal Stat. Soc.*, B 39, 1977, pp.1-38.

[2] T.G. Dietterich, "Ensemble methods in machine learning." In: *Multiple Classifier Systems*, Kittler J., Roli F., (Eds.), Springer, 2000, pp. 1-15.

[3] Y. Freund, R.E. Schapire, "Experiments with a new boosting algorithm." In: *Proc. 13th International Conference on Machine Learning*, Morgan Kaufmann, 1996, pp. 149-156.

[4] C. Furlanello, S. Merler, "Boosting of tree-based classifiers for predictive risk modeling in GIS." In: *Multiple Classifier Systems*, Kittler J., Roli F., (Eds.), Springer, 2000, pp. 220-229.

[5] J. Grim, "Information approach to structural optimization of probabilistic neural networks." In: *Proc. 4th System Science European Congress*, L. Ferrer et al. (Eds.), (pp. 527-540), Valencia: Sociedad Espanola de Sistemas Generales, 1999.

[6] J. Grim, P. Pudil, P. Somol, "Recognition of handwritten numerals by structural probabilistic neural networks." In: *Proceedings of the Second ICSC Symposium on Neural Computation*, Berlin, 2000. (Bothe H., Rojas R. eds.). ICSC, Wetaskiwin, 2000, pp. 528-534.

[7] J. Grim, J. Kittler, P. Pudil, P. Somol, "Combining multiple classifiers in probabilistic neural networks." In: *Multiple Classifier Systems*, Kittler J., Roli F., (Eds.), Springer, pp. 157 - 166.

[8] J.J. Rodriguez-Diez, C.J.A. Gonzalez, "Applying boosting to Similarity literals for time series classification." In: *Multiple Classifier Systems*, Kittler J., Roli F., (Eds.), Springer, 2000, pp. 210-219.

[9] M.I. Schlesinger, "Relation between learning and self-learning in pattern recognition." (in Russian), *Kibernetika*, (Kiev), No. 2, pp. 81-88.

[10] M. Skurichina, R.P.W. Duin, "Boosting in linear discriminant analysis." In: *Multiple Classifier Systems*, Kittler J., Roli F., (Eds.), Springer, 2000, pp. 190-199.

[11] D.M. Titterington, A.F.M. Smith, U.E. Makov, *Statistical analysis of finite mixture distributions.* New York: John Wiley & Sons, 1985.

[12] J. Wickramaratna, S. Holden, B. Buxton, "Perfomance degradation in boosting." In: *Multiple Classifier Systems*, Kittler J., Roli F., (Eds.), Springer, 2001, pp. 11-21.