Improving a basis function based classification method using feature selection algorithms

Kornél Kovács[1], András Kocsor[2,*]
Research Group on Artificial Intelligence
of the Hungarian Academy of Sciences and University of Szeged
H-6720 Szeged, Aradi vértanúk tere 1., Hungary
{[1]kkornel,[2]kocsor}@inf.u-szeged.hu

***Abstract** – The recently introduced Convex Networks (CN) method [1] is a convex reformulation of several well-known machine learning algorithms like certain boosting methods and various Support Vector Machine algorithms. The special feature of the CN method is that it employs a combination of basis functions to solve a classification task of machine learning. The nonlinear Gauss-Seidel iteration process for solving the CN problem converges globally and fast, according to the corresponding proof. The most important property of the CN solution is its sparsity, which means that the number of basis functions with nonzero coefficients is small, and can effectively be controlled by heuristics. The proposed techniques were inspired by an area of artificial intelligence, called Feature Selection. Numerical results and comparisons demonstrate the effectiveness of the proposed methods on publicly available datasets. As will be shown, the CN approach can perform learning tasks using far fewer basis functions and generate sparse solutions.*

***Keywords** – machine learning, kernel methods, feature selection, Gauss-Seidel iteration, sparse solutions*

I. INTRODUCTION

The widespread statistical machine learning algorithms employ pre-collected databases to calibrate parameters of the applied model. Due to the rapid development of communication networks, the size of the data sets has grown rapidly, making data mining methods essential. Such algorithms should store the extracted information in a compact and easily retrievable form. One of the most prevalent machine learning algorithms - Artificial Neural Networks (ANN) [2] - meets these requirements, as it has a compact form with a fast evaluation. However, the solution provided by its learning algorithm is only a local minima of the objective function, which may make the networks trained on the same database inconsistent. The ubiquitous Support Vector Machine (SVM) method [3], [4] leads to a quadratic programming task whose own global optima defines the compactness of the information retrieved. This automated selection can be beneficial since preliminary assumptions are not required, but it also makes the technique inapplicable in certain cases. A recently proposed approach, the Convex Networks (CN) method also has a globally optimal solution with commensurable performance, but the structure of the model employed gives the possibility of controlling the compactness of the result. Our aim is to define a special family of subset selection algorithms – inspired by Feature Selection [5] – which can effectively increase the sparsity of the CN solution without degrading its performance.

Now we will briefly outline the contents of the paper. First we overview the Convex Network (CN) method, which will lead to a constrained optimization formulation. In the following 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 some conclusions and ideas for future research.

II. BRIEF INTRODUCTION TO CONVEX NETWORKS

Tasks in machine learning often lead to classification and regression problems where models which employ 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 $\mathbf{x}_1, \ldots, \mathbf{x}_n$, where each point $\mathbf{x}_i$ belongs to one of 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 where $y_i \in \{-1, +1\}$, which is in many ways like the one-against-all method [6] or the output coding scheme [7]. Thus our investigation can be restricted to the problem of binary classification without any loss of generality.

The discriminative approach of the above problem utilizes a separator surface between classes, where the surface is defined by the following set for a fixed $\gamma \in \mathbb{R}$

$$\{\mathbf{z} \mid f(\mathbf{z}) = \gamma,\ \mathbf{z} \in \mathcal{X}\}, \qquad f : \mathcal{X} \to \mathbb{R}. \tag{1}$$

Now let us assume that the surface that separates points $\mathbf{x}_1, \ldots, \mathbf{x}_n$ optimally is searched for in the linear subspace of S, that is $f \in Span(S)$

$$\left\{ h : \mathcal{X} \to \mathbb{R} \mid h(\mathbf{x}) = \sum_{i=1}^{k} \alpha_i f_i(\mathbf{x}),\ \mathbf{x} \in \mathcal{X},\ \boldsymbol{\alpha} \in \mathbb{R}^k \right\}, \tag{2}$$

* The author was supported by the János Bolyai fellowship of the Hungarian Academy of Sciences.

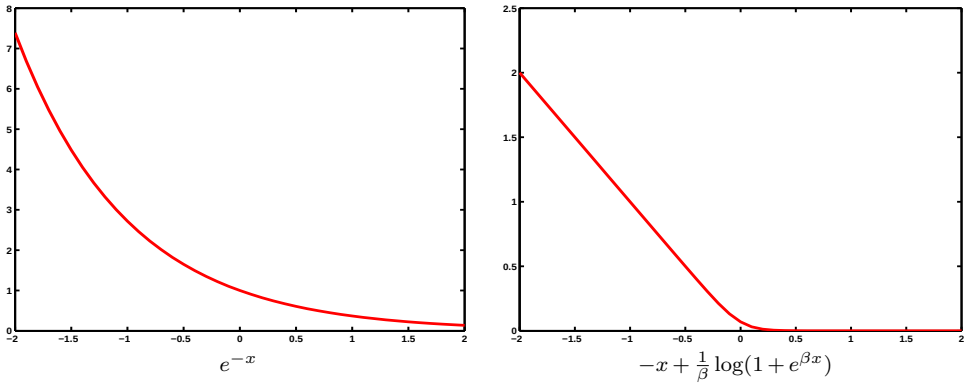

Fig. 1. Possible loss functions

where S denotes a finite set of continuous basis functions

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

The separation ability of a discriminative surface can be characterized based on the positions of the sample points: it is optimal when the points are as far from the surface as possible, and points with the same class label fall in the same half-space. The more optimal the position of a labelled point is, the larger the value $y_i f(\mathbf{x}_i)$ will be. Thus it can be applied to define a measure for the separation ability of a surface f over the sample $\mathbf{x}_1, \ldots, \mathbf{x}_n$. A possible formulation is

$$g(\boldsymbol{\alpha}) = \sum_{i=1}^{n} L\left(y_i \sum_{j=1}^{k} \alpha_j f_j(\mathbf{x}_i)\right), \tag{4}$$

where a twice continuously differentiable, monotone decreasing, lower bounded and convex loss-function $L : \mathbb{R} \to \mathbb{R}$ [**?**] was introduced. Of the many possibilities two candidates are shown in Fig. I.

The problem of approximating the parameter $\boldsymbol{\alpha}$ based on sparse sample data is ill-conditioned, and the classical way of solving it is to use regularization theory [8]. According to this theory, the optimal separator surface can be obtained by extending the separation measure of Eq. (4) with a regularization term, and searching for its minimum

$$\begin{array}{ll} \min_{\boldsymbol{\alpha}} & \tau(\boldsymbol{\alpha}) = \sum_{i=1}^{n} L\left(y_i \sum_{j=1}^{k} \alpha_j f_j(\mathbf{x}_i)\right) + \lambda \boldsymbol{\alpha}^T A \boldsymbol{\alpha} \\ \text{s.t.} & \boldsymbol{\alpha} \in \mathcal{A} = \mathcal{A}_1 \times \ldots \times \mathcal{A}_k \end{array} \tag{5}$$

where $\lambda > 0$ and $A \in \mathbb{R}^{k \times k}$ is an arbitrary symmetric positive-definite matrix. Here we restrict our investigation to the product space of non-empty intervals that also includes the unconstrained task with case $\mathcal{A}_i = (-\infty, \infty)$.

It can readily be seen that Eq. (5) is a convex programming task which can be solved by one of the many techniques [9]. The suggested method, the constrained Gauss-Seidel (GS) iteration technique, modifies one component of the solution at each step 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 projection mapping. The GS 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, every level set of the function should be bounded and there must exist a $\delta > 0$ such that $0 < \delta \leq \nabla^2_{ii}\tau(\mathbf{x}))$. The limit point of the iteration is the extreme of the function over $\mathcal{A}$ [9].

Definition 1 (constrained Gauss-Seidel iteration)

$$\alpha_i^{t+1} = \left[\alpha_i^t - \gamma \nabla_i \tau(\mathbf{z}_i^t)\right]_i^p \quad \gamma > 0$$

where

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

and

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

III. SPARSE SOLUTIONS

The separator surface coded by a CN problem takes the form

$$\{\mathbf{z} \mid \sum_{j=1}^{k} \alpha_j f_j(\mathbf{z}) = \gamma,\ \mathbf{z} \in \mathcal{X}\}, \qquad f : \mathcal{X} \to \mathbb{R}. \tag{6}$$

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 increase the sparsity by degrading the performance.

In order to control the complexity, the number of basis functions will be restricted by applying the following constraint in the CN task

$$\sum_{i=1}^{k} |sign(\alpha_i)| \leq q \tag{7}$$

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 only approach is to select a subset of order q from the available basis functions where the classification problem can be solved approximately as optimal as in the case of the complete one. This selection task is NP hard, so the employed heuristics can provide only suboptimal solutions by executing CN with different parameters several times.

Measure-based subset selection is an active area of other fields in artificial intelligence like Feature Selection [5], say. In that context one should select r features from the available m, so as to maximize the classification performance of a machine learning algorithm. The elaborated techniques can be employed for CN subset selection problem if the required measure is the optimal objective function value of the executed CN task. In the following let $CN(I)$ denote this optimal value of the objective function in CN when the basis function set is restricted in the following way:

$$S = \{f_{i_1}(\mathbf{x}), \ldots, f_{i_l}(\mathbf{x})\} \qquad i_1, \ldots, i_l \in I. \tag{8}$$

SFS The Sequential Forward Selection method is a greedy approach for the measure-based subset selection problem. Starting with the empty index set it extends the indices with the locally optimal element without backtracking.

```
SFS(q)
  Y = {1,...,k}; I = ∅;
  for i = 1...q
    t = argmin_{j∈Y−I} CN(I ∪ {j})
    I = I ∪ {t};
  return I;
```

PTA The SFS method is a sequential algorithm, hence previous steps cannot be modified when detecting their latter impact on the result. A solution to the problem is the Plus *l* Take Away *r* approach which periodically extends the actual index set by *l* elements and afterwards removes *r* ones. By doing this the effects of previous selections can be eliminated during the execution.

```
PTA(q, l, r)
  Y = {1,...,k};
  if (l > r) then
    i = 0; I = ∅; goto Step1;
  else
    i = k; I = Y; goto Step2;
 Step1:
  repeat l times
    t = argmin_{j∈Y−I} CN(I ∪ {j});
    I = I ∪ {t}; i = i + 1;
    if (i == q) goto Step3;
 Step2:
  repeat r times
    t = argmin_{j∈I} CN(I − {j})
    I = I − {t}; i = i - 1;
    if (i == q) goto Step3;
  goto Step1;
 Step3:  return I;
```

SFFS During PTA r removing steps always follow l extending ones. Hence it is possible to execute a removing step when the evolving set has a worse measure value than the previous one of the same order. Conversely, an extending step can be performed when it is better to remove a function at the that particular level. These problems are absent in the Sequential Forward Floating Selection algorithm. It removes elements after the extending step while the measure obtained is better than the previous ones of the same order.

```
SFFS(q)
  Y = {1,...,k};
  Y_0 = ∅; i = 0;
 Step1:
  t = argmin_{j∈Y−Y_i} CN(Y_i ∪ {j});
  Y_{i+1} = Y_i ∪ {t}; i = i + 1;
  if (i == q) goto Step3;
 Step2:
  t = argmin_{j∈Y_i} CN(Y_i − {j})
  if CN(Y_i − {t}) < CN(Y_{i−1}) then
    Y_{i−1} = Y_i − {t}; i = i - 1;
    goto Step2;
  else
    goto Step1;
 Step3:  return Y_q;
```

IV. RESULTS

We will now demonstrate the effectiveness of the CN approach by comparing its results with Artificial Neural Networks (ANN) and Support Vector Machines (SVM). In order to evaluate how well each algorithm classifies an unknown dataset, we performed a tenfold cross-validation test on publicly available datasets from the UCI repository [10].

We applied a feed-forward neural network (MLP) with one hidden layer, where the number of hidden neurons was set to three times the class number. The backpropagation learning rule was applied for training. MLP was trained five times on each dataset and then we chose the parameter values which gave the best performance on the training sample. For the SVM experiments, we employed our 1-norm SVM implementation where the bias term was absent [11] and multiclass cases were handled by the one-against-all approach. In addition, the cosine polynomial kernel we applied made the SVM method nonlinear with parameters $q = 3$ and $\sigma = 1$

$$\kappa(\mathbf{x}, \mathbf{y}) = \left(\frac{\mathbf{x}^T \mathbf{y}}{\|\mathbf{x}\| \, \|\mathbf{y}\|} + \sigma \right)^q . \qquad q \in \natural, \; \sigma \in \mathbb{R}_+ \tag{9}$$

The basis functions for the CN problem were defined with the aid of the above kernel function based on the points from the training set. There were

$$f_j(\mathbf{z}) = y_j \kappa(\mathbf{z}, \mathbf{x}_j) . \qquad j = 1, \ldots, n \tag{10}$$

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. (5) we set the identity matrix equal to A with $\lambda = 1$.

The correctness for the various methods obtained by tenfold testing are summarized in Table I. Besides the results produced by ANN and SVM at the settings described, the correctness of the CN method when sparsity was controlled by various heuristics is also shown. It confirms that the CN classification

Table I. Ten-fold cross-validation training and testing results on some UCI datasets. ANN is a feed-forward neural network with a hidden layer where the number of hidden units was set to three times the number of classes. SVM used the cosine polynomial kernel defined in Eq. (9) for nonlinearity. Using Eq. (10) the CN method applied the same basis functions. The sparsity was controlled by limiting the number of available basis functions at 10%, 20% and 30% to the complete sets, respectively.

SFS PTA SFFS	10%	20%	30%	100%	ANN	SVM
balance	94.66 94.75 **95.41**	94.77 95.19 94.97	94.91 94.92 95.08	**95.41**	86.35	90.63
bupa	69.46 69.20 70.37	69.22 70.66 71.20	68.96 71.41 69.88	71.92	68.07	**74.39**
glass	84.36 85.10 85.18	**88.62** 86.56 85.47	86.63 85.74 86.91	86.23	69.87	84.70
iono	89.24 91.51 92.13	91.81 92.25 **93.24**	90.71 92.71 92.48	92.41	86.17	91.09
monks	93.07 92.80 92.65	93.55 91.85 94.90	94.64 90.79 95.49	**96.51**	87.28	95.82
pima	77.86 **78.42** 77.74	76.43 77.14 76.49	75.97 76.55 78.03	74.82	76.09	75.58
wdbc	97.44 97.35 97.18	97.22 96.96 97.29	97.38 96.16 97.43	96.93	97.61	**97.62**
wpbc	76.26 78.06 75.82	75.92 78.19 76.39	74.75 77.03 **79.98**	79.63	76.41	77.36

method is indeed just as effective as the ubiquitous machine learning algorithms. In fact, their performances were even surpassed in many cases. What is more, even better classification performance could be achieved by restricting the solutions via introducing constraints on the parameters. In ANN terminology, accepting a locally optimal solution might be regarded as a restriction of this kind, as it prevents the method from overfitting the training data. A similar behavior can be observed when applying heuristics for controlling the sparsity in the CN method. The accuracies obtained from tenfold testing are shown in the table for cases when we limited the order of the selected subset of Eq. (10) to 10, 20 and 30% of the original one. The PTA algorithm was executed with settings $l = 3$ and $r = 1$. As can be seen, all the algorithms selected subsets that yielded reasonable test results. This kind of capacity reduction in the CN learning method induces a sort of regularization, as reflected in the results. Namely, reduced bases outperform the original ones in many cases. The various algorithms introduced here attained their best performance on different tasks. In general, different preferences during the learning phase will lead the user to choose different heuristics.

V. CONCLUSIONS

This paper gave an overview of the CN method, which is a reformulation of certain machine learning algorithms, including several well-known nonlinear classification methods. The proposed formula can be solved by the convergent nonlinear Gauss-Seidel iteration process. We also performed numerical tests, and the results indicate that it 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 feature selection based heuristics, as presented here. Future work includes a new heuristics 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 [12] for problems which do not fit in the memory.

REFERENCES

[1] K. Kovács and A. Kocsor, "Classification using a sparse combination of basis functions," *Acta Cybernetica*.

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

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

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

[5] P. Pudil, J. Novovicova, and J. Kittler, "Feature selection based on the approximation of class densities by finite mixtures of the special type," *Pattern Recognition*, vol. 28, no. 9, pp. 1389–1397, 1995.

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

[7] E. B. Kong and T. Dietterich, "Error-correcting output coding corrects bias and variance," in *Proc. of International Conference on Machine Learning*, 1995, pp. 313–321.

[8] A. N. Tikhonov and V. Y. Arsenin, Eds., *Solutions of Ill-posed Problems*, W. H. Winston, 1997.

[9] D.P. Bertsekas and J. N. Tsitsiklis, Eds., *Parallel and Distributed Computation: Numerical Methods*, Prentice Hall, 1989.

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

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

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