Classification using a sparse combination of basis functions

Kornél Kovács * András Kocsor *

Abstract

Combinations of basis functions are applied here to generate and solve a convex reformulation of several well-known machine learning algorithms like certain variants of boosting methods and Support Vector Machines. We call such a reformulation a Convex Networks (CN) approach. The nonlinear Gauss-Seidel iteration process for solving the CN problem converges globally and fast as we prove. A major property of CN solution is the sparsity, the number of basis functions with nonzero coefficients. The sparsity of the method can effectively be controlled by heuristics where our techniques are inspired by the methods from linear algebra. Numerical results and comparisons demonstrate the effectiveness of the proposed methods on publicly available datasets. As a consequence, the CN approach can perform learning tasks using far fewer basis functions and generate sparse solutions.

1 Introduction

Numerous scientific areas such as optical character and speech recognition, speaker verification, bioinformatics and pharmacology nowadays significantly depend on statistical machine learning algorithms of artificial intelligence. The common feature of these areas - artificial knowledge embedded in applications - is retrieved from pre-collected databases in a statistical way. Recently the size of the data sets for calibrating the methods has grown due to advances in global communication networks like the Internet. Processing this extra amount of data requires effective methods that store the extracted information in a compact and easily retrievable form.

One of the most prevalent machine learning algorithms - Artificial Neural Networks (ANN) [3] - meets these requirements as it has compact form with a fast evaluation. However the solution provided by the learning phase is only a local minima of the objective function, which makes the networks trained on the same database inconsistent. The ubiquitous Support Vector Machine (SVM) method [6, 9, 18] leads to a quadratic programming task whose own global optima defines the compactness of the information retrieved. This kind of functioning can be

*Research Group on Artificial Intelligence of the Hungarian Academy of Sciences, H-6720 Szeged, Aradi vértanúk tere 1., Hungary, e-mail: {kkornel,kocsor}@inf.u-szeged.hu

beneficial since preliminary assumptions are not required, but this is also why the technique might not be applicable in every case. Our aim is to define an algorithm which combines the advantages of the methods and, in particular, it has global optima even with controlled sparsity.

Now we will briefly outline the contents of the paper. First we state the pattern classification problem and derive the so called Convex Networks (CN) method from a constrained optimization formulation in Eq. (8). The nonlinear Gauss-Seidel iteration technique in Definition 2 for solving the CN problem converges globally as shown in the Optimization section without proof. To demonstrate CN's flexibility the original SVM quadratic programming task is re-expressed in a CN form. In the next section we introduce heuristics for controlling the sparsity of the solution. In the numerical tests and comparisons section we demonstrate the practical applicability of CN compared with ANN and SVM. Lastly, we round off with our conclusions and some ideas for future research.

2 Convex networks

Tasks in machine learning often lead to classification and regression problems where models employing a convex objective function might be beneficial. Consider the problem of classifying n points in a compact set $\mathcal{X}$ over $\mathbb{R}^m$, represented by $\mathbf{x}_1, \ldots, \mathbf{x}_n$, according to the membership of each point $\mathbf{x}_i$ in the classes $\{1, \ldots, c\}$ as specified by $y_1, \ldots, y_n$. A multiclass problem can be transformed into a set of binary classification tasks $y_i \in \{-1, +1\}$, which is in many ways like the one-against-all method [20] or the output coding scheme [13]. Thus our investigation can be restricted to the problem of the binary classification without any loss of generality.

Solutions to classification problems in practice are usually based on the model-method where the parameters of a fixed model structure are set by statistics-based optimization. The structure can depend on compact mathematical models [3, 18] or it could apply the points themselves of the available database [8]. Models accomplish the separation by estimating the probability density functions of different classes [1], or by utilizing a separator surface between the points. In both cases we need to look for models which return the following probabilities:

$$P(y \mid \mathbf{z}) \qquad y \in \{-1, +1\}, \quad \mathbf{z} \in \mathcal{X}. \tag{1}$$

The latter case is the discriminative approach where the separator surface is defined by the following set for a fixed $\gamma \in \mathbb{R}$

$$\{\mathbf{z} \mid f(\mathbf{z}) = \gamma,\ \mathbf{z} \in \mathcal{X}\}, \qquad f : \mathcal{X} \to \mathbb{R}. \tag{2}$$

The classification of an arbitrary point $\mathbf{z}$ is based on the sign of $f(\mathbf{z}) - \gamma$, and the probabilities in Eq. (1) could be derived by taking the amplitude of this quantity.

Now let S denote a finite set of continuous basis functions

$$S = \{f_1(\mathbf{x}), \ldots, f_k(\mathbf{x})\}, \qquad f_i : \mathcal{X} \to \mathbb{R} \tag{3}$$

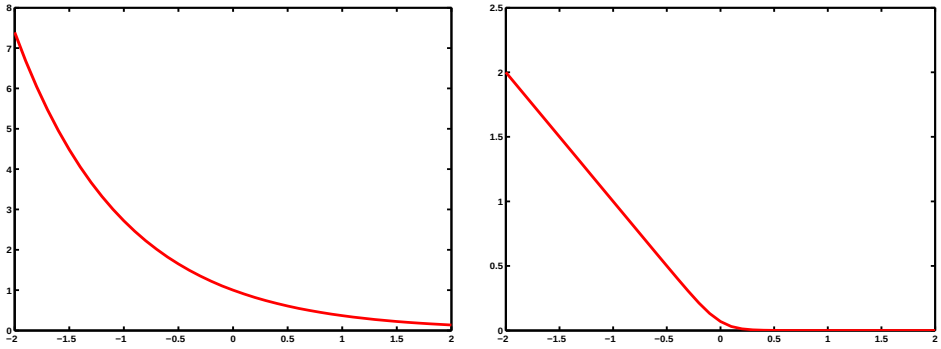

Figure 1: Possible loss functions

and the optimal separator surface of discriminative approach in Eq. (2) is searched for in the linear subspace of basis functions, $f \in Span(S)$, where

$$Span(S) = \left\{ h : \mathcal{X} \to \mathbb{R} \mid h(\mathbf{x}) = \sum_{i=1}^{k} \alpha_i f_i(\mathbf{x}),\ \mathbf{x} \in \mathcal{X},\ \boldsymbol{\alpha} \in \mathbb{R}^k \right\}. \tag{4}$$

Generally the optimality criterion is based on a special indicator of the sample points

$$y_i f(\mathbf{x}_i) \qquad 1 \leq i \leq n, \tag{5}$$

whose amplitudes are proportional to point-surface distances, positive values representing the well separated cases. Recalling that separable classification problems have an infinite number of separator surfaces that can classify the sample points perfectly, we introduce a twice continuously differentiable, monotone decreasing, lower bounded and convex loss-function $L : \mathbb{R} \to \mathbb{R}$ [9]. Of the many possibilities two candidates are shown in Fig. 1. Using a loss-function the separation measure $g(\boldsymbol{\alpha})$ can be defined for a function $f \in Span(S)$ and samples $\mathbf{x}_1, \ldots, \mathbf{x}_n$ by

$$g(\boldsymbol{\alpha}) = \sum_{i=1}^{n} L\left(y_i \sum_{j=1}^{k} \alpha_j f_j(\mathbf{x}_i) \right). \tag{6}$$

2.1 Optimization methods

A machine learning method can be regarded as a multivariate regression problem where the probabilities in Eq. (1) need to be approximated. The parameters of the applied model can be optimally set only if the estimated function is known over the whole space. The problem of approximating the parameters based on sparse sample data is ill-conditioned and the classical way of solving it is to use

regularization theory [17]. According to this theory the optimal separator surface has the minimal separation measure of Eq. (6) with a regularization term

$$\begin{array}{rl} \min_{\boldsymbol{\alpha}} & \sum_{i=1}^{n} L\left(y_i \sum_{j=1}^{k} \alpha_j f_j(\mathbf{x}_i)\right) + \lambda \boldsymbol{\alpha}^T A \boldsymbol{\alpha} \\ \text{s.t.} & \boldsymbol{\alpha} \in \mathbb{R}^k \end{array} \tag{7}$$

where $\lambda > 0$ and $A \in \mathbb{R}^{k \times k}$ is an arbitrary symmetric positive-definite matrix.

In practical applications constraints can be employed on the subspace of basis functions in the form of $\boldsymbol{\alpha} \in \mathcal{A} \subseteq \mathbb{R}^k$ where $\mathcal{A}$ is a non-empty, closed, convex set. We will restrict our investigation here to the case where the domain is a product of non-empty intervals, i.e. $\mathcal{A} = \mathcal{A}_1 \times \ldots \times \mathcal{A}_k$. The formalism includes the unconstrained task of Eq. (7) where $\mathcal{A}_i = (-\infty, \infty)$. The final form of the Convex Networks (CN) problem is

$$\begin{array}{rl} \min_{\boldsymbol{\alpha}} & \sum_{i=1}^{n} L\left(y_i \sum_{j=1}^{k} \alpha_j f_j(\mathbf{x}_i)\right) + \lambda \boldsymbol{\alpha}^T A \boldsymbol{\alpha} \\ \text{s.t.} & \boldsymbol{\alpha} \in \mathcal{A} = \mathcal{A}_1 \times \ldots \times \mathcal{A}_k \end{array} \tag{8}$$

It can be readily seen that the objective function in this equation is twice continuously differentiable, lower bounded and convex. Moreover, every level set is bounded. Actually, Eq. (8) is a convex programming task which can be solved by one of many techniques [2].

The Sequential Quadratic Programming (SQP) methods [7, 10] focus on the solving of Kuhn-Tucker (KT) equations, which are sufficient conditions for global optima in the convex programming case. SQP is an iterative algorithm for solving a quadratic programming subproblem at each step. The convergence of SQP is super-linear due to the special update rule of second order information about KT equations.

In contrast to SQP, the Gauss-Seidel (GS) iteration technique is a kind of convergent algorithm that modifies one component of the solution at each step - in other words a simple convex optimization subproblem with one variable is solved at each step. Hence the resource requirements of the method remain bounded even for large-sized datasets. That is why we prefer to use the GS method to solve a CN task.

Definition 1 (projection mapping)

$$[\ ]^p : \mathbb{R}^k \to \mathcal{A} \qquad [\boldsymbol{\alpha}]^p = \mathbf{z} \Leftrightarrow \|\boldsymbol{\alpha} - \mathbf{z}\|^2 = \min_{\mathbf{y} \in \mathcal{A}} \|\boldsymbol{\alpha} - \mathbf{y}\|^2$$

Definition 2 (constrained Gauss-Seidel iteration)

$$\boldsymbol{\alpha}_i^{t+1} = \left[\boldsymbol{\alpha}_i^t - \gamma \nabla_i f(\mathbf{z}_i^t)\right]_i^p$$

where

$$\gamma > 0, \qquad \mathbf{z}_i^t = (\alpha_1^{t+1}, \ldots, \alpha_{i-1}^{t+1}, \alpha_i^t, \ldots, \alpha_k^t), \qquad \boldsymbol{\alpha}^{t+1} = \mathbf{z}_{k+1}^t.$$

During the iteration process each component of the actual solution $\boldsymbol{\alpha}^t$ is successively upgraded by the gradient rule. If the solution falls outside the domain it will be replaced by the nearest point of the set with the aid of the projection mapping. The constrained GS iteration method is convergent for every function $\tau : \mathcal{A} \to \mathbb{R}$ over a non-empty, convex and closed set $\mathcal{A}$, where τ is twice continuously differentiable and lower bounded. Moreover, the gradient should be a Lipschitz function and there must exist a $\delta > 0$ such that $0 < \delta \leq \nabla^2_{ii}\tau(\boldsymbol{\alpha}))$. The limit point of the iteration is the extreme of the function over $\mathcal{A}$ [2].

However it can be proved that the Lipschitz condition respecting the gradient can be ignored if every level set of the function is bounded. Therefore the constrained Gauss-Seidel iteration procedure with low resource requirements is proposed for solving the CN task.

2.2 Methods involved

The CN formalism includes several well-known machine learning algorithms e.g. variants of boosting methods [11, 12] and Support Vector Machines (SVM) [6, 14, 16].

The standard SVM problem is given by the following for some $C > 0$, taking into account the fact that the bias in the separator hyperplane may be eliminated from the equation [15]:

$$\begin{array}{rrcl} \min_{\mathbf{w}} & C\mathbf{e}^T\boldsymbol{\xi} + \frac{1}{2}\mathbf{w}^T\mathbf{w} & & \\ \text{s.t.} & YX\mathbf{w} + \boldsymbol{\xi} & \geq & \mathbf{e} \\ & \boldsymbol{\xi} & \geq & \mathbf{0} \end{array}, \tag{9}$$

where Y is a diagonal matrix with $y_1, \ldots, y_n$ along its diagonal, $X = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^T$ and $\mathbf{e}$ is a column vector of ones of arbitrary dimension. To solve this optimization problem we have to find the saddle point of the Lagrangian

$$\begin{array}{rl} \max_{\mathbf{w},\boldsymbol{\alpha}} & C\mathbf{e}^T\boldsymbol{\xi} + \frac{1}{2}\mathbf{w}^T\mathbf{w} - \boldsymbol{\alpha}^T(YX\mathbf{w} + \xi - \mathbf{e}) \\ \text{s.t.} & \boldsymbol{\alpha}, \boldsymbol{\xi} \geq \mathbf{0} \end{array} \tag{10}$$

The parameters that maximize the Lagrangian must satisfy the conditions

$$\mathbf{w} = X^T Y\boldsymbol{\alpha} \qquad \mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{e}. \tag{11}$$

These set of constraints can be employed in the original problem of Eq. (9) because the duality gap disappears when the objective function is convex

$$\begin{array}{rrcl} \min_{\boldsymbol{\alpha}} & C\mathbf{e}^T\boldsymbol{\xi} + \frac{1}{2}\boldsymbol{\alpha}^T YKY\boldsymbol{\alpha} & & \\ \text{s.t.} & YKY\boldsymbol{\alpha} + \boldsymbol{\xi} & \geq & \mathbf{e} \\ & \boldsymbol{\alpha} & \geq & \mathbf{0} \\ & -\boldsymbol{\alpha} & \geq & C\mathbf{e} \\ & \boldsymbol{\xi} & \geq & \mathbf{0} \end{array}, \tag{12}$$

where $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$ is the kernel matrix of the sample. Mapping $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a Mercer-kernel [6] which can define some implicit nonlinear transformation of

the original points so that $K = XX^T$ means a linear mapping. For a solution $\boldsymbol{\alpha}$ of Eq. (12), $\boldsymbol{\xi}$ is given by $(\mathbf{e} - YKY\boldsymbol{\alpha})_+$ where

$$(\mathbf{z}_+)_i = \max\{0, z_i\} \qquad i = 1, \ldots, n \tag{13}$$

Exploiting this in Eq. (12) we get

$$\begin{array}{ll} \min_{\boldsymbol{\alpha}} & \sum_{i=1}^{n} \left(1 - y_i \sum_{j=1}^{n} \alpha_j y_j K_{ij}\right)_+ + \frac{1}{2C}\boldsymbol{\alpha}^T YKY\boldsymbol{\alpha} \\ \text{s.t.} & \mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{e} \end{array} \tag{14}$$

which is a CN problem with the following parameters

$$\begin{array}{lll} k = n & L(x) = (1-x)_+ & f_j(\mathbf{z}) = y_j \kappa(\mathbf{z}, \mathbf{x}_j) \\ \lambda = \frac{1}{2C} & A = YKY & \mathcal{A} = [0, C]^n \end{array} \tag{15}$$

if the plus function $(1-x)_+$ is replaced by a very accurate smooth approximation $p(x) = -(1-x) + \frac{1}{\beta}\log(1 + e^{\beta(1-x)})$, $\beta \to \infty$. Actually, it can be shown that as the smoothing parameter β tends to infinity the unique solution of the smoothed problem approaches the unique solution of the equivalent task in Eq. (15) [14].

3 Sparse solutions

The separator surface coded by a CN problem takes the form

$$\{\mathbf{z} \mid \sum_{j=1}^{k} \alpha_j f_j(\mathbf{z}) = \gamma,\ \mathbf{z} \in \mathcal{X}\}, \qquad f_j : \mathcal{X} \to \mathbb{R}. \tag{16}$$

for a fixed threshold $\gamma \in \mathbb{R}$. Basis functions with zero coefficients can be eliminated when evaluating the model and the remaining terms define the complexity of the CN solution. The more the number of zero coefficients the faster the evaluation, which makes the CN method suitable for fast or real-time applications. However the coefficients are determined by the optimal solution of the mathematical programming task, and the parameters can only influence the sparsity by degrading the performance.

For the sake of controlling the complexity the number of basis functions will be restricted by making the following assumption on the CN domain

$$\sum_{i=1}^{k} |sign(\alpha_i)| \leq q \tag{17}$$

Such a condition violates the closed and convex properties of the domain so the suggested nonlinear Gauss-Seidel technique and other iterative methods cannot be applied to the problem. The last remaining approach is the combinatorial selection of basis functions. Our aim is to select from the available basis functions a subset of order q where the classification problem can be optimally solved. This task is NP hard so the only effective way here is to employ heuristics which can be based on the execution of CN with different parameters or their own objective functions. In the next part we will outline methods from the latter group.

3.1 Heuristics

In this section we deal with algorithms that do not use the CN objective function itself during the optimal basis function subset selection of order q.

`RANDOM` The simplest strategy is the random selection approach when we randomly select q basis functions from among the k basis functions. This approach does not have an objective function that can be minimized so we will choose instead the subset with the best performance after several executions.

`MGRAMM` The CN method approximates the optimal separator surface using a linear combination of the basis functions. Hence the approximation can be performed on an orthogonal basis of the function space, as in the case of the result of the Gramm-Schmidt orthogonalization algorithm. Despite this, the dimension of the basis is the rank of the function set which can exceed the desired number q. Moreover, the algorithm generates an orthogonal function system with linear combinations of basis functions instead of selecting the individual functions.

To solve the above we will define a greedy iterative selection strategy based on a modified version of the Gramm-Schmidt orthogonalization algorithm. Among the available basis functions we choose the one with a maximal residual norm after the Gramm-Schmidt process at each step. The result of this greedy method is not the orthogonal function system itself but the basis functions used in the linear combinations.

`GRAMM(q)`
$Y = \{1, \ldots, k\}$; $I = \emptyset$;
`for i = 1...q`
$t = argmax_{j \in Y - I} \ \| f_j - \sum_{l=1}^{i-1} \frac{\langle f_j, f_l^* \rangle}{\langle f_l^*, f_l^* \rangle} f_l^* \|_2$
$I = I \cup \{t\}$;
$f_i^* = f_t - \sum_{l=1}^{i-1} \frac{\langle f_t, f_l^* \rangle}{\langle f_l^*, f_l^* \rangle} f_l^*$;
`return` I;

Assume that the basis functions are elements of L_2 so the dot product is the integral of the product function. When analytical computations of the integrals are not possible we utilize the following approximation in the algorithm using the sample points

$$\langle f, g \rangle = \sum_{i=1}^{n} f(\mathbf{x}_i) g(\mathbf{x}_i) \qquad f, g : \mathcal{X} \to \mathbb{R}. \tag{18}$$

`CORR` The `MGRAMM` method tries to choose an orthogonal basis of the functions with the help of the Gramm-Schmidt process. The choice might be good when the dot product of functions is available. Employing the approximation in Eq. (18) the result of the algorithm will be also just an approximation of the desired basis.

Such an estimation can be carried out in different ways. The orthogonality of the elements in the basis can be also employed, since the mutual correlation coefficients must be zero. Our aim is to select functions such that the squared sum of the element in the correlation matrix should be minimal. Similar to `MGRAMM` this method will be a greedy iterative process and also exploit the fact that the mutual correlation coefficient for normalized functions takes the form of Eq. (18).

```
CORR(q)
  Y = {1,...,k}; I = ∅;
  for i = 1...q
    t = argmin_{j∈Y−I}  Σ_{l=1}^{i−1} ⟨f_j, f_l^*⟩² / ⟨f_j, f_j⟩
    I = I ∪ {t};
    f_i^* = f_t / ⟨f_t, f_t⟩^0.5;
  return I;
```

4 Results

We now demonstrate the effectiveness of the CN approach by comparing its results with other methods. In order to evaluate how well each algorithm classifies an unknown dataset, we performed a tenfold cross-validation on publicly available datasets from the UCI repository [4]. The performance of the CN method was compared with Artificial Neural Networks (ANN) and Support Vector Machines (SVM).

We applied a feed-forward neural network (MLP) with one hidden layer, where the number of hidden neurons was set at three times the class number. The back-propagation learning rule was applied for training. MLP was executed five times on each dataset and then we chose the parameter values which gave the best performance on training sample.

For an impartial comparison we employed our 1-norm SVM implementation where the bias term was absent [15]. Multiclass cases were handled by the one-against-all approach. Additionally, the cosine polynomial kernel we applied made the SVM method nonlinear

$$\kappa\left(\mathbf{x},\mathbf{y}\right)=\left(\frac{\mathbf{x}^T\mathbf{y}}{\left\|\mathbf{x}\right\|\left\|\mathbf{y}\right\|}+\sigma\right)^q, \qquad q\in\mathbb{N},\ \sigma\in\mathbb{R}_+ \tag{19}$$

with parameters $q=3$ and $\sigma=1$.

The basis functions for the CN problem were defined by the above kernel function using the sample points of a training set, as shown in Eq. (14). Thus

$$f_j\left(\mathbf{z}\right)=y_j\kappa\left(\mathbf{z},\mathbf{x}_j\right). \qquad j=1,\ldots,n \tag{20}$$

The coefficients of the basis functions were not restricted in our tests, i.e. we used the domain $\mathcal{A}=(-\infty,\infty)^n$. In the regularization term of Eq. (8) we set the identity matrix equal to A with $\lambda=1$.

	ANN	SVM	CN
balance	89.03	93.55	99.79
	86.35	90.63	**95.41**
bupa	72.01	81.73	80.69
	68.07	**74.39**	71.92
glass	84.24	99.79	100.0
	69.87	84.70	**86.23**
iono	93.35	99.40	99.94
	86.17	91.09	**92.41**
monks	90.64	97.50	99.05
	87.28	95.82	**96.51**
pima	78.68	82.49	80.55
	76.09	75.58	74.82
wdbc	98.71	99.47	100.0
	97.61	**97.62**	96.93
wpbc	85.71	98.47	99.04
	76.41	77.36	**79.63**

Table 1: Ten-fold cross-validation training and testing results on some UCI datasets using three different methods. ANN is a feed-forward neural network with one hidden layer where the number of hidden units was set at three times the class number. SVM used the cosine polynomial kernel defined in Eq. (19) with $q = 3$ and $\sigma = 1$ for nonlinearity. With the help of Eq. (14) the CN method applied the same basis functions.

It turned out that, on most of the datasets tested, the tenfold testing correctness of the CN problem was the highest for these methods. We summarize all these results in Table 1. It confirms that the CN classification method is indeed just as effective as the ubiquitous machine learning algorithms. Moreover, their performances were surpassed in many cases. It can be readily seen that the problem of overfitting the data was present more often in the methods with global optima. It might be explained with the locally optimal solution of the ANN method, which can be regarded as a kind of regularization. Similar results are expected when using sparse heuristics to solve a CN problem.

We also examined the performance of the proposed heuristics controlling CN sparsity. We compared the methods on the Iono database by examining the value of the CN objective function, regardless of how the methods worked. Now the `RANDOM` method chose its best from 5 executions. The results of the heuristics are shown in Fig. 2. We used the performance of the `RANDOM` method as a reference so the results of other algorithms are expressed in percentages. As the reader will notice the `MGRAMM` and the `CORR` approaches achieved similar results, and both of them outperformed the `RANDOM` method here. Despite the fact that these algorithms require computational time the selected basis perform better.

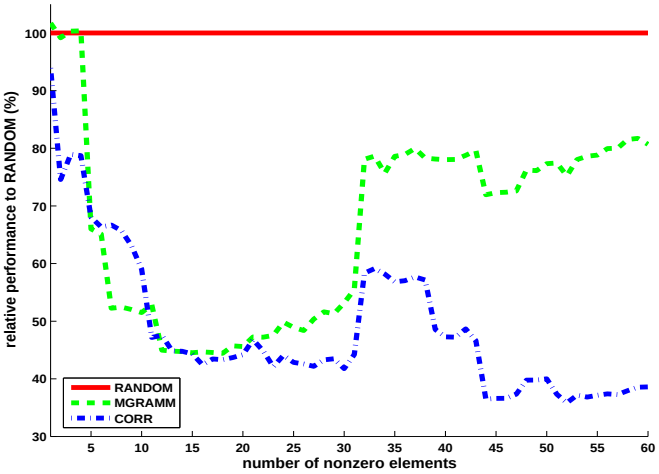

Figure 2: Performances of the proposed heuristics controlling CN sparsity on the Iono database expressed in percentages of the `RANDOM` method result. The CN measures were used as performance indicators regardless of how the methods works.

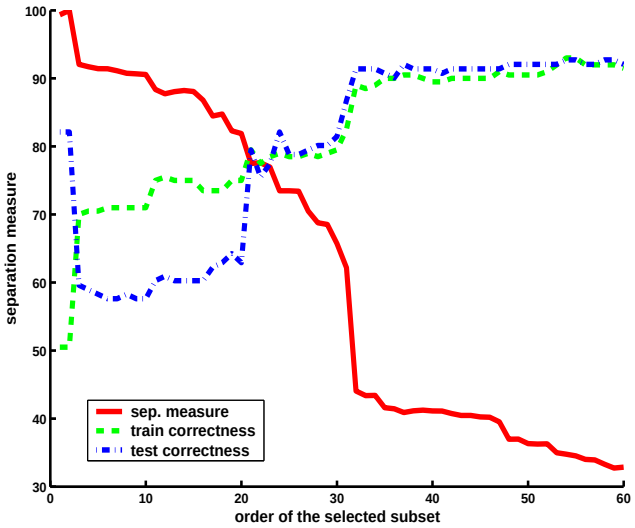

Figure 3: The consistency of the measure in the CN method and its abstraction ability with the aid of the `MGRAMM` method on the Iono database. The decreasing CN measure means a better testing correctness.

During the subset selection we optimize some measures while the abstraction ability is the most important in the machine learning sense. The consistency of the measure in the CN method and its abstraction ability can be seen in Fig. 3 with the aid of the `MGRAMM` method on the previous database. As can readily be seen, the decreasing CN measure value means a better abstraction ability, i.e. testing correctness. Thus the measure of the CN approach might indeed be employed as an objective function of machine learning algorithms.

The performance of heuristics were examined with the help of ten-fold cross-validation. We summarize our results here in Table 2. The sparsity of solutions were maximized using 10%, 20% and 30% of the available functions. The `RANDOM`

`RANDOM` / `MGRAMM` / `CORR`	10%	20%	30%	100%
	95.10	95.25	95.40	
balance	95.25	95.40	95.25	95.41
	95.41	95.10	95.25	
	70.49	71.35	69.14	
bupa	69.14	70.53	**71.61**	71.92
	69.12	69.12	69.42	
	84.75	**89.66**	87.00	
glass	85.16	86.62	85.91	86.23
	85.18	85.16	86.66	
	89.16	**93.19**	92.68	
iono	91.23	92.04	90.54	92.41
	91.58	91.32	91.85	
	93.18	93.70	94.76	
monks	92.94	91.66	90.89	96.51
	92.65	94.40	**95.11**	
	78.51	77.24	75.97	
pima	77.89	76.43	77.87	74.82
	77.62	76.22	76.60	
	97.44	97.27	97.44	
wdbc	97.25	96.93	96.09	96.93
	97.10	96.93	96.93	
	78.27	78.29	77.29	
wpbc	76.37	75.14	73.20	79.63
	74.05	75.93	**79.70**	

Table 2: Ten-fold cross-validation testing results of the Convex Networks method using the heuristics `RANDOM`, `MGRAMM` and `CORR`. The sparsity was controlled by maximizing the number of available basis functions to 10%, 20% and 30% of the complete sets, respectively.

method had the same parameter as that above. As observed, all of the algorithms selected subsets with adequate testing correctness. This kind of capacity reduction in the CN learning method brings about a sort of regularization which is reflected in the results: results with a reduced basis outperform the original ones in many cases. The various algorithms here have their best performance on different tasks. In general, different requirements in the learning phase will lead the user to select one of the available heuristics.

5 Conclusions

We proposed a reformulation of certain machine learning algorithms that includes several well-known nonlinear classification methods. The CN problem can be solved by the convergent nonlinear Gauss-Seidel iteration process, which is sufficiently fast for this task. The numerical results on its abstraction ability show that the CN method can be considered as a rival classification method to both ANN and SVM. Moreover, the sparsity of the CN problem can be effectively controlled by the proposed heuristics. Future work includes a new heuristic based on a CN objective function which can be utilized in very large classification problems. We also plan to use chunking algorithms like those described in [5] for problems which do not fit in the memory.

References

[1] Alder, M. D. *Principles of Pattern Classification: Statistical, Neural Net and Syntactic Methods of Getting Robots to See and Hear,* http://ciips.ee.uwa.edu.au/~mike/PatRec, 1994.

[2] Bertsekas, D.P. And Tsitsiklis, J. N. *Parallel and Distributed Computation: Numerical Methods,* Prentice Hall, 1989; republished by Athena Scientific, 1997.

[3] Bishop, C. M., *Neural Networks for Pattern Recognition,* Oxford University Press, 1995.

[4] Blake, C. L. and Merz, C. J. *UCI repository of machine learning databases,* http://www.ics.uci.edu/ mlearn/MLRepository.html, 1998.

[5] Bradley, P. S. and Mangasarian, O. L. *Massive data discrimination via linear support vector machines,* Optimization Methods and Softwares, vol. 13, pp. 1-10, 2000.

[6] Cristianini, N. And Shawe-Taylor, J. *An Introduction to Support Vector Machines and other kernel-based learning methods,* Cambridge University Press, 2000.

[7] Conn, A. R., Gould, N. I. M., Toint, T. L. *Trust-region methods,* Society for Industrial and Applied Mathematics, 2000.

[8] Duda, R. and Hart, P. *Pattern Classification and Scene Analysis,* Wiley and Sons, New York, 1973.

[9] Evgeniou, T., Pontil, M., Poggio, T. *Regularization Networks and Support Vector Machines,* Advances in Computational Mathematics, Vol. 13/1, pp. 1-50, 2000.

[10] Fletcher, R. *Practical Methods of Optimization,* John Wiley and Sons, 1987.

[11] Freund, Y. and Schapire, R.E. *A decision-theoretic generalization of online learning and an application to boosting,* J. Comput. Syst. Sci., vol. 55/1, pp. 119-139, 1997.

[12] Friedman, J., Hastie, T., Tibshirani, R. *Additive logistic regression: A statistical view of boosting,* The Annals of Statistics, vol. 28/2, pp. 337-407, 2000.

[13] Kong, E. B. and Dietterich, T. *Error-Correcting Output Coding Corrects Bias and Variance* International Conference on Machine Learning, pp. 313-321, 1995.

[14] Lee, Y.-J. and Mangasarian, O. L. *SSVM: A Smooth Support Vector Machine for Classification,* Computational Optimization and Applications, vol. 20/1, pp. 5-22, 2001.

[15] Poggio, T., Mukherjee, S., Rifkin, R., Rakhlin, A., Verri, A. *b,* in Proceedings of the Conference on Uncertainty in Geometric Computations, 2001.

[16] Suykens, J.A.K. and Vandewalle, J. *Least squares support vector machine classifiers,* Neural Processing Letters, 1999.

[17] Tikhonov, A. N. And Arsenin, V. Y. *Solutions of Ill-posed Problems,* W. H. Winston, Washington, D.C., 1977.

[18] Vapnik, V. N. *Statistical Learning Theory,* John Wiley & Sons Inc., 1998.

[19] Wahba, G. *Splines models for Observational Data,* Series in Applied Mathematics, Vol. 59, SIAM, Philadelphia, 1990.

[20] Weston, J. and Watkins, C. *Support vector machines for multiclass pattern recognition,* Proceedings of the Seventh European Symposium On Artificial Neural Networks, 1999.