ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

Linear kernel combination using boosting

Alexis Lechervy[1]*, and Philippe-Henri Gosselin [1] and Frédéric Precioso [2]

1- ETIS - CNRS/ENSEA/Université de Cergy-Pontoise,
6 avenue du Ponceau, F-95000 Cergy-Pontoise - France

2- I3S - UMR7271 - UNS CNRS
2000, route des Lucioles, 06903 Sophia Antipolis - France

Abstract. In this paper, we propose a novel algorithm to design multi-class kernels based on an iterative combination of weak kernels in a schema inspired from the boosting framework. Our solution has a complexity linear with the training set size. We evaluate our method for classification on a toy example by integrating our multi-class kernel into a kNN classifier and comparing our results with a reference iterative kernel design method. We also evaluate our method for image categorization by considering a classic image database and comparing our boosted linear kernel combination with the direct linear combination of all features in a linear SVM.

1 Context

Recent machine learning techniques have demonstrated their power for classifying data in challenging contexts (large databases, very small training sets, huge training sets, dealing with user interactions...). However, emerging problems are pushing these methods to their limits with sereval hundred of image categories to be classified, with millions of images both in training and testing datasets.

A key component in a kernel machine is a kernel operator which computes for any pair of instance their inner-product in some induced vector space. A typical approach when using kernels is to choose a kernel before the training starts. In the last decade, many researches have been focused on learning the kernel to optimally adapt it to the context and propose a computational alternative to predefined base kernels. Such approaches are particularly interesting in the context aforementioned of huge image databases with hundreds of object categories. In fact, it is pretty much unlikely that a unique base kernel would be adequate to separate all categories while grouping all data from the same category. For instance, Gehler et al. [1] propose to linearly combine several base kernels in order to improve the performance of the major kernel function hence designed for multi-class supervised classification.

In this paper, we propose to design a linear combination of weak base kernels using the boosting paradigm, similarly to [2]. However, we focus on a strategy for multi-class learning using many different features.

Before designing a method for the combination of base kernels, it is necessary to define a target kernel $K^\star$ that reflects the ideal case on a given training set. In the case of two-class context, this kernel can be defined by $K^\star(\mathbf{x}_i, \mathbf{x}_j) = 1$

*Thanks to DGA agency for funding.

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

if the training samples $\mathbf{x}_i$, $\mathbf{x}_j$ are in the same class, -1 otherwise. This can be expressed on a training set as the Gram matrix $\mathbf{K}^\star = \mathbf{L}\mathbf{L}^\top$, where L_i is the class label of the i^{th} training sample. Then, the design of the kernel combination $K(.,.)$ is driven by the optimization of a criterion between the Gram matrix of kernel combination $\mathbf{K}$ and target kernel $\mathbf{K}^\star$. Several criteria have been proposed among which *class separability* [3] and *data centering* [4]. The most classic one is probably the *kernel alignment* proposed by Cristianini et al. [5] which is defined as the cosine of the angle between the two Gram matrices $\mathbf{K}_1$ and $\mathbf{K}_2$ of two kernels k_1 and k_2.

In the context of multi-class classification, the definition of target kernel is not straightforward. Let $\mathbf{L}$ be the $n_X \times n_C$ matrix of annotations that is for n_X training samples and n_C classes, with $L_{ic} = 1$ if the i^{th} training sample belongs to c class and -1 otherwise. Then a naive target kernel can be defined by the Gram matrix $\mathbf{K}_\mathbf{L} = \mathbf{L}\mathbf{L}^\top$. In the following we denote the outer-product $\mathbf{X}\mathbf{X}^\top$ by $\mathbf{K}_\mathbf{X}$. A first improvement of this target kernel is the introduction of centering, which accounts the high unbalance of multi-class context. Thus, it is recommended to center all Gram matrix (target and combination) using the centering matrix $\mathbf{H}$, and centered Kernel-Target Alignment $\mathcal{A}_\mathbf{H}$ as in [6]. Furthermore, Vert [7] proposed a solution that handles the case of the classes with correlations.

Our learning method based on boosting is presented in section 2. In the following section 3, we present a target for weak learner. The section 4 presents some experiments of classifying both toy data and real data from a standard image database. We then conclude and present the perspectives of this work.

2 Linear kernel combination using boosting

To overcome the inter-class dependency, we propose to consider the matrix $\mathbf{Q}$ of the QR decomposition of $\mathbf{HL}$. We only select the columns where the diagonal element of $\mathbf{R}$ is not zero. Thus $\mathbf{Q}$ is a $n_X \times n_C$ full rank matrix, assuming that classes are independent. Our target Gram matrix is then defined as $\mathbf{K}_\mathbf{Q} = \mathbf{Q}\mathbf{Q}^\top$. The specific form of this target matrix is further exploited to find the optimal boosting increment (i.e. the kernel evolution direction towards the next best major kernel alignment). Furthermore, as we will see in the next section, the properties of the QR decomposition ensure the convergence of our strategy.

The second contribution is a new boosting method for kernel design. We design a kernel function $K(.,.)$ as a linear combination of base kernel functions $k_t(.,.)$:

$$K_T(\mathbf{x}_i, \mathbf{x}_j) = \sum_{t=1}^{T} \beta_t k_t(\mathbf{x}_i, \mathbf{x}_j)$$

where $\mathbf{x}_i$ is the feature vector for training sample i, for instance colors and textures in the case of images. We consider base kernel functions defined by $k_t(\mathbf{x}_i, \mathbf{x}_j) = f_t(\mathbf{x}_i) f_t(\mathbf{x}_j)$, where $f_t(.)$ is a function built by a weak learner.

In order to build the combination, we work on finite matrix on a given

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

training set $\mathbf{X}$, which leads to the following expression, with $\mathbf{f}_t = f_t(\mathbf{X})$ and $\mathbf{F}_t = (\mathbf{f}_1\ \mathbf{f}_2\ \dots\ \mathbf{f}_t)\boldsymbol{\beta}^{\frac{1}{2}}$:

$$\mathbf{K}_t = \sum_{s=1}^{t} \beta_s \mathbf{f}_s \mathbf{f}_s^\top = \mathbf{F}_t \mathbf{F}_t^\top$$

In the following we mix functions (f, K,...) and their values on the training set (written in bold $\mathbf{f}$, $\mathbf{K}$...).

We select base kernels iteratively in a boosting scheme:

$$\mathbf{K}_t = \mathbf{K}_{t-1} + \beta_t \mathbf{f}_t \mathbf{f}_t^\top \Leftrightarrow \mathbf{F}_t = (\mathbf{F}_{t-1}\ \beta_t^{\frac{1}{2}} \mathbf{f}_t)$$

where $\beta_t, f_t = \zeta(\mathbf{F}_t)$ is the result of the problem solved by ζ :

$$\zeta(\mathbf{F}) = \text{argmax}_{\beta>0, f}\, \mathcal{A}_{\mathbf{H}}(\mathbf{F}\mathbf{F}^\top + \beta \mathbf{f}\mathbf{f}^\top, \mathbf{Q}\mathbf{Q}^\top)$$

For a f given, the optimal β can be solved analytically by methods of linear algebra. We build f using least mean squares (LMS) and a target function presented in the following section.

3 Weak learners optimal target

In order to train weak learners, we need to choose a target function $f^\star$, a function that leads to the best alignment. In the case of two-class context, it can be defined by $f^\star(\mathbf{x}_i) = 1$ if training sample i is in the first class, -1 otherwise. However, in the case of multi-class context, this not obvious, since we need to spread each class data around equidistant centers [8, 7, 9].

We propose to consider the centers of (orthonormalized) classes in the space induced by the current combination kernel $\mathbf{K}_t = \mathbf{F}_t \mathbf{F}_t^\top$:

$$\mathbf{G}_t = \mathbf{Q}^\top \mathbf{F}_t$$

The rows of $\mathbf{G}_t$ are coordinates of class centers.

The idea of our method is to move each center to make it equidistant from others. In [8], Vapnick states that the largest possible margin is achieved when the c vertices of $(c-1)$-dimensional unitary simplex are centered onto the origin. A sufficient means to achieve this properties is to build c orthonormal vertices, whose projection on a $(c-1)$ dimension space is the unitary simplex. In our case, that means that an ideal target set of class centers $\mathbf{G}_t^\star$ is such that $\mathbf{G}_t^\star (\mathbf{G}_t^\star)^\top$ is proportional to the identity matrix $\mathbf{Id}_{c,c}$.

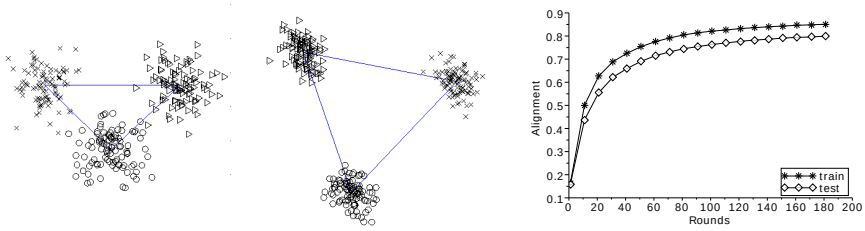
If we apply the Cauchy-Schwarz inequality to the alignment, we can show a similar observation:

$$\mathcal{A}_{\mathbf{H}}(\mathbf{K}_t, \mathbf{Q}\mathbf{Q}^\top) = 1 \Longleftrightarrow \xi(\mathbf{G}_t)\xi(\mathbf{G}_t)^\top = \mathbf{Id}_{c,c} \text{ with } \xi(\mathbf{G}) = \sqrt{\frac{\|\mathbf{Q}\mathbf{Q}^\top\|_F}{\|\mathbf{H}\mathbf{F}(\mathbf{H}\mathbf{F})^\top\|_F}}\mathbf{G}$$

The aim is to find weak learners that lead to this identity. In other words we are looking for a function $f^\star$ such as:

$$\|\mathbf{Id}_{c,c} - \xi(\mathbf{G}_t)\xi(\mathbf{G}_t)^\top\| > \|\mathbf{Id}_{c,c} - \xi(\mathbf{G}_t^\star)\xi(\mathbf{G}_t^\star)^\top\|$$

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.



(a) (b) (c)

Fig. 1: Converge of the method to regular 2-simplex in 3 classes case after 2 steps(a) and 45 steps(b). Evolution of alignment for 10 classes(c)

As we proceed iteratively, we can only focus on the new column $\mathbf{g}^\star$ of $\mathbf{G}_t^\star = (\mathbf{G}_t\ \mathbf{g}^\star)$. A good candidate is:

$$\mathbf{g}^\star = \sqrt{1 - \lambda \frac{\|\mathbf{Q}\mathbf{Q}^\top\|_F}{\|\mathbf{H}\mathbf{F}_\mathbf{T}(\mathbf{H}\mathbf{F}_\mathbf{T})^\top\|_F}}\mathbf{v}$$

where λ is the smaller eigen value of $\mathbf{G}_t\mathbf{G}_t^\top$ and $\mathbf{v}$ the eigenvector.

Thanks to this result, we can select the target function $\mathbf{f}^\star$ for weak learners. It can be shown that $\mathbf{f}^\star = \mathbf{Q}\mathbf{g}^\star$ always leads to the convergence of the whole algorithm.

4 Experiments and results

In a first experiment (Figure 1) we illustrate the convergence of the method to a regular $(\#class - 1)$-simplex.

We consider a toy dataset with 2 classes and 200 examples per class (100 for training and 100 for testing). We use a pool of 10 features of 2 dimensions. For each class c and each feature f, a center $C_{c,f}$ is picked up at uniformly random in $[0, 1]^2$. Each example is described by the 10 features with a gaussian with 0.5 standard deviation centered on $C_{c,f}$. For visualisation we use PCA on F_t and select the two first dimensions. After 2 iterations Figure 1 (a), our algorithm separates each class but the distances between classes are uneven. After 45 iterations Figure 1 (b), the classes barycenters converge to the vertex of a regular 2-simplex.

In a second experiment, we compare experimentally our method with a recent Multiple Kernel Learning algorithm proposed by Kawanabe et al. [6] which computes the optimal linear combination of features. We consider a toy dataset with 10 classes and 60 examples per class (30 for training and 30 for testing). We use a pool of 25 features of 16 dimensions. For each class c and each feature f, a center $C_{c,f}$ is picked up at uniformly random in $[0, 1]^{16}$. Each example is described by the 25 features with a gaussian with 0.5 standard deviation centered on $C_{c,f}$.

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

k	1	2	3	4	5	6	7	8	9	10
Kawanabe	20	24	16	13	**8**	**6**	**5**	**5**	**4**	**5**
Our method	**14**	**18**	**11**	**9**	**8**	7	7	6	5	**5**

Fig. 2: % of error with k-nearest neighbor algorithm

Classes	1	2	3	4	5	6	7	8	9	10	All
lab16	12.4	9.4	24.6	16.0	9.4	11.2	9.0	8.0	27.2	10.9	14.0
lab32	10.7	8.1	45.7	27.1	34.2	15.9	10.0	9.1	27.0	25.6	22.8
lab64	10.0	12.5	47.9	28.5	37.5	19.1	9.9	16.4	31.7	50.3	26.3
lab128	18.7	24.7	46.6	28.8	34.1	20.0	16.0	16.5	33.2	50.0	28.7
qw16	14.0	46.8	55.5	15.1	7.5	14.6	8.3	21.0	25.6	11.2	22.0
qw32	38.5	52.2	60.2	22.2	7.7	15.9	8.9	36.0	36.9	25.6	30.4
qw64	43.0	53.1	63.4	22.0	14.2	18.8	13.2	43.3	37.5	36.2	34.4
qw128	47.9	57.4	65.6	25.8	14.8	20.3	21.6	45.5	33.2	48.6	37.6
All	52.1	58.2	72.2	37.4	38.5	27.1	26.7	44.4	39.5	56.1	45.3
Our	**52.2**	**63.1**	**75.9**	**43.8**	**41.6**	**27.6**	**27.2**	**52.7**	**41.9**	**56.6**	**48.3**

Fig. 3: Average precision in % (VOC2006) for linear SVM

Figure 1(c) shows that the alignment of our method increases at each iteration on both training and testing data. When comparing the two methods with respect to their alignment results for the same dataset and the same features, their alignment is 0.772469 on train and 0.773417 on test while our method, as seen on Figure 1(c), reaches after 180 iterations an alignment of 0.8507977 on train and of 0.8034482 on test.

Both methods have linear complexity in the number of training samples but Kawanabe et al. approach [6] has a quadratic complexity in the number of features while our method is linear.

We have also compared our features and Kawanabe features in a multi-class classification context. On the same second synthetic dataset, we classified test data with k-nearest neighbor classifier (kNN) for different k (Figure 2). Our method outperforms Kawanabe et al. when considering fewer neighbors. It aggregates more examples per class.

In a third experiment, we have also compared our method on real data. We evaluate the performance of our algorithm on Visual Object Category (Voc)2006 dataset. This database contains 5304 images provided by Microsoft Research Cambridge and Flickr. Voc2006 database contains 10 categories (cat, car, motorbike, sheep ...). All images can belong to several categories. There are two distinct sets, one for training and one for testing with 9507 annotations. We create our weak kernels from 8 initial features: normalized (L2) histograms of 16, 32, 64, 128-bins for both color CIE L*a*b and quaternion wavelets.

Then we use linear SVM (normalized with L2) to compare the features extracted from the final F matrix with the initial features. We have also evaluated the performance of each extracted feature form F again a feature concatenating all 8 initial features (Figure 3). For all classes our methods reaches higher average precision.

We numerically assess the performance of our method on Oxford Flowers 102

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

[10]. As the authors of this base [10], we use four different χ^2 distance matrices to describe different properties of the flowers. Results show that our method improves the performance from 72.8% [10] to 77.8%.

5 Conclusion

In this paper, we propose a new algorithm to create a linear combination of kernels for multi-class classification context. This algorithm is based on an iterative method inspired from boosting framework. We thus reduce both the computation time of final kernel design and the number of weak kernels used.

Considering the QR decomposition leads to a new solution to address the problem of inter-class dependency and provides quite interesting properties to develop an interactive method.

The proposed solution is linear in the number of training samples.

Our method shows good results both on a toy dataset when compared to a reference kernel design method and on a real dataset in image classification context. We are currently working on a generalization of our method to collaborative learning context. Indeed, the same algorithm can target a kernel matrix for collaborative learning by considering that initial annotation matrix stores all previous retrieval runs.

References

[1] P. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *IEEE International Conference on Computer Vision*, pages 221–228, 2009.

[2] K. Crammer, J. Keshet, and Y. Singer. Kernel design using boosting. In *Advances in Neural Information Processing Systems*, pages 537–544. MIT Press, 2003.

[3] L. Wang and K. Luk Chan. Learning kernel parameters by using class separability measure. In *Advances in Neural Information Processing Systems*, 2002.

[4] M. Meila. Data centering in feature space. In *International Workshop on Artificial Intelligence and Statistics*, 2003.

[5] N. Cristianini, J. Shawe-Taylor, A. Elisseff, and J. Kandola. On kernel target alignement. In *Advances in Neural Information Processing Systems*, pages 367–373, Vancouver, Canada, December 2001.

[6] M. Kawanabe, S. Nakajima, and A. Binder. A procedure of adaptive kernel combination with kernel-target alignment for object classification. In *ACM International Conference on Image and Video Retrieval*, 2009.

[7] R. Vert. Designing a m-svm kernel for protein secondary structure prediction. Master's thesis, DEA informatique de Lorraine, 2002.

[8] V. Vapnick. *Estimation of Dependences Based on Empirical Data*. Springer-Verlag, 1982.

[9] P.H. Gosselin and M. Cord. Feature based approach to semi-supervised similarity learning. *Pattern Recognition, Special Issue on Similarity-Based Pattern Recognition*, 39:1839–1851, 2006.

[10] M-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.