ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 197-201

On different ensembles of kernel machines

Michiko Yamana, Hiroyuki Nakahara, Massimiliano Pontil,
and Shun-ichi Amari*

Abstract. We study some ensembles of kernel machines. Each machine is first trained on a bootstraped subset of a whole dataset and then, they together are combined linearly by optimizing an objective function. We discuss two different objective functions inspired by boosting methods. The present preliminary experiments show merits and drawbacks of our approach in comparison to standard SVM and bagging SVM.

1. Introduction

Support Vector Machines (SVMs) [1] prove to show remarkable generalization performances in many classification problems, compared to other learning algorithms. However, SVMs suffer from the computational time of their training algorithm, which scales as quadratic at least in the number of training examples. A possibility to overcome this difficulty is to train many machines on a small random subset of the original dataset. There are various ways to combine these SVMs. [2] showed that by simply combining SVMs by average (Bagging), the performance is nearly the same as that of a single SVM but with a significant gain on stability of the ensemble [2]. It is thus natural to expect that a finer choice of the coefficients in the linear combination of SVMs could improve these results. In this paper, we present a systematic procedure to derive optimal combinations of SVMs. This is based on optimization of objective functions (criterion), which were used in boosting optimization methods [3]. Preliminary experiments indicates some advantages in this approach. Other possible approaches which we do not explore here are those based on Bayesian learning - see, e.g., [4] and references therein.

2. Kernel Machines Ensembles

Let $\mathcal{D}$ be a training set, $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{\ell}$, where $(\mathbf{x}_i, y_i) \in \mathrm{I\!R}^n \times \{-1, 1\}$. A kernel machine is a function of the form $f(\mathbf{x}) = \sum_{i=1}^{\ell} \alpha_i y_i \mathcal{K}(\mathbf{x}_i, \mathbf{x})$, where $\mathcal{K}$ is a symmetric and positive definite kernel function, e.g. a Gaussian. The coefficients α_i are determined by solving the following optimization problem:

$$\begin{aligned} &\max\nolimits_{\alpha} \textstyle\sum_{i=1}^{\ell} S(\alpha_i) - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_i \mathcal{K}_{ij} \\ &\text{subject to}: \; 0 \leq \alpha_i \leq C \end{aligned} \tag{1}$$

where $S(\cdot)$ is a cost function, C a constant, and we have defined $\mathcal{K}_{ij} \equiv \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$. SVM is a particular instance of Eq. (1) with the choice of $S(\alpha) = \alpha$. For SVM, points for which $\alpha_i \neq 0$ are called support vectors.
Let us formulate an ensemble of kernel machines. Let T be the number of machines. We generate T subsets of an original data set $\mathcal{D}$, denoted by $\mathcal{D}_1, \cdots, \mathcal{D}_T$.

[*] MJ, HN, and SA are with Laboratory for Mathematical Neuroscience, RIKEN, Brain Science Institute, Saitama, Japan (Email: {*yamana,hiro,amari*}*@brain.riken.go.jp*). MP is with Department of Computer Science, University College London, Gower St., London WC1E 6BT, UK (Email: *m.pontil@cs.ucl.ac.uk*).

Each kernel machine, trained on $\mathcal{D}_t$, is denoted by f_t $(t = 1, \ldots, T)$. Once we obtain these kernel machines $(f_1, \cdots, f_T)$, they are linearly combined to form the ensemble

$$F(\mathbf{x}) = \sum_{t=1}^{T} c_t f_t(\mathbf{x}). \tag{2}$$

The coefficients $\{c_t\}$ are determined by optimizing an objective function which we discuss in Section 3. The coefficient c_t may be constrained to be positive and normalized to 1. Finally, we obtain the hypothesis as $H(\mathbf{x})$ =sign$(F(\mathbf{x}))$.
There are two important issues to be investigated in order to achieve a good performance of these ensembles, namely, (i) how to determine the number of kernel machines, T, and (ii) how to divide the training set $\mathcal{D}$ into the subsets $\mathcal{D}_1, \cdots, \mathcal{D}_T$. In the present study we randomly pick up subsets from $\mathcal{D}$.

3. Optimizing Ensemble Coefficients

Coefficients $\{c_t\}$ are optimized by means of an iterative procedure with a given objective function. Notably, a whole dataset is used in this step. We consider two objective functions (criterion), inspired boosting methods [3]. Since our optimization updates the coefficients simultaneously, our procedure can be regarded as a parallel implementation of the boosting method discussed in [5].

1. Exponential Criterion. This criterion is defined by

$$J_{\mathrm{E}}(\mathbf{c}) = \ln \left\{ \sum_{i=1}^{\ell} \exp\left(-y_i F(\mathbf{x}_i)\right) \right\}.$$

It is used by some boosting methods - see, e.g., the discussion in [3].

2. Maximum Likelihood Criterion. A second natural way to determine the coefficients in Eq. (2) is through maximum likelihood. For convenience, we denote the outputs of kernel machines by $T-$dimensional vector $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_T(\mathbf{x}))$. We then consider a likelihood function $\mathcal{L}(\mathbf{c})$ for the ensemble machines $\mathbf{f}(\mathbf{x})$,

$$\mathcal{L}(\mathbf{c}) \simeq \prod_{i=1}^{\ell} p(y_i, \mathbf{x}_i, \mathbf{f}(\mathbf{x}_i) \mid \mathbf{c}) = \prod_{i=1}^{\ell} p_{\mathbf{c}}(y_i \mid \mathbf{f}(\mathbf{x}_i), \mathbf{x}_i)\ p(\mathbf{f}(\mathbf{x}_i), \mathbf{x}_i) \tag{3}$$

where $p(\mathbf{f}(\mathbf{x}), \mathbf{x})$ is a joint probability density of $\mathbf{f}(\mathbf{x})$ and $\mathbf{x}$. We assume the following exponential form for $p_{\mathbf{c}}(y \mid \mathbf{f}(\mathbf{x}), \mathbf{x})$:

$$p_{\mathbf{c}}(y \mid \mathbf{f}(\mathbf{x}), \mathbf{x}) = \exp(yF(\mathbf{x}) - \psi(\mathbf{c}, \mathbf{x})) \tag{4}$$

where $\psi(\mathbf{c}, \mathbf{x})$ provides the normalization factor, which is determined by

$$\exp(\psi(\mathbf{c}, \mathbf{x})) = \sum_{y=\pm 1} \exp(yF(\mathbf{x})) = \exp(\sum_t c_t f_t(\mathbf{x})) + \exp(-\sum_t c_t f_t(\mathbf{x})). \tag{5}$$

Then, we can rewrite the likelihood function as

$$\mathcal{L}(\mathbf{c}) = \prod_{i=1}^{\ell} \frac{1}{1 + \exp(-2y_i F(\mathbf{x}_i))} \tag{6}$$

where we omit the factor involving the joint probability density $p(\mathbf{f}(\mathbf{x}), \mathbf{x})$ since it is independent of $\mathbf{c}$. Finally we define $J_{\mathrm{M}}(\mathbf{c}) = -\ln \mathcal{L}(\mathbf{c})$ and minimize $J_{\mathrm{M}}(\mathbf{c})$ with respect to $\mathbf{c}$. We briefly comment on the relation between the two above criterion. Taking a Taylor series expansion of J_{M} around $F(\mathbf{x}) = 0$ up to second order gives

$$J_{\mathrm{M}}(\mathbf{c}) = \sum_i \ln\left[1 + \exp(-2y_i F(\mathbf{x}_i))\right] \simeq \sum_i \left[\exp(-y_i F(\mathbf{x}_i)) + \ln 2 - 1\right]. \quad (7)$$

Thus, $J_{\mathrm{M}}(\mathbf{c})$ is equivalent to J_{E} up to second order of $F(\mathbf{x})$ [3]. To minimize the objective functions (J_{E} or J_{M}), we use the steepest descent method which consists in iteratively updating the parameter vector $\mathbf{c}$, where an initial value $\mathbf{c}^0$ should be provided (e.g., the uniform solution $c_t^0 = 1/T$, for $t = 1, \cdots, T$). The $n+1$ step is given by

$$\mathbf{c}^{\mathrm{n}+1} = \mathbf{c}^{\mathrm{n}} - a^{\mathrm{n}} \ \mathrm{grad} J(\mathbf{c}^{\mathrm{n}}) \quad (8)$$

which can be written in terms of the objective function as

$$J(\mathbf{c}^{\mathrm{n}+1}) = J(\mathbf{c}^{\mathrm{n}} - a^{\mathrm{n}} \ \mathrm{grad} J(\mathbf{c}^{\mathrm{n}})).$$

The parameter a^{n} is determined by minimizing the r.h.s. of the above equation. We skip the explicit formula of $\mathrm{grad} J(\mathbf{c}) = J(\mathbf{c})/\partial \mathbf{c}$ for lack of space. We note that another possible choice for $\mathrm{grad} J(\mathbf{c})$ would be the natural gradient [7], but we do not investigate this issue in the present paper.

3.1. Learning Convex Ensembles

We also consider minimizing J_{E} or J_{M} under the constraint $\sum_t c_t = 1,\ 0 \leq c_t \leq 1$. To this end we minimize the objective function

$$J_{\mathrm{M}}(\mathbf{c}; \theta) = -\ln \mathcal{L}(\mathbf{c}) + \theta \sum_t c_t \quad (9)$$

by means of the the above steepest descent method. The parameter θ is a Lagrange multiplier which is determined in order to satisfy the constraint $\sum_t c_t = 1$. To impose the condition $c_t > 0$, we define the variable $c_t = (c_t')^2$ and minimize Eq. (9) w.r.t. c_t'. We remark that such ensembles can be studied theoretically trough stability considerations. In the following we show the qualitative idea of this approach - see [6, 2] for more information. Let F^i be the ensemble combination trained on the set $\mathcal{D} \backslash \{(x_i, y_i)\}$. The stability β_ℓ of the ensemble combination is the smallest positive real number such that

$$E[\|F - F^i\|] \leq \beta_\ell, \text{ for every } i \in \{1, \ldots, \ell\}$$

where E denote the average w.r.t. the training set and $\|\cdot\|$ is the L_1-norm w.r.t. the probability of the input. The difference between test and training error, denoted by Δ, is controlled by the stability parameter. Formally $\Delta = O(\sqrt{\beta_\ell/\delta})$ with probability at least $1-\delta$. Likewise, we can define the stability of the underlying learning algorithm, which we denote by $\hat{\beta}_\ell$. In the case of Bagging, [2] showed a link between β_ℓ and $\hat{\beta}_\ell$. Formally, it establishes that $\beta_\ell = O(\hat{\beta}_k k/\ell)$. The same result can be extended to the case of convex combinations. This implies that if $\hat{\beta}_\ell$ is sub-linear in ℓ^{-1}, the ensemble stability improves and, so, the difference between test and training error decreases, preventing the occurrence of overfitting. For SVM, $\hat{\beta}_\ell = C$ [6] (i.e. the stability is independent of ℓ), where C is the parameter defined in Problem (1).

4. Experimental Results

We carried out experiments on three datasets from the UCI Benchmark Repository maintained by G. Rätsch[1]: Breast-cancer (277 data, 200 used for training), Banana (5300 data, used 400 training), and Diabetes (768 data, 468 used for training). We compared bagging (Bag) and SVM to the proposed SVM ensembles. These are: the exponential criterion (EC), the maximum likelihood criterion (ML), EC with convex constrains (EC-C), and ML with convex constrains (ML-C). Each of the T machines in the ensemble was trained on a bootstrap set of equal size. All SVMs in the ensemble as well as the single SVM were trained using the same Gaussian kernel and the same parameter C. For convenience this parameter and the variance in the Gaussian were chosen so as to optimize the single SVM. We studied how the training error and the test error behave in relation to two factors, namely the number of machines, T, and the percentage of samples used to train each machine, denoted by P. In all experiments, errors were averaged over 10 random trials. Each cell in the tables below shows the average training error with its standard deviation (upper line) and the average test error with its standard deviation (lower line). Table 1 shows the effects of imposing the convexity constraints on the coefficients $(0 \leq c_t \leq 1, \sum_t c_t = 1)$ in relation to different methods and different values of the percentage P, on the Diabetis dataset. Here T equals 30. Note that the ensemble SVMs without constraints shows "too small" training errors. However, the test errors tend to become bigger, which indicates the occurrence of "overfitting. This tendency appears in both the maximum likelihood criterion and the exponential criterion. We thus consider that the constraints is especially important for noisy (hard) datasets. The same trend was observed on the Breast-cancer and Banana datasets. Table 2 shows the results of the comparison among bagging and the two criterion with the convex constraint, over different values of T and P on the three datasets. The proposed criterion show better performances than that of Bagging and better or nearly the same performance of a single SVM. (Cancer: training = 17.0, test = 26.0, $C = 15$, $\sigma = 5$; Banana: training = 5.25, test = 12.2, $C = 316, \sigma = 1$; Diabetes, training = 20.3, test = 23.7, $C = 100, \sigma = 20$). This finding is important since the computational complexity of an SVM scales at least quadratically in ℓ, while we expect the ensemble SVM to be nearly linear in ℓ. However, the dataset used in the present experiments were too small to enlighten this effect. Looking again at Table 1, we note that an EC-C ensemble with $P = 2\%$ achieves better performance than a single SVM. As a final remark, note that the free parameters of the SVM (regularization and variance

Table 1: Experimental results (Diabetis dataset). See text for a description.

Method/P	2%	3%	4%	5%	9%

Bag	24.3(1.48)
	25.4(1.29)
		24.3(1.48)
		25.4(1.29)
			22.4(0.97)
			25.0(0.86)
				22.4(1.18)
				24.1(1.18)
					21.3(0.61)
					23.6(0.67)

EC	20.1(0.55)
	25.0(0.79)
		20.1(0.55)
		25.0(0.79)
			20.1(0.68)
			24.8(0.54)
				20.3(0.82)
				24.5(1.00)
					20.2(0.72)
					24.2(0.80)

EC-C	21.8(0.63)
	22.8(0.75)
		21.8(0.63)
		22.8(0.75)
			22.0(0.44)
			23.2(0.99)
				21.6(0.30)
				22.9(0.76)
					21.7(0.33)
					22.9(0.59)

ML	19.0(0.62)
	24.8(0.96)
		19.0(0.62)
		24.8(0.96)
			19.2(0.51)
			24.5(0.75)
				19.4(0.63)
				24.7(0.86)
					19.5(0.62)
					24.2(0.76)

ML-C	21.4(0.46)
	23.4(0.77)
		21.4(0.46)
		23.4(0.77)
			21.1(0.43)
			23.6(0.49)
				21.0(0.43)
				23.7(0.20)
					20.6(0.37)
					23.9(0.42)

[1] Available at http://ida.first.gmd.de/˜raetsch/data/benchmarks.htm

of the kernel) were optimized w.r.t. the single SVM architecture. Thus all the presented results may have some bias in favor of the single SVM.

Table 2: Experimental results for Cancer (left), Banana (center), and Diabetis (right). See text for a description.

T/P		5%	10%	20%	10%	20%	41%	5%	10%	20%
10	Bag	24.8	23.8	22.5	11.8	8.53	7.63	22.8	22.0	21.0
		28.4	27.4	25.3	15.1	13.1	11.9	25.1	24.1	24.2
	EC-C	24.6	23.0	22.2	10.0	8.50	7.63	21.6	22.0	21.7
		27.0	25.8	26.6	13.4	12.5	11.8	24.4	23.7	23.7
	ML-C	23.9	22.7	21.5	9.75	8.00	7.33	21.2	21.5	21.0
		26.1	25.2	26.6	13.3	12.3	11.7	23.9	23.2	24.4
20	Bag	25.0	23.9	21.8	10.8	8.53	7.33	22.0	21.5	20.7
		27.1	26.4	25.5	14.4	12.8	12.0	24.4	23.5	23.6
	EC-C	24.2	22.6	22.1	8.63	8.13	7.13	21.7	21.6	21.6
		25.7	24.9	25.3	12.8	12.2	11.9	22.7	22.9	23.1
	ML-C	23.8	21.4	20.1	8.40	7.93	6.50	21.0	20.8	20.9
		25.7	24.9	25.3	12.5	12.2	11.8	23.4	23.3	23.5
30	Bag	24.2	23.5	22.0	9.35	8.55	7.30	22.4	21.1	20.6
		27.7	26.8	25.6	13.7	12.8	12.1	24.1	24.3	23.3
	EC-C	23.7	21.9	20.0	8.70	7.80	7.05	21.6	21.8	21.5
		25.6	24.3	24.5	12.6	12.5	11.8	22.9	22.7	23.4
	ML-C	23.3	20.9	19.6	8.08	7.65	6.63	21.0	20.7	20.6
		25.5	24.5	24.7	12.3	12.5	11.9	23.7	23.8	23.5

5. Conclusions

We presented kernel machine ensembles which are based on the minimization of boosting-like criterion such as the exponential criterion and maximum likelihood criterion. An important feature of our approach is that the machines are trained on small random subsets of an initial training set, which can be easily implemented on a parallel computer. We also investigated the case that the machines are combined under the additional constraint that the coefficients form a convex combination. Experimental results show that this constraint tends to avoid overfitting, especially when the dataset contains a lot of noise. The experiments provide some indications that the ensemble improves performance over the single SVM that would be trained with the whole dataset, while potentially reducing the computational complexity of training.

Acknowledgments: M.Y. would like to thank Masato Inoue for his help in computer programming.

References

[1] *The nature of statistical learning theory* V. N. Vapnik, Springer, 1995.

[2] "Leave-one-out Error, Stability, and Generalization of Voting Combinations of Classifiers", T. Evgeniou, M. Pontil, A. Elisseeff, *Machine Learning*, 2003.

[3] *The elements of statistical learning: Data mining, inference, and prediction*, T. Hastie, R. Tibshirani, and J. H. Friedman, Springer, 2001.

[4] *Least Squares Support Vector Machines*, J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle, World Scientific, Singapore, 2002.

[5] "Boosting and maximum likelihood for exponential models", G. Lebanon and J. Lafferty, Proc. of NIPS'01, MIT Press, 2001.

[6] "Stability and generalization", O. Bousquet and A. Elisseeff, J. of Machine Learning Research, **2** 499-526, 2002.

[7] "Natural Gradient Works Efficiently in Learning", S. Amari, Neural Comp. **10** 251-276, 1998.