Constructing discriminative biorthogonal bases for classification

Wit Jakuczun
Warsaw University of Technology

October 21, 2004

Abstract We present a method for constructing discriminative biorthogonal bases for classification. In the proposed method we use idea of the *lifting scheme* a method for constructing biorthogonal wavelets and *Support Vector Machines* used for building classifiers. Combining those two ideas resulted in the construction of classifiers which are based on information selected form the available data and still yield very good classification accuracies. We have shown that our method can be treated as a simple feature extractor for other classification algorithms or as a method that produces a set of classifiers whose outputs may be combined to give final classification.

Keywords: lifting scheme, feature extraction, pattern classification, support vector machines, combining classifier

1 Introduction

Many classification algorithms such as artificial neural networks induce classifiers which have good accuracy but do not give an insight into the real process which is hidden behind the problem. Although predictions are made with high precision such classifiers do not answer the question "Why?". Even such algorithms as decision trees or rule inducers very often produce enormous classifiers analysis of which is almost intractable by the human mind. It is even worse when those algorithms are used for problems of signal classification such as EEG. Very often, for biologists, good accuracy without an explanation of the classification process is useless. They need both accuracy and comprehensibility.

In this article we describe an approach which can help in building classifiers which are very accurate and comprehensible simultaneously. This method is based on the idea of the *lifting scheme* developed by Wim Sweldens [Sweldens(1998)]. The Lifting scheme is used for constructing biorthogonal wavelet bases using only spatial domain in contrast to the classical approach in which the frequency domain is used. As original lifting scheme did not give us enough freedom in incorporating adaptation we used a modified version called *updade-first* proposed in [Claypoole et al.(1998)Claypoole, Baraniuk, and Nowak].

Assume we act in space $\mathbb{R}^n$. The Lifting scheme is a method in which given vector $x \in \mathbb{R}^n$ is expanded in a biorthogonal wavelet base

$$x = \sum_{i=1}^{n} \alpha_i \phi_i$$

where $\alpha_i = \left\langle \tilde{\phi}_i, x \right\rangle$. Vectors $\{\phi_i\}_{i=1}^n$ and $\{\tilde{\phi}_i\}_{i=1}^n$ are biorthogonal in the sense

that

$$\left\langle \phi_i, \tilde{\phi}_j \right\rangle = \delta_{ij}$$

where $\delta_{ij} = 1$ if $i = j$ and 0 otherwise. A very important feature of vectors $\{\tilde{\phi}_i\}_{i=1}^n$ is that they are nonzero only for few indices. It implies that for calculating $\left\langle \tilde{\phi}_i, x \right\rangle$ only part of the vector x is needed. This feature is called *locality*.

The aim of method presented in this article is to find a biorthogonal base $\{(\phi_i, \tilde{\phi}_i)\}_{i=1}^n$ in which the base coefficients $\alpha_i = \left\langle \tilde{\phi}_i, x \right\rangle$ are as discriminative as possible for classified signals. More specifically we assume that a training set $X = \{(x_i, y_i) : x_i \in \mathbb{R}^n, y_i \in \{-1, +1\}\}_{i=1}^m$ is given. For each base vector $\tilde{\phi}_j$ we get a vector of coefficients $\alpha^j \in \mathbb{R}^m$

$$\alpha^j(k) = \left\langle \tilde{\phi}_j, x_k \right\rangle$$

For each such vector we can find a number $b^j \in \mathbb{R}$ called bias for which

$$\operatorname{sgn}(\alpha^j(k) + b^j) = y_k$$

for as many as possible indices $k \in \{1, \ldots, m\}$.

For constructing bases we used the idea of *Support Vector Machines (SVM)* developed by Vladimir Vapnik [Vapnik(1998)]. SVM proved to be one of the best classifier inducers. Combining the power of SVM and the locality feature of the designed base we were able to build classifiers with very good classification accuracy which are also easily interpreted. We presents only experiments based on artificial datasets. They allowed us to verify the usefulness of our method for classification problems.

2 Method Description

This section is divided into three subsections. In the first subsection we will shortly describe the modification of the *lifting scheme* [Sweldens(1998)] called *update-first* [Claypoole et al.(1998)Claypoole, Baraniuk, and Nowak]. In the second we will discuss *linear predictors*. In the third section linear predictors will be generalised into *nonlinear predictors*. The generalisation will be based on the idea of *Support Vector Machines (SVM)* by [Vapnik(1998)].

2.1 Update-first modification of lifting scheme

Here we will describe a modification of the *lifting scheme*. This modification, called *update-first lifting scheme* allowed us to exploit the idea of SVM for designing adaptive biorthogonal bases in $\mathbb{R}^n$. For simplicity we assume that $n = 2^s$ for some $s \in \mathbb{N}$[1]. The method consists of three main steps

- **Split** In this step we divide signal x into two disjoint parts x_e (even indexed samples) and x_o (odd indexed) samples):

$$\begin{aligned} x_e(k) &= x(2k) \quad k = 0, 1, \ldots, n/2 - 1 \\ x_o(k) &= x(2k+1) \quad k = 0, 1, \ldots, n/2 - 1 \end{aligned}$$

[1]This assumption is not necessary but it makes analysis much easier.

- **Update** From even and odd indexed samples we create a coarse approximation c of original signal x:

$$c(k) = (x_e(k) + x_o(k))/2 \quad k = 0, 1, \ldots, \frac{n}{2} - 1$$

The vector c is half the length of the vector x.

- **Predict** Using c and x_o we create vector d of *wavelet coefficients*:

$$d(k) = x_o(k) - P^k(c, L^k) \quad k = 0, \ldots, \frac{n}{2} - 1$$

where $P^k(c, L^k)$ is a prediction operator, $L^k \in \mathbb{N}$. This operator can be any function which uses L^k indices of vector c. Quantities $d(k)$ are called *wavelet coefficients*. As with vector c, d is also half the length of the original vector x. Mostly L^k is much less than $\frac{n}{2}$ (which is the length of the vector c) so each wavelet coefficient depends only on small part of the original signal. This feature is called *locality*.

By iterating these three steps using output c as an input for the next iteration we get a complete set of coefficients corresponding to some biorthogonal base in $\mathbb{R}^n$. It is worth noting that this procedure can be inverted very easily by reversing the three steps. Moreover all computations can be done in place.

2.2 Predictors based on inner products

One of the most natural forms of the operators $P^k(c, L^k)$ is an inner product.

$$P^k(c, L^k) = \left\langle \tilde{c}, p^k \right\rangle$$

where $p^k \in \mathbb{R}^{L^k}$ is a coefficients vector and $\tilde{c}$ consists of L^k samples from c. In our method we use the following algorithm for choosing $\tilde{c}$[2]. If k and L^k ($k = 0, 1, \ldots, \frac{n}{2} - 1$) fulfills

- $0 \leq k < \frac{L^k}{2} - 1$ then we select $\tilde{c} = [c(0), \ldots, c(L^k - 1)]$
- $\frac{L^k}{2} - 1 \leq k < \frac{n}{2} - \frac{L^k}{2}$ then we select $\tilde{c} = [c(k - \frac{L^k}{2} - 1), \ldots, c(k + \frac{L^k}{2})]$
- $\frac{n}{2} - \frac{L^k}{2} \leq k < \frac{n}{2}$ then we select $\tilde{c} = [c(\frac{n}{2} - L^k), \ldots, c(\frac{n}{2} - 1)]$

Now $d(k)$ is given by the following formula

$$d(k) = x_o(k) - P^k(c, L^k) = x_o(k) - \left\langle \tilde{c}, p^k \right\rangle$$

2.3 Nonlinear biorthogonal bases for classification

Suppose we are given the training data $X = \{(x_i, y_i)\}_{i=1,\ldots,l}$, $y_i \in \{-1, 1\}$, $x_i \in \mathbb{R}^n$ generated independently at random according to some fixed but unknown distribution D over $\mathbb{R}^n \times \{-1, 1\}$. The problem of classification is to learn mapping $x \to y$ for any pair (x, y) generated by D using only given training data X. On of the methods for solving the classfication problem is the method

[2]We are assuming that L^k is even

of finding a maximal margin hyperplane [Vapnik(1998)]. This method require the solution of the following optimisation problem:

$$\min_{w,b,\psi} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{l}\xi_i$$

subject to constraints

$$\begin{aligned} y^i(\langle x^i, w\rangle + b) &\geq 1-\xi_i \\ \xi_i &\geq 0 \\ &\forall i = 1,\ldots,l \end{aligned}$$

where $w \in \mathbb{R}^n$, $b \in \mathbb{R}$, $C \in \mathbb{R}$ and $C > 0$.

Let us return to the prediction operators described above. For each such operator P^k we build a new training data $\tilde{X}^k = \{(\tilde{c}_i^k, y^i)\}_{i=1,\ldots,l}$ where $\tilde{c}_i^k$ is created from x^i as described previously ($\tilde{c}_k^i \in \mathbb{R}^{L^k}$). Now we can find coefficients p^k solving following optimisation problem:

$$\min_{p^k,b^k,C^k} \frac{1}{2}\|p^k\|_2^2 + C^k\sum_{i=1}^{l}\xi_i$$

subject to the constraints

$$\begin{aligned} y^i(x_o^i(k) + \langle \tilde{c}_k^i, p^k\rangle + b^k) - 1 &\geq 0 \\ \xi_i &\geq 0 \\ &\forall i = 1,\ldots,l \end{aligned}$$

where $p^k \in \mathbb{R}^{L^k}$, $b^k \in \mathbb{R}$, $C^k \in \mathbb{R}$ and $C^k > 0$[3].

It can be shown [Vapnik(1998)] that the optimal coefficients p^k are given by the following formula

$$p^k = \sum_{i\in I_{SV}} \alpha^i y^i \tilde{c}_k^i$$

where I_{SV} is subset of coefficients $1 \leq i \leq l$ for which $\alpha_i > 0$. Vectors $\tilde{c}_k^i$ for $i \in I_{SV}$ are called *support vectors*.

Now coefficients $d^i(k)$ can be calculated as follows

$$d^i(k) = x_o^i(k) + \sum_{j\in I_{SV}} y^j\alpha^j \left\langle \tilde{c}_k^j, \tilde{c}_k^i \right\rangle$$

It is easily seen that coefficients $d^i(k)$ are either smaller or bigger than bias b^k depending on y_i and that we do not need to calculate p^k directly.

As in SVM method we can use *kernel* functions for calculating inner products. That would give us following formula for $d^i(k)$

$$d^i(k) = x_o^i(k) + \sum_{j\in I_{SV}} y^j\alpha^j K(\tilde{c}_k^j, \tilde{c}_k^i)$$

The advantage of using kernels is that we can find hyperplanes in highly dimensional spaces.

The most frequently used kernels are

[3]Although it is possible to choose a unique parameters L^k, C^k and σ for each P^k we decided to use global values of those parameters in our experiments.

- Polynomial: $K(v,w) = (\langle v,w\rangle + 1)^d$, $d \in \mathbb{N}$
- Radial Basis Function (RBF): $K(v,w) = \exp(-\sigma\|v-w\|^2)$, $\sigma \in \mathbb{R}$, $\sigma > 0$
- Sigmoid: $K(v,w) = \tanh(\kappa\langle v,w\rangle - \delta)$, $\kappa, \delta \in \mathbb{R}$

In our experiments we use RBF kernel function.

3 Building classifiers using nonlinear biorthogonal bases

In this section we will describe two possible applications of the proposed method. In the first subsection we will show that the method can be treated as a local feature extractor for classifiers such as decision trees. In the second subsection we will present a method of constructing classification based on voting schemes.

Assume that we are given a training set

$$X = \{(x_i, y_i)\}_{i=1}^m$$

where $x_i \in \mathbb{R}^n$ and $y_i \in \{-1, +1\}$.

One can think of the proposed method in one of two ways: (local) feature extractor as in [Saito(1994)] or finite set of classifiers.

3.1 Local feature extractors

By a local feature extractor we will understand any feature extractor, that is a mapping

$$\Phi : \mathbb{R}^n \to \mathbb{R}^m$$

where mainly $m << n$. They are called *local* because we are looking for such a mapping Φ that uses only a part of each example $x_i \in X$. The proposed predictors are such *local* mappings. To be more precise, after applying our method we can use any method for selecting coefficients. In our experiments we used well known decision-tree induction algorithm called C4.5[4].

3.2 Combining classifiers by voting

As was mentioned above, we can treat results of the proposed method in a different way. To understand this approach we need to realize that we can classify our examples using each coefficient $d(k)$ by applying this simple rule

$$\text{class}(x_i, d(k), b^k) = \text{sgn}(d(k) + b^k)$$

where $\text{sgn}(x) = +1$ for $x \geq 0$ and -1 for $x < 0$. We used the following algorithm

- To each training example $x_i \in X$ assign weight $w_i = \frac{1}{m}$.
- Find coefficient $d(k)$ with the highest possible accuracy and biggest possible *margin*, where the *margin* is defined by the following expression

$$\text{margin}(d(k), X) = \sum_{i=1}^{m} y_i w_i \text{class}(x_i, d(k), b^k)$$

[4]We used Weka [Ian H. Witten(1999)] implementation of C4.5

- For the coefficient $d(k)$ set weight equal to

$$\alpha_k = \log((1-\epsilon_k)/\epsilon_k)$$

where $\epsilon = \frac{|\{(x_i,y_i)\in X:\, \mathrm{class}(x_i,d(k),b^k)\neq y_i\}|}{|X|}$ is called *classification error.*

- Adjust examples' weights according to the following rule

$$w_i \rightarrow w_i \exp(-y_i \mathrm{class}(x_i, d(k), b^k))$$

- Repeat until $\epsilon_k > \frac{1}{2}$ or the number of selected coefficients is sufficient.
- Combine the output of selected coefficients by voting scheme

$$\mathrm{class}(x_i) = \mathrm{sgn}\left(\sum_{i=1}^{L} \alpha_i \mathrm{class}(x_i, d(k_i), b^{k_i})\right)$$

A careful reader will notice that this algorithm is a duplicate of the AdaBoost algorithm [Freund and Schapire(1995)]. In fact there is a slight difference since we are choosing from a given set of already build classifiers and in AdaBoost each classifier is build on a reweighted training set.

3.3 Multiclass classification problems

Our method was developed for twoclass classification problems but there are many problems for which number of decision classes is greater than 2. For such problems we acted in the following manner

- Let C be the number of decision classes and set $X = \{(x_i, y_i)\colon y_i \in 1, \ldots, C\}_{i=1}^{m}$ be a given training set
- For each pair (i, j) such that $i \neq j$ and $i, j \in 1, \ldots, C$ we build a classifier $\Psi_{(i,j)}$ which separates classes i and j. As $\Psi_{(i,j)} \equiv \Psi_{(j,i)}$ then we need to build $\binom{C}{2}$ classifiers.
- To classify a new example x we use the following formula

$$\mathrm{class}(x) = \arg\max_{i=1,\ldots,C} |\{\Psi_{(i,j)}(x) = i : j \neq i, j \in \{1, \ldots, C\}\}|$$

4 Results of experiments

We tested our method on five synthetic datasets. In all experiments we used RBF kernels. Parameters C and σ were chosen by applying MCCV-10 (Monte Carlo version of CV-10). Presented results are obtained on a separate test set.

The first three datasets were taken from Breiman [Breiman(1998)]. They are called: twonorm, threenorm and ringnorm. They are 32-dimension, 2-class datasets.

Next two datasets were taken from Saito [Saito(1994)]. They are called: waveform and shape (also known as Cylinder, Bell and Funel (CBF)). Waveform is 32-dimension, 3-class dataset and Shape is 128-dimension, 3-class dataset.

4.1 Local feature extractors

In this experiment we wanted to check whether coefficients produced by our method may lead to better classification accuracy than the original coefficients for decision-tree classifiers. As an exemplary classifier we used a well known algorithm called C4.5 taken from *Weka* package [Ian H. Witten(1999)].

Dataset	**Misclassification ratio**		**Tree size**	
	Original	New	Original	New
Twonorm	0.215	0.015	35	3
Ringnorm	0.150	0.048	23	13
Threenorm	0.302	0.216	51	29
Waveform	0.290	0.183	-	-
Shape	0.088	0.062	-	-

Table 1: Effect of feature extraction for C4.5. Numbers are misclassification ratios.

The results presented in Table 1 show that using our method as a feature extractor leads to a significant improvement of classifiers accuracy with simultaneous size reduction.

4.2 Voting

Unfortunately coefficients chosen by C4.5 were not always the best possible. That led us to the idea of combining the output of a few best coefficients. It turned out that combining a few coefficients, which separately gave almost perfect accuracy on the training set, resulted in an increase of accuracy on the test set. The results presented in Table 2 are much better than the results in

Dataset	**Misclassification ratio**	
	3 coefficients	15 coefficients
Twonorm	0.007	0.007
Ringnorm	0.070	0.047
Threenorm	0.170	0.187
Waveform	0.211	0.178
Shape	0.025	0.018

Table 2: Misclassification ratios for voting scheme. We were combining 3 and 15 coefficients.

Table 1. The explanation of this accuracy increase is very simple. In the previous experiment we noted that trees built by C4.5 using coefficients produced by our method very often had only one node. It means that decisions made by those trees were based on only one coefficient. Moreover, such algorithms as C4.5 stop searching when the first coefficient which fulfils the searching criteria. In proposed voting scheme, we search for coefficients with a wide margin which correctly classify previously misclassified examples. We believe that voting makes classification more stable and robust. From our experiment we can also see that combining more classifiers does not always lead to better classifica-

tion accuracy. It is quite optimistic as combining smaller number of coefficients results in more comprehensible classifier.

4.3 Conclusions and future work

Presented method gave very optimistic results on artificial datasets. The misclassification ratios were small and obtained classifiers were very simple. The proposed voting scheme gave very good results even for small number of coefficients. In future we plan to experiment with real datasets and other methods of combining classifiers such as stacking [Wolpert(1990)] or rough mereology [Polkowski and Skowron(1996)].

References

[Breiman(1998)] L. Breiman. Arcing classifiers, 1998. URL `http://citeseer.ist.psu.edu/breiman98arcing.html`.

[Claypoole et al.(1998)Claypoole, Baraniuk, and Nowak] R. Claypoole, R. Baraniuk, and R. Nowak. Adaptive wavelet transforms via lifting, 1998. URL `http://citeseer.ist.psu.edu/claypoole98adaptive.html`.

[Freund and Schapire(1995)] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory*, pages 23–37, 1995. URL `http://citeseer.ist.psu.edu/article/freund95decisiontheoretic.html`.

[Ian H. Witten(1999)] Eibe Frank Ian H. Witten. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, 1999.

[Polkowski and Skowron(1996)] L. Polkowski and A. Skowron. Rough mereology: A new paradigm for approximate reasoning, 1996. URL `http://citeseer.ist.psu.edu/polkowski96rough.html`.

[Saito(1994)] Naoki Saito. *Local Feature Extraction and Its Application Using a Library of Bases*. PhD thesis, Yale University, 1994. URL `http://www.math.ucdavis.edu/~saito/publications/saito_phd.html`.

[Sweldens(1998)] Wim Sweldens. The lifting scheme: A construction of second generation wavelets. *SIAM Journal on Mathematical Analysis*, 29(2):511–546, 1998. URL `http://citeseer.ist.psu.edu/sweldens98lifting.html`.

[Vapnik(1998)] Vladimir Vapnik. *Statistical Learning Theory*. John Wiley & Sons, 1998.

[Wolpert(1990)] D. H. Wolpert. Stacked generalization. Technical Report LA-UR-90-3460, Los Alamos, NM, 1990. URL `http://citeseer.ist.psu.edu/wolpert92stacked.html`.