

Scoring Systems and Large Margin Perceptron Ranking

Bernd-Juergen Falkowski, University of Applied Sciences Stralsund, Zur Schwedenschanze 15, D-18435, Stralsund, Germany; E-mail: Bernd.Falkowski@fh-stralsund.de

Martin Appelt, University of Applied Sciences Stralsund, Zur Schwedenschanze 15, D-18435, Stralsund, Germany

Christian Finger, University of Applied Sciences Stralsund, Zur Schwedenschanze 15, D-18435, Stralsund, Germany

Sebastian Koch, University of Applied Sciences Stralsund, Zur Schwedenschanze 15, D-18435, Stralsund, Germany

Hendrik van der Linde, University of Applied Sciences Stralsund, Zur Schwedenschanze 15, D-18435, Stralsund, Germany

ABSTRACT

Perceptron learning is proposed in the context of so-called scoring systems used for assessing creditworthiness as stipulated in the Basel II central banks capital accord of the G10-states. The approximate solution of a related ranking problem using a large margin algorithm is described. Some experimental results obtained by utilizing a Java prototype are exhibited. From these it becomes apparent that combining the large margin algorithm presented here with the pocket algorithm provides an attractive alternative to the use of support vector machines. Related algorithms are briefly discussed.

1. INTRODUCTION

At least since the Basel II central banks capital accord of the G10-states, cf. e.g. [1], the individual objective rating of the creditworthiness of customers has become an important problem. To this end so-called scoring systems, cf. e.g. [12], [23], [17], [6] have been used for quite some time. Generally these systems are simple classifiers that are implemented as (linear) discriminants where customer characteristics such as income, property assets, liabilities and the likes are assigned points or grades and then a weighted average is computed, where a customer is judged “good” or “bad” according to whether the average exceeds a cut-off point or not. In an extreme case the attributes are just binary ones where 0 respectively 1 signifies that the property does not hold respectively holds. This situation frequently arises in practice. The weights can then either be computed using classical statistical methods or more recently employing artificial neural networks, cf. e.g. [19], provided that suitable bank records are available for training.

However, the use of only two classes for the classification of customers presents certain problems. The event of a credit default for example is not precisely defined, cf. [1], p. 92, so that banking records would almost certainly need at least one more class (e.g. “doubtful (?) customers”). This indicates that a finer distinction among customers could be useful. Indeed, after a computation of default probabilities (again usually based on two classes) banks divide customers into a larger number of classes. This, of course, seems rather counter-intuitive, since surely the division should (and could) be based on experience and be effected before probabilities are computed. Hence in this paper it is assumed that training data are available, where banking customers are divided into mutually disjoint risk classes C_1, C_2, \dots, C_k . Here class C_i is preferred to C_j if $i < j$. It was shown in [8] how this preference relation may be learned employing a generalized version of the pocket algorithm to solve the associated ranking problem. Unfortunately this is not a large margin algorithm, cf. e.g. [11], and hence in this paper a large margin perceptron ranking algorithm based on the work of Krauth and Mezard, cf. [14], will be presented.

Note that the use of several classes has been investigated beforehand, see e.g. [2], p. 237. Moreover, the use of ranking functions has been recognized in an information retrieval context, cf. e.g. [25], for solving certain financial problems, cf. [15], and for collaborative filtering, cf. [20], [21]. However, at least in the banking business, ranking functions, as described in section 2 below, see also [7], [22], apparently have not been used before for the rating of customers. Note also that the fixed margin ranking algorithm described in [20] and [21] involves

solving a quadratic and a linear programming problem successively whilst the algorithm presented below is obtained from a reduction of the ranking problem that allows a surprisingly simple albeit approximate solution using a modified perceptron learning algorithm.

2. REDUCTION OF THE RANKING PROBLEM

Suitable anonymous training data from a large German bank were available. In abstract terms then t vectors $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_t$ from \mathfrak{R}^n (think of these as having grades assigned to individual customer characteristics as their entries) together with their risk classification (i.e. their risk class C_s for $1 \leq s \leq k$, where the risk classes were assumed to constitute a partition of pattern space) were given. Hence implicitly a preference relation (partial order) “ $>$ ” in pattern space was determined for these vectors by

$$\mathbf{x}_i > \mathbf{x}_j \quad \text{if } \mathbf{x}_i \in C_i \quad \text{and} \quad \mathbf{x}_j \in C_j \quad \text{where } i < j.$$

It was then required to find a map $m_w: \mathfrak{R}^n \rightarrow \mathfrak{R}$ that preserves this preference relation, where the index w of course denotes a weight vector. More precisely one must have

$$\mathbf{x}_i > \mathbf{x}_j \quad \Rightarrow \quad m_w(\mathbf{x}_i) > m_w(\mathbf{x}_j)$$

If one now specializes by setting $m_w(\mathbf{x}) := \langle \varphi(\mathbf{x}), \mathbf{w} \rangle$, denoting the scalar product by $\langle \cdot, \cdot \rangle$ and an embedding of \mathbf{x} in a generally higher (m -) dimensional feature space by φ , then the problem reduces to finding a weight vector \mathbf{w} and constants (“cut-offs”) $c_1 > c_2 > \dots > c_{k-1}$ such that

$$\begin{aligned} \mathbf{x} \in C_1 & \text{ if } \langle \varphi(\mathbf{x}), \mathbf{w} \rangle > c_1 \\ \mathbf{x} \in C_s & \text{ if } c_{s-1} \geq \langle \varphi(\mathbf{x}), \mathbf{w} \rangle > c_s \\ \text{for } & s = 2, 3, \dots, k-1 \\ \mathbf{x} \in C_k & \text{ if } c_{k-1} \geq \langle \varphi(\mathbf{x}), \mathbf{w} \rangle. \end{aligned}$$

The problem may then be reduced further to a standard problem:

Let \mathbf{e}_i denote the i -th unit vector in \mathfrak{R}^{k-1} considered as row vector and construct a matrix \mathbf{B} of dimension $(m_1 + 2m_2 + k - 2) \times (m + k - 1)$, where $m_1 := |C_1 \cup C_k|$ (here $|S|$ denotes the cardinality of set S) and $m_2 := |C_2 \cup C_3 \cup \dots \cup C_{k-1}|$, as follows:

$$\mathbf{B} := \begin{bmatrix} \mathbf{R} \\ \mathbf{D} \end{bmatrix}, \quad \text{dimension } R = (k-2) \times (m+k-1), \quad \text{and the } i\text{-th row of } \mathbf{R} \text{ is given by the row vector } (0, \dots, 0, \mathbf{e}_i, -\mathbf{e}_{i+1}) \text{ with } m \text{ leading zeros. Moreover } \mathbf{D} \text{ is described by:}$$

For every vector \mathbf{x} in C_i respectively C_k \mathbf{D} contains a row vector $(\varphi(\mathbf{x}), -\mathbf{e}_i)$

respectively $(-\varphi(\mathbf{x}), \mathbf{e}_{k-1})$, whilst for every vector \mathbf{x} in C_s with $1 < s < k$ it contains the vectors $(\varphi(\mathbf{x}), -\mathbf{e}_s)$ and $(-\varphi(\mathbf{x}), \mathbf{e}_{s-1})$. The reduction of the problem to a system of inequalities is then proved by the following lemma.

Lemma 1: A weight vector \mathbf{w} and constants $c_1 > c_2 > \dots > c_{k-1}$ solving the ranking problem may (if they exist) be obtained by solving the standard system of inequalities $\mathbf{B}\mathbf{v} > \mathbf{0}$ where $\mathbf{v} := (\mathbf{w}, c_1, c_2, \dots, c_{k-1})^T$.

Proof (see also [7]): Computation.

Of course, it must be admitted that the existence of a suitable weight vector \mathbf{v} is by no means guaranteed. However, at least in theory, the map φ may be chosen such that the capacity of a suitable separating hyperplane is large enough for a solution to exist with high probability, cf. [4].

The price one has to pay for this increased separating capacity consists on the one hand of larger computation times. On the other hand, and perhaps more importantly, a loss of generalization capabilities due to a higher VC-dimension of the separating hyperplanes, cf. e.g. [24], must be taken into account. Hence it seemed advisable to employ fault tolerant perceptron learning using a generalized version of the pocket algorithm, cf. e.g. [11], [7]. In order to further improve the generalization properties here a large margin perceptron ranking algorithm based on the work of Krauth and Mezard will be presented. This may be used to construct a separating hyperplane that has the large margin property for the vectors correctly separated by the pocket algorithm. The reader should compare this to the large margin ranking described in [20]: There the problem is solved using a (soft margin) support vector machine. Unfortunately computation of the complete set of cut-offs requires the solution of an additional linear optimization problem.

3. LARGE MARGIN PERCEPTRON RANKING

Here the minimal distance of any vector to the closest cut-off will be maximized. The reader should compare this to the fixed margin strategy in the sense of [20].

3.1 Pseudo Code for Perceptron Ranking

First note that the reduction of the ranking problem in section 2 immediately leads to an elegant perceptron ranking algorithm (where separability is assumed).

The **pseudo code** for this algorithm reads as follows.

Perceptron Ranking Algorithm

Input: Binary vectors $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_l$ (or vectors with integer entries) from Z^n with corresponding classifications b_1, b_2, \dots, b_l from $\{1, 2, \dots, k\}$ (where the classes C_1, C_2, \dots, C_k for simplicity have been denoted by their indices) as training vectors, and a function $\varphi: Z^n \rightarrow Z^m$, where in general $m > n$.

Output: A weight vector \mathbf{w} and $k-1$ cut-offs $c_1 > c_2 > \dots > c_{k-1}$ as vector \mathbf{c} that solve the ranking problem.

Initialize \mathbf{w}, \mathbf{c} arbitrarily.

Cycle through the $t+k-2$ vectors $\mathbf{e}_1 - \mathbf{e}_2, \mathbf{e}_2 - \mathbf{e}_3, \dots, \mathbf{e}_{k-2} - \mathbf{e}_{k-1}, \varphi(\mathbf{x}_1), \varphi(\mathbf{x}_2), \dots, \varphi(\mathbf{x}_l)$ and do until no further erroneous classifications occur

If $\langle \mathbf{e}_p - \mathbf{e}_{p+1}, \mathbf{c} \rangle \leq 0$ then

$c_p := c_p + 1;$

$c_{p+1} := c_{p+1} - 1;$

End If

If $\mathbf{x}_p \in C_s$ & $\langle \varphi(\mathbf{x}_p), \mathbf{w} \rangle \leq c_s$ & $1 \leq s \leq k-1$ then

$\mathbf{w} := \mathbf{w} + \varphi(\mathbf{x}_p);$

$c_s := c_s - 1;$

End If

If $\mathbf{x}_p \in C_s$ & $\langle \varphi(\mathbf{x}_p), \mathbf{w} \rangle \geq c_{s-1}$ & $2 \leq s \leq k$ then

$\mathbf{w} := \mathbf{w} - \varphi(\mathbf{x}_p);$

$c_{s-1} := c_{s-1} + 1;$

End If

Return \mathbf{w}, \mathbf{c}

Remark: The restriction on the entries of the training vectors, which would be rather a nuisance for practical applications, can be removed fairly easily, cf. [18].

3.2 Correctness Proof for Perceptron Ranking

This follows immediately from the Perceptron Learning Theorem, cf. [16] and [3] by observing that its application to the ranking problem as presented in section 2 leads to the update operations given in the pseudo code above. Note here that for perceptron learning under the assumed separability the monotonicity of the cut-offs is guaranteed already if the inequalities resulting from the block matrix \mathbf{D} in section 2 are satisfied. Hence here the pseudo code could be shortened accordingly. However, if application of the pocket algorithm is envisaged, then the inequalities resulting from the block matrix \mathbf{R} in section 2 constitute rules that must be fulfilled and hence cannot be ignored if a small number of faults is considered admissible. The reader may wish to consult [22], for a similar ranking algorithm. It is, however, given using a kernel version and its Novikoff bound will be somewhat worse in general as can fairly easily be seen.

3.3 Pseudo Code for Large Margin Perceptron Ranking

The work of Krauth and Mezard concerning large margin perceptron learning is described in [14]. Certain modifications were necessary in order to combine it with 3.1 and obtain a large margin algorithm, cf. [9]

The **pseudo code** for this algorithm reads as follows.

Input: Binary vectors $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_l$ (or vectors with integer entries) from Z^n with corresponding classifications b_1, b_2, \dots, b_l from $\{1, 2, \dots, k\}$ (where the classes C_1, C_2, \dots, C_k for simplicity have been denoted by their indices) as training vectors, and a function $\varphi: Z^n \rightarrow Z^m$, where in general $m > n$. In addition a real number $\alpha > 0$ must be chosen.

Output: A weight vector \mathbf{w} and $k-1$ cut-offs $c_1 > c_2 > \dots > c_{k-1}$ as vector \mathbf{c} that approximate the maximal margin solution of the ranking problem. The approximation improves with increasing α .

Initialize \mathbf{w}, \mathbf{c} with $\mathbf{0}, \mathbf{0}$.

Loop

For the given vectors $\varphi(\mathbf{x}_1), \varphi(\mathbf{x}_2), \dots, \varphi(\mathbf{x}_l)$ compute the minimum of the following expressions:

(i) $\langle \varphi(\mathbf{x}_i), \mathbf{w} \rangle - c_s$ if $1 \leq s \leq k-1$

for $\mathbf{x}_i \in C_s, 1 \leq i \leq l$

(ii) $-\langle \varphi(\mathbf{x}_i), \mathbf{w} \rangle + c_{s-1}$ if $2 \leq s \leq k$

Then m either has the form

(a) $m = \langle \varphi(\mathbf{x}_j), \mathbf{w} \rangle - c_s$ for some j und $\mathbf{x}_j \in C_s$

or

(b) $m = -\langle \varphi(\mathbf{x}_k), \mathbf{w} \rangle + c_{s-1}$ for some k und $\mathbf{x}_k \in C_s$

If $m > \alpha$ then

display $\mathbf{w}, \mathbf{c};$

stop;

Else

If (a) then

$\mathbf{w} := \mathbf{w} + \varphi(\mathbf{x}_j);$

$c_s := c_s - 1;$

Else

$\mathbf{w} := \mathbf{w} - \varphi(\mathbf{x}_k);$

$c_{s-1} := c_{s-1} + 1;$

End If

End If

So in contrast to the ordinary perceptron ranking in the wide margin perceptron ranking the update operation is performed with the "worst" classified element as opposed to with an arbitrary misclassified element.

Note that for the case $\alpha = 0$ the original perceptron ranking as in 3.1 is obtained. Note also that a correctness proof of the algorithm follows from the correctness proof of the slightly modified Krauth/Mezard algorithm as given in [9] and the correctness proof in 3.2.

Perhaps it should also be pointed out that in analogy to ordinary perceptron learning kernel versions of both algorithms are readily deducible since in both cases only scalar products need to be computed to decide on the update operation.

3.4 Experimental Results

In order to test the large margin algorithm and with a view to further extensions a Java prototype was constructed. This was connected to an Access database via ODBC. In addition an Excel system with the Solver installed was employed for quadratic programming.

The experiments were carried out with 58 data vectors, which allowed perfect separation, provided by a German financial institution. The customers had been divided into 5 preference classes and the method by which the classes had been obtained was not disclosed (originally there were only 4 classes but six likely looking candidates were assigned to class 5 thus creating a slightly artificial situation). Each customer was characterized by 8 attributes where each attribute had been assigned a grade (from 1 to 5, where 1 is the best grade) based on evaluations by internal computer programs (again the details of this evaluation are not disclosed to outsiders). This led to an obvious reversal in some inequalities of the algorithm since the lowest weighted average grade was considered the best. The experiments were conducted on a standard laptop (1.47 GHz clock, 512 MB RAM). In order to test the quality of approximation measurements were conducted for various values of α (denoted by alpha in the Excel diagrams). For simplicity the function ϕ appearing in the algorithm was taken to be the identity. Moreover for comparison purposes the optimal large margin weights and cut-offs were calculated by solving the following quadratic programming problem employing the Excel Solver:

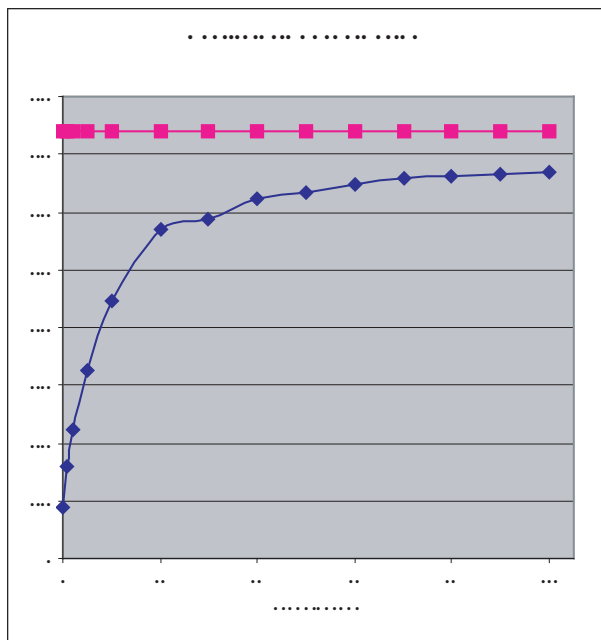
$$\begin{aligned} &\text{Minimize } \|\mathbf{w}\|^2 \quad \text{subject to} \\ &\langle \mathbf{x}_s, \mathbf{w} \rangle - c_{s-1} \geq 1 \text{ for } s = 2, 3, 4, 5 \\ &\langle \mathbf{x}_s, \mathbf{w} \rangle - c_s \leq -1 \text{ for } s = 1, 2, 3, 4 \end{aligned}$$

where $\mathbf{x}_i \in C_s$ and $i = 1, 2, 3, \dots, 58$.

In this programming problem the entries of the vector \mathbf{w} and the cut-offs were declared as variable to the Excel Solver so as to simultaneously get an optimal weight vector and optimal cut-offs

As a measure of the quality of approximation the distance of the “worst-classified” element to the nearest cut-off was computed. In diagram 1 the result for the optimal solution namely 0.0739745 is marked by a horizontal line. Note that the time measurements refer to elapsed time only and hence cannot be entirely ac-

Diagram 1



curate since for example cache effects have not been taken into account. However, for the purposes of the present paper this somewhat crude form of measurement was deemed adequate.

The results obtained were as follows in Diagrams 1 and 2.

As may be seen from diagram 1 the approximation to the optimal solution improves quite fast with increasing α up to about 80. Thereafter, however, only slow progress is made. Nevertheless, for practical purposes this approximation may be quite sufficient.

Clearly the time requirements increase linearly with increasing α as can be seen from diagram 2 (where times are given in milliseconds) and thus appear quite reasonable.

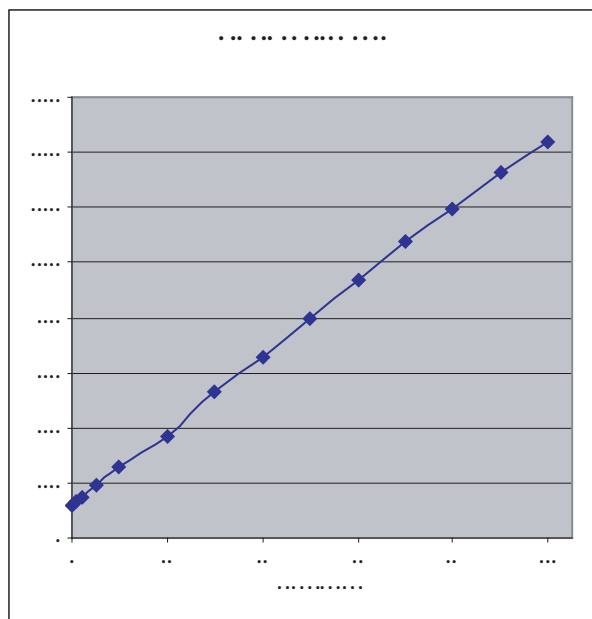
4. CONCLUSION AND OUTLOOK

A new large margin ranking algorithm has been presented. Encouraging experimental evidence has been obtained using “real life” data from a financial institution. The algorithm is based on a reduction of the ranking problem and a combination of the resulting ranking algorithm together with a result essentially due to Krauth and Mezard. In contrast to the wide margin ranking algorithm described in [20] it can be implemented with a surprisingly compact Java encoding. This is due to the fact that it can be seen as an extension of classical perceptron learning. On the other hand, of course, it gives only an approximate solution which may, however, as indicated by the experimental results, be quite satisfactory for practical applications. To clarify the situation additional experiments are needed and it is envisaged to perform these as soon as suitably large data sets become available.

In addition the algorithm only works for separable sets. However, it is intended to combine it with a modified version of the pocket algorithm by applying it to those data sets only that are correctly separated. This way an empirical risk minimization would be performed which is then followed by maximizing the margin. This seems attractive since that way certain approximations inherent to the soft margin support vector machine as utilized in [20] are avoided. Again it is intended to conduct suitable experiments as soon as possible.

Finally a few comments on related algorithms seem in order. The large margin algorithm in [20] has been briefly discussed already. The ranking algorithms in [5] and [13] appear inferior from the results given in [20]. In [26] large margin perceptron learning was introduced for the pocket algorithm. However, in spite of reasonable experimental evidence, the theoretical basis appears slightly shaky, for details see e.g. [10]. The ranking algorithm in [22] (soft margin version) appears to contain a gap since the monotonicity condition for the cut-offs seems to

Diagram 2



be neglected. Moreover an additional vector is ignored without explaining the consequences. In short then the algorithm closest to the one presented here seems to appear in [20]. Of course, it has been tested in a completely different context and an objective comparison concerning the banking application envisaged here is still outstanding.

REFERENCES

- [1] Banking Committee on Banking Supervision: International Convergence of Capital Measurements and Capital Standards, A Revised Framework, Bank for International Settlements, <http://www.bis.org/publ/bcbcsa.htm> (June 2004)
- [2] Bishop, C.M.: *Neural Networks for Pattern Recognition*. Oxford University Press, (1998)
- [3] Block, H.D.; Levin, S.A.: On the Boundedness of an Iterative Procedure for Solving a System of Linear Inequalities. *Proc. AMS*, (1970)
- [4] Cover, T.M.: Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition. *IEEE Trans. on Electronic Computers*, Vol. 14, (1965)
- [5] Crammer, K.; Singer, Y.: *Pranking with Ranking*, NIPS, (2001)
- [6] Episcopos, A.; Pericli, A.; Hu, J.: Commercial Mortgage Default: A Comparison of the Logistic Model with Artificial Neural Networks. *Proceedings of the Third International Conference on Neural Networks in the Capital Markets*, London, England, (1995)
- [7] Falkowski, B.-J.: *Lernender Klassifikator*, Offenlegungsschrift DE 101 14874 A1, Deutsches Patent- und Markenamt, München, (Learning Classifier, Patent Number DE 101 14874 A1, German Patent Office, Munich) (2002)
- [8] Falkowski, B.-J.: On a Ranking Problem Associated with Basel II. In: *Proceedings of the 5th International Conference on Hybrid Intelligent Systems (HIS 2005)*. Eds. N. Nedjah, L.M. Mourelle, M.M.B.R. Vellasco, A Abraham, M. Köppen, IEEE Computer Society, (2005)
- [9] Falkowski, B.-J.: A Note on a Large Margin Perceptron Algorithm, to appear in *Information Technology and Control*
- [10] Falkowski, B.-J.: On Large Margin Pocket Algorithms, to appear in *Proceedings of the IASTED International Conference on Computational Intelligence*, CI 2006
- [11] Gallant, S.I.: *Perceptron-based Learning Algorithms*. *IEEE Transactions on Neural Networks*, Vol. 1, No. 2, (1990)
- [12] Hand, D.J.; Henley, W.E.: Statistical Classification Methods in Consumer Credit Scoring: a Review. *Journal of the Royal Statistical Society, Series A*, 160, Part 3, (1997)
- [13] Herbrich, R.; Graepel, T.; Obermayer, K.: Large Margin Rank Boundaries for Ordinal Regression. In: *Advances in Large Margin Classifiers* (Eds. Smola, A.J.; Bartlett, P.; Schölkopf, B.; Schuurmans, D.), MIT Press, Neural Information Processing Series, (2000)
- [14] Krauth, W.; Mezard, M.: Learning Algorithms with Optimal Stability in Neural Networks. *J. Phys. A: Math. Gen.* 20, (1987)
- [15] Mathieson, M.: Ordinal Models for Neural Networks. *Neural Networks in Financial Engineering, Proceedings of the 3rd International Conference on Neural Networks in the Capital Markets*, World Scientific, (1996)
- [16] Minsky, M.L.; Papert, S.: *Perceptrons*. MIT Press, (Expanded Edition 1990)
- [17] Müller, M.; Härdle, W.: Exploring Credit Data. In: Bol, G.; Nakhneizadeh, G.; Racher, S.T.; Ridder, T.; Vollmer, K.-H. (Eds.): *Credit Risk-Measurement, Evaluation, and Management*, Physica-Verlag, (2003)
- [18] Muselli, M.: On Convergence Properties of Pocket Algorithm. *IEEE Trans. on Neural Networks*, 8 (3), 1997
- [19] Shadbolt, J.; Taylor, J.G. (Eds.): *Neural Networks and the Financial Markets*. Springer-Verlag, (2002)
- [20] Shashua, A.; Levin, A.: Taxonomy of Large Margin Principle Algorithms for Ordinal Regression Problems. Technical Report 2002-39, Leibniz Center for Research, School of Computer Science and Eng., the Hebrew University of Jerusalem, (2002)
- [21] Shashua, A.; Levin, A.: Ranking with Large Margin Principle: Two Approaches, NIPS 14, (2003)
- [22] Shawe-Taylor, J.; Cristianini, N.: *Kernel Methods for Pattern Analysis*. Cambridge University Press, (2004)
- [23] Thomas, L.C.: A Survey of Credit and Behavioural Scoring: Forecasting Financial Risk of Lending to Consumers. *International Journal of Forecasting*, 16, (2000)
- [24] Vapnik, V.N.: *Statistical Learning Theory*. John Wiley & Sons, (1998)
- [25] Wong, S.K.M.; Ziarko, W.; Wong, P.C.N.: Generalized Vector Space Model in Information Retrieval. *Proceedings of the 8th ACM SIGIR Conference on Research and Development in Information Retrieval*, USA, (1985)
- [26] Xu, J.; Zhang, X.; Li, Y.: Large Margin Kernel Pocket Algorithm. In: *Proceedings of the International Joint Conference on Neural Networks 2001, IJCNN'01*, Vol. 2. New York: IEEE, (2001)

0 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/proceeding-paper/scoring-systems-large-margin-perceptron/33213

Related Content

Covering Based Pessimistic Multigranular Approximate Rough Equalities and Their Properties

Balakrushna Tripathy and Radha Raman Mohanty (2018). *International Journal of Rough Sets and Data Analysis* (pp. 58-78).

www.irma-international.org/article/covering-based-pessimistic-multigranular-approximate-rough-equalities-and-their-properties/190891

A Comparative Study of Infomax, Extended Infomax and Multi-User Kurtosis Algorithms for Blind Source Separation

Monorama Swaim, Rutuparna Panda and Prithviraj Kabisatpathy (2019). *International Journal of Rough Sets and Data Analysis* (pp. 1-17).

www.irma-international.org/article/a-comparative-study-of-infomax-extended-infomax-and-multi-user-kurtosis-algorithms-for-blind-source-separation/219807

From Synergy to Symbiosis: New Directions in Security and Privacy?

Vasilios Katos, Frank Stowell and Peter Bednar (2009). *International Journal of Information Technologies and Systems Approach* (pp. 1-14).

www.irma-international.org/article/synergy-symbiosis-new-directions-security/4023

Organizational Characteristics and Their Influence on Information Security in Trinidad and Tobago

Kyle Papin-Ramcharan and Simon Fraser (2015). *Encyclopedia of Information Science and Technology, Third Edition* (pp. 4358-4372).

www.irma-international.org/chapter/organizational-characteristics-and-their-influence-on-information-security-in-trinidad-and-tobago/112878

Information Technology as a Service

Robin G. Qiu (2009). *Utilizing Information Technology Systems Across Disciplines: Advancements in the Application of Computer Science* (pp. 261-278).

www.irma-international.org/chapter/information-technology-service/30730