Babes-Bolyai University of Cluj-Napoca, Romania
Faculty of Mathematics and Computer Science
Department of Computer Science

Ruxandra Stoean (former Gorunescu)

Evolutionary Computation. Application in Data Analysis and Machine Learning

PhD Dissertation

Supervisor: D. Dumitrescu

Thesis commitee:

Prof. Dr. Thomas Bartz-Beielstein, University of Cologne, Germany
Prof. Dr. Henri Luchian, Al. I. Cuza University, Iasi, Romania
Prof. Dr. Doina Tatar, Babes-Bolyai University, Cluj-Napoca, Romania

February 2008

Dedication

To my husband, Catalin, for we have *evolved* together.

Acknowledgements

I would like to thank Professor D. Dumitrescu for all the constant guiding, help and understanding. His coordination goes further beyond teachings about the importance of the study on recent literature, the necessity of the argument for the development of a new technique or the principles for a rigorous writing of a paper. He taught me the courage and contentment to investigate new ideas, to explore unattended and to strongly put forward the results.

My next thanks go to my dear collaborator, Mike Preuss. His influence has been present above the work we have done together.

I am moreover grateful to Professor Peter Millard and Dr. Elia El-Darzi, whose counsel and assistance were priceless.

I would also like to express my gratitude to Professor Nicolae Tandareanu who directed me towards evolutionary computation and who has always kindly tried to make these doctoral years easier for me back at my home university.

I cannot enough thank Professors Hans-Paul Schwefel and Günther Rudolph for the immense opportunity they have provided me by agreeing to a research period at their Chair of Algorithm Engineering at the University of Dortmund. I am also grateful to Professor Thomas Bartz-Beielstein, now at the University of Koeln, for explaining the mechanisms of his automatic tuning method, which I used in my experiments, and for all the kind words of support. Lots of thanks go also to the great staff of the Chair of Algorithm Engineering at the University of Dortmund. I additionally want to mention Ingo Mierswa from the Chair of Artificial Intelligence, University of Dortmund, for the useful discussions in connection to support vector machines.

A lot of appreciation goes towards the staff of the Department of Computer Science at Babes-Bolyai University who have provided valuable aid and advice in the improvement of my work, during the exams and reports that I delivered within the doctoral training.

Finally, I am thankful to the National University Research Council (Consiliul National al

Cercetarii Stiintifice din Romania - CNCSIS) for the doctoral scholarship they supplied.

On a personal plane, I thank my husband for all his patience and back-up. Having a collaborator as well as a critic in the family was of great luck. I am also obliged to my parents and sister for the continuous encouragement.

Published Work Associated with Thesis

International journals

Journals indexed by ISI

1. **Ruxandra Stoean**, Mike Preuss, Catalin Stoean, Elia El-Darzi, D. Dumitrescu, An Evolutionary Resemblant to Support Vector Machines for Classification and Regression, Journal of the Operational Research Society (2006 Impact Factor: 0.597), Palgrave Macmillan, submitted, 2007.

2. Catalin Stoean, Mike Preuss, **Ruxandra Stoean**, D. Dumitrescu, Multimodal Optimization by means of a Topological Species Conservation Algorithm, IEEE Transactions on Evolutionary Computation (2006 Impact Factor: 3.77), IEEE Intelligence Computational Society, submitted, 2008.

3. **Ruxandra Stoean**, D. Dumitrescu, Mike Preuss, Catalin Stoean, Evolutionary Support Vector Machines for Classification with Multiple Outcomes, Journal of Universal Computer Science (2006 Impact Factor: 0.34), Springer-Verlag, in press, 2008.

4. Catalin Stoean, D. Dumitrescu, Mike Preuss, **Ruxandra Stoean**, Cooperative Coevolution as a Paradigm for Classification, Journal of Universal Computer Science (2006 Impact Factor: 0.34), Springer-Verlag, in press, 2008.

5. **Ruxandra Stoean**, Catalin Stoean, Mike Preuss, D. Dumitrescu, Forecasting Soybean Diseases from Symptoms by Means of Evolutionary Support Vector Machines, Phytologia Balcanica (indexed by ISI), Vol. 12, No. 3, pp. 345 - 350, Sofia, Bulgaria, 2006.

Journals indexed by Zentralblatt

6. Florin Gorunescu, Marina Gorunescu, **Ruxandra Gorunescu**, A metaheuristic GAs method as a decision support for the choice of cancer treatment, Siberian Journal of Numerical Mathematica (Siberian Branch of the Russian Academy of Science - Novosibirsk), Vol. 7, No. 4, pp. 301-307, 2004.

Journals indexed by INSPEC

7. **Ruxandra Gorunescu**, P.H. Millard, D. Dumitrescu, Evolutionary Placement Decisions of a Multidisciplinary Panel using Genetic Chromodynamics, Journal of Enterprise Information Management (indexed by INSPEC), Emerald Group Publishing, Vol. 21, No. 1, pp. 93-104, 2008.

National journals

Journals indexed by CNCSIS

8. **Ruxandra Stoean**, Catalin Stoean, Mike Preuss, D. Dumitrescu, Evolutionary Detection of Separating Hyperplanes in E-mail Classification, Acta Cibiniensis, Vol. LV Technical series, University ”Lucian Blaga” Sibiu Press, pp. 41-46, 2007.

9. **Ruxandra Stoean**, D. Dumitrescu, Catalin Stoean, Nonlinear Evolutionary Support Vector Machines. Application to Classification, Studia Babes-Bolyai, Seria Informatica, Vol. LI, No. 1, pp. 3-12, 2006.

10. **Ruxandra Stoean**, D. Dumitrescu, Linear Evolutionary Support Vector Machines for Separable Training Data, Annals of the University of Craiova, Mathematics and Computer Science Series, Vol. 33, pp. 141-146, ISSN: 1223-6934, 2006.

11. **Ruxandra Stoean**, D. Dumitrescu, Evolutionary Linear Separating Hyperplanes within Support Vector Machines, Scientific Bulletin, University of Pitesti, Mathematics and Computer Science Series, Issue 11, pp. 75-84, 2005.

12. Catalin Stoean, **Ruxandra Gorunescu**, Mike Preuss, D. Dumitrescu, An Evolutionary Learning Classifier System Applied to Text Categorization, Annals of West University of Timisoara, Mathematics and Computer Science Series, vol. XLII, special issue 1, pp. 265-278, 2004.

13. D. Dumitrescu, **Ruxandra Gorunescu**, Evolutionary clustering using adaptive prototypes, Studia Univ. Babes - Bolyai, Informatica, Volume XL IX, Number 1, pp. 15 - 20, 2004.

14. **Ruxandra Gorunescu**, D. Dumitrescu Evolutionary clustering using an incremental technique, Studia Univ. Babes - Bolyai, Informatica, Volume XL VIII, Number 2, pp. 25 - 33, 2003.

International conferences

Conferences indexed by IEEE

15. **Ruxandra Stoean**, Mike Preuss, Catalin Stoean, D. Dumitrescu, Concerning the Potential of Evolutionary Support Vector Machines, The IEEE Congress on Evolutionary Computation (CEC 2007), Singapore, pp. 1436 - 1443, 2007.

16. Catalin Stoean, Mike Preuss, D. Dumitrescu, **Ruxandra Stoean**, Cooperative Evolution of Rules for Classification, IEEE Postproceedings SYNASC 2006, IEEE Press, Lisa O'Conner (Ed.), Los Alamitos, CA, USA, ISBN 0-7695-2740-X, pp. 317-322, 2006.

17. **Ruxandra Stoean**, Mike Preuss, D. Dumitrescu, Catalin Stoean, Evolutionary Support Vector Regression Machines, IEEE Postproceedings SYNASC 2006, IEEE Press, Lisa O'Conner (Ed.), Los Alamitos, CA, USA, ISBN 0-7695-2740-X, pp. 330-335, 2006.

18. **Ruxandra Stoean**, Catalin Stoean, Mike Preuss, D. Dumitrescu, Evolutionary Support Vector Machines for Spam Filtering, RoEduNet IEEE International Conference, Sibiu, Romania, pp. 261-266, 2006.

19. **Ruxandra Stoean**, Catalin Stoean, Mike Preuss, Elia El-Darzi, D. Dumitrescu, Evolutionary Support Vector Machines for Diabetes Mellitus Diagnosis, Proceedings 3rd Inter-

national IEEE Conference on Intelligent Systems - IS 2006, University of Westminster, London, pp. 182-187, ISBN 1-4244-0196-8, 2006.

20. Catalin Stoean, Mike Preuss, **Ruxandra Gorunescu**, D. Dumitrescu, Elitist Generational Genetic Chromodynamics - a New Radii-Based Evolutionary Algorithm for Multimodal Optimization, The 2005 IEEE Congress on Evolutionary Computation - CEC 2005, Edinburgh, UK, Vol. 2, pp. 1839 - 1846, ISBN 0-7803-9363-5, 2005.

Conferences indexed by ACM

21. Catalin Stoean, Mike Preuss, **Ruxandra Stoean**, D. Dumitrescu, Disburdening the Species Conservation Evolutionary Algorithm of Arguing with Radii, The ACM Genetic and Evolutionary Computation Conference (GECCO 2007), London, UK, pp. 1420 - 1427, 2007.

22. Catalin Stoean, **Ruxandra Stoean**, Elia El-Darzi, Breast Cancer Diagnosis by Means of Cooperative Coevolution, Proceedings of the Third ACM International Conference on Intelligent Computing and Information Systems (ICICIS 2007), Police Press, Cairo, pp. 493-497, ISBN 977-237-172-3, 2007.

ISI proceedings of the conferences above

(a) **Ruxandra Stoean**, Mike Preuss, Catalin Stoean, D. Dumitrescu, Concerning the Potential of Evolutionary Support Vector Machines, The IEEE Congress on Evolutionary Computation (CEC 2007), Singapore, pp. 1436 - 1443, 2007.

(b) Catalin Stoean, Mike Preuss, **Ruxandra Stoean**, D. Dumitrescu, Disburdening the Species Conservation Evolutionary Algorithm of Arguing with Radii, The ACM Genetic and Evolutionary Computation Conference (GECCO 2007), London, UK, pp. 1420 - 1427, 2007.

(c) Catalin Stoean, Mike Preuss, **Ruxandra Gorunescu**, D. Dumitrescu, Elitist Generational Genetic Chromodynamics - a New Radii-Based Evolutionary Algorithm for Multimodal Optimization, The 2005 IEEE Congress on Evolutionary Computation - CEC 2005, Edinburgh, UK, Vol. 2, pp. 1839 - 1846, ISBN 0-7803-9363-5, 2005.

Other international conferences

23. Catalin Stoean, **Ruxandra Stoean**, Mike Preuss, D. Dumitrescu, Competitive Coevolution for Classification, Proceedings of the 7th International Conference on Artificial Intelligence and Digital Communications (AIDC 2007), pp. 28-39, 2007.

24. D. Dumitrescu, Catalin Stoean, **Ruxandra Stoean**, Genetic Chromodynamics for the Job Shop Scheduling Problem, International Conference Knowledge Engineering Principles and Techniques (KEPT 2007), Studia Univ. Babes - Bolyai, Informatica, Special Issue, pp. 153-160, 2007.

25. Catalin Stoean, **Ruxandra Stoean**, Elia El-Darzi, Breast Cancer Diagnosis by Means of Cooperative Coevolution (2), Proceedings First International Conference on Medical Informatics (ICMI2007), Misr University for Science and Technology, March 19th, Cairo, Egypt, pp. 19-24, 2007.

26. Catalin Stoean, **Ruxandra Stoean**, Elia El-Darzi, Breast Cancer Diagnosis by Means of Cooperative Coevolution (3), Proceedings Workshop "Medical Informatics (in frame of Third ACM International Conference on Intelligent Computing and Information Systems ICICIS2007), pp. 33-38, 2007.

27. Catalin Stoean, D. Dumitrescu, Mike Preuss, **Ruxandra Stoean**, Cooperative Coevolution for Classification, Bio-Inspired Computing: Theory and Applications, BIC-TA 2006, China, D. Dumitrescu, Linqing Pan (Eds.), pp. 289 - 298, 2006.

28. **Ruxandra Stoean**, D. Dumitrescu, Mike Preuss, Catalin Stoean, Different Techniques of Multi-class Evolutionary Support Vector Machines, Bio-Inspired Computing: Theory and Applications, BIC-TA 2006, China, D. Dumitrescu, Linqing Pan (Eds.), pp. 299-306, 2006.

29. Catalin Stoean, Mike Preuss, D. Dumitrescu, **Ruxandra Stoean**, A Cooperative Coevolutionary Algorithm for Multi-class Classification, 8th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing - SYNASC 2006, pp. 7-14, 2006.

30. **Ruxandra Stoean**, Mike Preuss, D. Dumitrescu, Catalin Stoean, Epsilon - Evolutionary Support Vector Regression, 8th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing - SYNASC 2006, pp. 21-27, 2006.

31. Catalin Stoean, **Ruxandra Stoean**, Mike Preuss, D. Dumitrescu, Spam Filtering by Means of Cooperative Coevolution, 4th European Conference on Intelligent Systems and Technologies, ECIT 2006, Iasi, Romania, Advances in Intelligent Systems and Technologies - Selected Papers, H. N. Teodorescu (Ed.), Performantica Press, pp. 157-159, ISBN 973-730-246-X , 2006.

32. **Ruxandra Stoean**, An Evolutionary Support Vector Machines Approach to Regression, 6th International Conference on Artificial Intelligence and Digital Communications - AIDC 2006, Thessaloniki, Greece, Research Notes in Artificial Intelligence and Data Communications, N. Tandareanu (Ed.), Reprograph Press, pp. 54-61, ISBN 973-742-413-1, 2006.

33. Catalin Stoean, **Ruxandra Stoean**, Mike Preuss, D. Dumitrescu, A Cooperative Evolutionary Algorithm for Classification, International Conference on Computers and Communications - ICCC 2006, Baile Felix Spa - Oradea, Romania, pp. 417-422, ISSN 1841-9836, 2006.

34. **Ruxandra Stoean**, Catalin Stoean, Mike Preuss, D. Dumitrescu, Evolutionary Multi-class Support Vector Machines for Classification, International Conference on Computers and Communications - ICCC 2006, Baile Felix Spa - Oradea, Romania, pp. 423-428, ISSN 1841-9836, 2006.

35. Catalin Stoean, **Ruxandra Stoean**, Mike Preuss, D. Dumitrescu, Diabetes Diagnosis through the Means of a Multimodal Evolutionary Algorithm, Proceedings of the First East European Conference on Health Care Modelling and Computation - HCMC 2005, Craiova, Romania, pp. 277-289, ISBN 973-7757-67-X, 2005.

36. Catalin Stoean, **Ruxandra Gorunescu**, D. Dumitrescu, A New Evolutionary Model for the Optimization of Multimodal Functions, The Anniversary Symposium Celebrating 25 Years of the Seminar Grigore Moisil and 15 Years of the Romanian Society for Fuzzy Systems and A.I., International participation and committee, 2005, Iasi, Romania, Intelligent Systems, Selected Papers, H. N. Teodorescu, J. Watada, J. Gil Aluja, M. Mihaila (Eds.), Performantica Press, pp. 65 - 72, ISBN 973-730-070-X, 2005.

37. Catalin Stoean, **Ruxandra Gorunescu**, Mike Preuss, D. Dumitrescu, An Evolutionary Learning Spam Filter System, SYNASC 2004, Timisoara, Romania, 8th International Sym-

posium on Symbolic and Numeric Algorithms for Scientific Computing, D. Petcu, V. Negru, D. Zaharie, T. Jebelean (Eds.), Mirton Press, pp. 512-522, ISBN 973-661-441-7, 2004.

38. Catalin Stoean, **Ruxandra Gorunescu**, Mike Preuss, D. Dumitrescu, Evolutionary Discovery of Adaptive Rules for Spam Detection, Fourth International Conference on Applied Mathematics - ICAM4, North University of Baia Mare, Romania, Abstract Proceedings, pp. 34, 2004.

39. Catalin Stoean, **Ruxandra Gorunescu**, Mike Preuss, D. Dumitrescu, Evolutionary Detection of Rules for Text Categorization. Application to Spam Filtering, Third European Conference on Intelligent Systems and Technologies-ECIT'2004, July 21-23, 2004, Iasi, Romania, Inteligent Systems, Selected papers, H. N. Teodorescu (Ed.), Performantica Press, pp. 87-95, ISBN 973-7994-85-X, 2004.

40. **Ruxandra Gorunescu**, P. H. Millard, An Evolutionary Model of a Multidisciplinary Review Panel for Admission to Long-term Care, Proceedings of International Conference on Computers and Communications - ICCC 2004, Baile - Felix, Oradea, pp. 181 - 185, 2004.

41. **Ruxandra Gorunescu**, D. Dumitrescu, An Evolutionary Approach to Fuzzy Clustering, 4th International Conference on Artificial Intelligence and Digital Communications, Craiova, Romania, June 2004, Research Notes in Artificial Intelligence and Data Communications, 4, N. Tandareanu (Ed.), Reprograph Press, pp. 267-274, 2004.

42. **Ruxandra Gorunescu**, Evolutionary Incremental Clustering. A New Technique for Detecting Natural Grouping, 3rd International Conference on Artificial Intelligence and Digital Communications, Craiova, Romania, June 2003, Research Notes in Artificial Intelligence and Data Communications, N. Tandareanu (Ed.), Reprograph Press, pp. 73 - 81, 3, 2003.

43. D. Dumitrescu, **Ruxandra Gorunescu**, Adaptive prototypes in evolutionary clustering, 3rd International Conference on Artificial Intelligence and Digital Communications, Craiova, Romania, June 2003, Research Notes in Artificial Intelligence and Data Communications, N. Tandareanu (Ed.), Reprograph Press, pp. 48 - 55, 3, 2003.

National conferences

44. **Ruxandra Stoean**, D. Dumitrescu, Evolutionary Support Vector Machines - a New Learning Paradigm. The Linear Non-separable Case, Proceedings of the Colocviul Academic Clujean de Informatica, Babes-Bolyai University of Cluj-Napoca, Faculty of Mathematics and Computer Science, pp. 15-20, 2005.

45. D. Dumitrescu, **Ruxandra Gorunescu**, Evolutionary Adaptive Fuzzy Clustering, Proceedings of the Zilele Academice Clujene Symposium, May 25, 2004, Babes-Bolyai University of Cluj-Napoca, pp. 61-66, 2004.

Technical reports

International reports

46. Catalin Stoean, **Ruxandra Stoean**, Mike Preuss, D. Dumitrescu, Coevolution for Classification, Technical Report Nr. CI-239/08, Collaborative Research Center on Computational Intelligence, University of Dortmund, 2008.

47. **Ruxandra Stoean**, Mike Preuss, Catalin Stoean and D. Dumitrescu, Evolutionary Support Vector Machines and their Application for Classification, Technical Report Nr. CI-212/06, Collaborative Research Center on Computational Intelligence, University of Dortmund, 2006.

National reports

48. **Ruxandra Stoean**, D. Dumitrescu, Evolutionary Support Vector Machines - a New Hybridized Learning Technique. Application to Classification, Technical Report, Department of Computer Science, Faculty of Mathematics and Computer Science, Babes-Bolyai University, 2005, http://www.cir.cs.ubbcluj.ro.

Contents

List of Figures

List of Tables

Abstract

The thesis presents a novel evolutionary technique constructed as a resemblant alternative of the standard support vector machines paradigm. The approach adopts the learning strategy of the latter but aims to simplify and generalize its training, by offering a transparent evolutionary substitute to the initial black-box. Contrary to the canonical technique, the evolutionary resembling support vector machines can at all times explicitly acquire the coefficients of the decision function, without any further constraints. Moreover, in order to converge, the evolutionary approach does not require properties of positive (semi-)definition for kernels within nonlinear learning.

The concept is furthermore extended to handle large amounts of data, a problem frequently occurring e.g. in spam mail detection, one of the considered test cases. An adapted chunking technique is therefore alternatively used. In addition to the two different representations, a crowding variant of the evolutionary algorithm is tested in order to investigate whether the performance of the method is preserved; its global search capabilities would be important for the prospected coevolution of non-standard kernels. An *all-in-one* evolutionary framework that also evolves the support vector machine hyperparameters is eventually proposed; the result, of valuable practical importance, confirms its employment.

Several potential structures, enhancements and additions are hence proposed and experiments on various configurations of real-world problems prove the validity of the new approach in terms of flexibility, prediction accuracy and runtime.

Chapter 1

Introduction

1.1 Problem Statement. Motivation and Aims

Support vector machines are a modern and powerful machine learning technique that has achieved competitive results in targeting data mining tasks, such as classification and regression. Despite the originality and performance of the learning vision of support vector machines, the inner training engine is intricate, constrained, rarely transparent and unable to offer convergence for any decision function. This has offered the motivation to investigate a resembling alternative training, based on evolutionary algorithms, which are known to be adaptable and robust [Dumitrescu, 2000], [Dumitrescu et al., 2000], [Eiben and Smith, 2003].

There have been numerous and recent other liaisons between support vector machines and evolutionary algorithms [Eads et al., 2002], [de Souza et al., 2005], [Friedrichs and Igel, 2004], however, the approach proposed herein is significantly different. Within the evolutionary resemblant to support vector machines, the learning path remains unchanged, but the coefficients of the decision function can now be evolved with respect to the optimization objectives of accurateness and generalization. The reason is to overcome the initial drawbacks from a remote perspective upon an otherwise effective training. Apart from the theoretical reasons, evolutionary resembling support vector machines offer a straightforward and efficient tool for practical applications [Stoean et al., 2006d], [Stoean et al., 2007a], [Stoean et al., 2007b].

1.2 Contributions

The original aspects of this dissertation can be summarized as follows:

- The evolutionary resembling technique considers the learning task as in support vector machines but uses an evolutionary algorithm to solve the optimization problem of determining the decision function.
- Classification and regression particularities are treated separately. The optimization problem is tackled through two possible evolutionary algorithms; one allows for a more relaxed evolutionary learning condition [Stoean et al., 2008b], while the second is more similar to support vector training [Stoean et al., 2007a]. Validation is achieved by considering five diverse real-world learning tasks [Stoean et al., 2008b], [Stoean et al., 2006h], [Stoean et al., 2006i], [Stoean et al., 2006d], [Stoean et al., 2007c].
- Besides comparing results, the potential of the utilized, simplistic evolutionary algorithm through parameterization is investigated [Stoean et al., 2007a], [Stoean et al., 2007b].
- To enable handling large data sets, the first approach is enhanced by the use of a chunking technique, resulting in a more versatile technique [Stoean et al., 2007a].
- The behavior of a crowding-based evolutionary algorithm on preserving the performance of the technique is examined with the purpose of its future employment for the coevolution of nonstandard kernels [Stoean et al., 2007a].
- The second methodology, which is more straightforward, is generalized through the additional evolution of internal hyperparameters within support vector machines; a very general method of practical importance is therefore achieved [Stoean et al., 2007b].
- The evolutionary approach demonstrates to be an efficient tool for real-world application in vital domains like disease diagnosis and prevention or spam filtering. Accordingly, various practical tasks are addressed and solved by proposed technique: Diabetes mellitus diagnosis [Stoean et al., 2006i], spam detection [Stoean et al., 2006g], [Stoean et al., 2007c], iris recognition [Stoean et al., 2008b], soybean disease diagnosis [Stoean et al., 2006h] and Boston housing [Stoean et al., 2006c], [Stoean et al., 2006d].

Obtained results have proven the suitability and efficiency of the proposed technique, so, in this context, the evolutionary resembling learning approach qualifies as a viable simpler alternative to the classical support vector machines. Nevertheless, the multicriterial optimization task can be further pursued through a specific evolutionary mechanism. More important, as a large potential of the evolutionary resemblant lies in the possible immediate implementation of unconstrained kernels, a method for their evolutionary determination must be appointed.

The thesis contributes some key elements to both evolutionary algorithms and support vector machines:

- The evolutionary resembling support vector machines combine the strong characteristics of the two important artificial intelligence fields, namely: The original learning concept of support vector machines and the flexibility of the direct search and optimization power of evolutionary algorithms.
- The novel alternative approach resolves the complexity of the support vector training.
- The proposed resemblant offers the possibility of a general evolutionary solution to all support vector machine components.
- The evolutionary resembling support vector machines open the direction towards the evolution and employment of nonstandard kernels.

1.3 Thesis Organization

The work is organized as follows.

Chapter 2 outlines the general working scheme of an evolutionary algorithm. The main elements and mechanisms are explained and exemplified.

Chapter 3 introduces the basic aspects of support vector machines and describes the geometrical learning idea within. The particular challenges and solutions to classification and regression tasks are presented in turn.

Chapter 4 brings along the principles and methods of the standard training manner in determining the optimal decision surface inside support vector machines. The conditions, constraints and solving steps are illustrated in detail. The demonstrations to theorems and propositions that

are not general but pertain to particularities of support vector machines have been adapted and are illustrated.

Chapters 5 and 6 present the evolutionary resembling technique. The two potential encodings of the training problem are reported and analyzed. The multiple improvements for a realistic perspective are proposed and tested. The novel paradigm is validated on various real-world tasks both for generality and practical reasons. Critical assessment and comparison to the rival canonical support vector machines is additionally undertaken.

Chapter 7 shows the real-world problems that had been targeted and solved by means of the evolutionary resembling support vector machines. Obtained results are revealed in comparison to those of the classical paradigm and of other successful artificial intelligent techniques.

Chapter 2

Overview of Evolutionary Algorithms

2.1 Aims of This Chapter

This chapter puts forward the basic aspects involved in the natural paradigm of evolutionary computation (EC) . The biological inspiration, elements and general scheme of an evolutionary algorithm (EA) are described. The role of each EA component is explained and the most common choices for every element are outlined. The eventual purpose of this chapter is to emphasize a general evolutionary framework that can be established as a starting point towards a practical solver for a considered problem.

2.2 Underlying Concepts

The pursuit of a solution to a problem could be imagined as a search held in the space of all potential solutions. If interest also lies in acquiring the best solution, search is doubled by an optimization process. During the last decades, the development of search and optimization techniques has changed course from the classical approaches, with restrictions, convergence issues and single point movement, to some flexible classes of methods based on principles of evolution and hereditary, the EAs [Baeck, 1996], [Baeck et al., 1997], [Dumitrescu, 2000], [Dumitrescu et al., 2000], [Eiben and Smith, 2003], [Fogel, 1995], [Michalewicz, 1992], [Schwefel et al., 2003], [Sarker et al., 2002].

The methods of EC are simple, general and fast, with several potential solutions that exist at the same time. EAs are semi-probabilistic techniques that combine local search - the exploitation

of the best available solutions at some point - with global search - the exploration of the search space. The different properties of continuity, convexity or derivability that have been standardly required for an objective function are of no further concern within EAs. As for success, an EA obviously cannot outperform a problem tailored method but gains ground as concerns generalization. Conversely, an EC technique is usually better than a random search strategy.

The EA idea finds its inspiration in what it is that governs nature. A population of initial individuals (genomes) goes through a process of adaptation through selection, recombination and mutation; these phenomena encourage the appearance of fitter solutions. The best performing individuals are selected in a probabilistic manner as parents of a new generation and gradually the system evolves to the optimum. The fittest individual(s) obtained after a certain number of iterations is (are) the solution to the problem.

2.3 Components of an Evolutionary Algorithm

A canonical EA is defined by several main constituents [Dumitrescu et al., 2000], [Dumitrescu, 2000], [Eiben and Smith, 2003]:

- a representation , an encoding for the candidate solutions (individuals, chromosomes)
- a fitness (evaluation, objective) function to measure the performance, quality of individuals
- a population of potential solutions and a method for generating its initial configuration
- a mechanism for parent selection
- a set of variation operators to create new individuals
- a means for survivor selection (replacement)
- a stop condition
- values for parameters: Population size, number of generations/fitness evaluations, recombination and mutation probabilities, mutation strength.

2.4 Design of an Evolutionary Algorithm

The standard behavior of an EA is outlined in Algorithm 1 [Dumitrescu, 2000], [Dumitrescu et al., 2000], [Eiben and Smith, 2003] .

Algorithm 1 A canonical evolutionary algorithm

Require: A search/optimization problem

Ensure: The fittest individual(s)

```
begin
t ← 0
initialize population P(t)
evaluate P(t)
while stop condition not reached do
    t ← t + 1
    select parents in P(t) from P(t-1)
    recombination on P(t)
    mutation on P(t)
    select survivors in P(t)
    evaluate P(t)
end while
return fittest individual(s)
end
```

2.4.1 Representation of Individuals

Although EAs were originally imagined and designed to work with a binary string encoding of potential solutions, their impact with real-world applications triggered the incorporation of problem knowledge into the structure of an individual . Thus, genes can refer integers, real numbers, ordinal values, strings, trees etc. It is common fashion that all individuals share the same length.

2.4.2 Fitness Function

The fitness function plays the role of the environment to which the individuals must adapt. It integrates the objective function of the specific problem at hand and, depending on the case, can or cannot suffer certain alterations.

2.4.3 Population and Initialization

A population of individuals intervenes in every iteration. Its cardinal is usually constant, but may decrease/increase in special types of EAs. The initial configuration is usually appointed of individuals whose every gene is generated in a pseudo-random manner with respect to the corresponding domain of definition .

2.4.4 Parent Selection

The goal of parent selection is to offer, at every generation, more chances of reproduction to the fittest individuals in the population. Its role is therefore that of exploitation of the best solutions that have been obtained up to the current time. A parents pool is consequently formed through a certain preferred and suitable mechanism. In what follows, we will refer to the most widely employed [Dumitrescu, 2000], [Dumitrescu et al., 2000], [Eiben and Smith, 2003].

The usual choice is fitness proportional selection where every individual is assigned a selection probability, which is taken as a measure of its absolute performance as compared to the absolute fitness values of the other individuals in the population; the higher one's fitness, the greater the chance to be chosen. Although straightforward, there are nevertheless two drawbacks to this type of parent selection: the dominance of a single fit individual (premature convergence) or the quasi-random selection when there are only small differences between the fitness values of individuals in a population (lack of selection pressure).

A more advantageous alternative way of selecting parents can be performed by ranking selection. Every individual in the current population is evaluated and the population is ordered decreasingly according to these values. Subsequently, a selection probability that corresponds to the rank in this order and to the value of the pressure of selection is attributed to each individual. The pressure of selection is defined as the mean number of offspring of the fittest individual and is a constant of this selection mechanism, usually considered as a number from the [1, 2] interval.

The fact that this approach relates on relative fitness makes it overcome the previously discussed disadvantages.

The selection probabilities are typically implemented by the roulette wheel mechanism. The cumulative probabilities are calculated and a roulette is built accordingly, where each slot reflects the performance of an individual. Each time a spin takes place, one corresponding individual is selected for the parents pool.

A third and perhaps the most simple and practical selection method is tournament selection, which is also based on a relative fitness comparison. For the number of times required to fill the parents pool, two or more individuals are randomly chosen and the best one of them, based on their evaluations, is selected for reproduction.

2.4.5 Variation

Selection alone cannot introduce variation into the population. This task relies on the recombination and mutation operators .

Recombination

Recombination aims to achieve the exploration of the promising regions of the search space. In this respect, it combines the genetic material of some (usually two) considered individuals. The resulting offspring (generally one or two) have thus characteristics of both parents. The two parents that undergo recombination are chosen probabilistically, according to the value that is appointed to the corresponding parameter of the EA.

The process usually takes place as follows. For every individual in the current population, a random number in the [0, 1] interval is generated; if it is smaller than the given recombination probability , then the current individual is chosen for recombination. If the number of chosen individuals is odd, then they recombine as pairs which are randomly formed. Otherwise, one individual is deleted or another one is added from the parents pool, a decision which is also randomly taken.

As concerns the different possibilities of recombination, several situations can be distinguished: Discrete, continuous or problem dependent mechanisms; the choice obviously depends on the specific task and chosen encoding. The types are numerous, however only the most common schemes within the more general discrete and continuous representations are further outlined

[Dumitrescu, 2000], [Dumitrescu et al., 2000], [Eiben and Smith, 2003].

For the discrete representation, the best known type is the one point recombination. A random position is generated which is the point of split for the two parents. The resulting offspring take the first part from one parent and the other from the second, respectively. A generalized form is the multiple point recombination, where several cut points are considered. Furthermore, within adaptive recombination, the splitting points also undergo evolution by adapting to previous splits that took place. Additionally, segmented recombination is a variant of the multiple point recombination where the number of points can vary from one individual to another.

Uniform recombination does not use split points. For every gene of the first offspring it is probabilistically decided which parent gives the value of that component, while the corresponding gene of the second offspring gets the value of that of the other parent. This operator could also be considered such that the values for both offspring individuals are computed in the same probabilistic manner, but independently.

Shuffle recombination is an add-on to an arbitrary discrete recombination scheme. The genes of the two parents are shuffled randomly, remembering their initial positions. The resulting individuals may then undergo any kind of discrete recombination. Resulting offspring are un-shuffled.

These recombination mechanisms are obviously not suitable for a continuous encoding. Intermediate recombination presumes that the value of each gene of the offspring is a convex combination of the corresponding values of the parents.

Mutation

Through the use of mutation , which induces small changes to an individual, potential solutions that could never be obtained otherwise are introduced in the population. The effect of mutation is the modification of the values of several genes of an individual. The genes that undergo mutation are chosen probabilistically, according to the value that is appointed to the corresponding parameter of the EA.

For every individual in the current population and each gene of that individual, a random number in the [0, 1] interval is generated. If the mutation probability is higher than the generated number, then that gene suffers mutation. Sometimes global search is intended in the initial evolutionary phases and local search (fine tuning) is planned towards the final steps of the EA. In this respect, the mutation probability may decrease with the increase in the number of generations.

Another distinct situation concerns those cases when the change of a gene lying at the beginning of a individual could make a significant modification to the individual in question, while the same change at the end of the individual would induce a less significant alteration; here, while the number of generations increases, the probability of mutation of the first genes in every individual may decrease and that of the final ones increase.

There are several forms of this operator as well, again depending on the specific task and the considered representation. The more usual situations of discrete and continuous encodings are further outlined and the most frequent types are described [Dumitrescu, 2000], [Dumitrescu et al., 2000], [Eiben and Smith, 2003].

In the particular binary case, a strong mutation presumes that, when a gene undergoes mutation, 1 changes into 0 and 0 into 1. Weak mutation supposes that the above change does not take place automatically as before, but 1 or 0 is probabilistically chosen and attributed to that position. In this way, the newly generated value could be identical to the old one, so no effective change would actually occur. For the universal discrete case, the flipping is generalized in the sense that the value of a gene probabilistically changes into one of the allowed choices for that position.

For the continuous instances, mutation customary performs a small perturbation in the value of the selected gene, which is induced by a parameter called mutation strength multiplied by a number that is randomly generated, usually either following a uniform or normal distribution.

2.4.6 Survivor Selection

Any EA has to determine in what amount do the individuals in the population of one generation find themselves among the individuals in the next generation or, in other words, survive that generation [Dumitrescu, 2000], [Dumitrescu et al., 2000], [Eiben and Smith, 2003]. Generally, the population of the next generation is formed of the newly obtained individuals plus the individuals that were not selected for reproduction; this means that the offspring resulting after either recombination or mutation automatically replace their parents. Conversely, the offspring and the parents may fight for a place in the subsequent generation. Another possibility is to copy, without any modification, a fixed number of the best individuals from the current population to the next and let the same individuals also participate in the parent selection.

2.4.7 Stop Condition

The evolutionary process generally stops either after a predefined number of generations or a certain number of fitness evaluations . There are also some situations when the EA is terminated after a number of iterations with no fitness improvement or the solution is reached with the desired accuracy or when the population diversity falls below a given limit [Dumitrescu, 2000], [Dumitrescu et al., 2000], [Eiben and Smith, 2003].

2.5 Applications to Data Mining

EC methods have a wide range of applications, acting either as stand-alone or in hybridization with problem-specific methods. As the practical side of this thesis targets the data mining domain, many important evolutionary techniques that regard this direction and, moreover, have been concerned with the particular problems of classification and clustering can be mentioned.

Two classical evolutionary classifiers are represented by the Michigan and Pittsburgh learning strategies. The Pittsburgh approach [Holland, 1986] aims to evolve a complete classification rule set by encoding the entire structure into each individual. Conversely, the Michigan alternative [Michalewicz, 1992] considers every individual as the representative of one rule, uses a credit assignment system to reward/penalize the collaboration between rules and struggles to achieve the complete optimal rule set as the output of the EA. Automated classification can be accomplished through the means of an evolution program that simultaneously searches for both the optimum number of classes and the optimal classification [Luchian et al., 1994]. Cooperative coevolution can also be used as a learning engine [Stoean et al., 2005]. Clustering [Dumitrescu and Gorunescu, 2004] or classification [Stoean et al., 2008a] can be otherwise approached by certain multimodal EAs. As regards the hybridization of EAs with classification-tailored methods, the combination with the powerful paradigm of neural networks [Yao and Liu, 1997] can be as well successfully employed for the classification task. Additionally, the present thesis will focus on the mixture of EAs with the other state-of-the-art learning technique of support vector machines.

2.6 Remarks

EAs are easy and straightforward for any task, while their performance as general optimizers is very competitive. Their power and success lie in the simplicity of their mathematical functioning, the naturalness of underlying metaphor and the easy tailoring for any given problem.

Chapter 3

Learning Within Support Vector Machines

3.1 Aims of This Chapter

This chapter presents the learning conception of the state-of-the-art technique of support vector machines (SVMs) . The SVM paradigm represents *a system for efficiently training linear learning machines in kernel-induced feature spaces, while respecting the insights of generalization theory and exploiting optimization theory* [Cristianini and Shawe-Taylor, 2000].

SVMs have lately been established as a viable approach for classification and regression [Haykin, 1999], [Hsu and Lin, 2004], [Mierswa, 2006b], [Olafsson et al., 2006], [Trafalis and Gilbert, 2006], [Vapnik, 1995b]. The chapter outlines the SVM perception upon learning and the principles that govern the technique (section 3.2). Classification and regression have fundamentally different learning tasks and, consequently, SVMs approach them in distinctive ways. The learning view upon classification of SVMs is targeted in section 3.3, while that concerning support vector regression is pursued in section 3.4.

3.2 Fundaments of Support Vector Machines

Given $\{(x_i, y_i)\}_{i=1,2,\ldots,m}$, a training set where every $x_i \in R^n$ represents a data sample and each y_i corresponds to a target, a learning task is concerned with the discovery of the optimal function that minimizes the discrepancy between the given targets of data samples and the predicted ones; the outcome of previously unknown samples is then tested.

The SVM technique pursues equally classification and regression problems. The task for

classification is to achieve an optimal separation of given data into classes. SVMs regard learning in this situation from a geometrical point of view: They assume the existence of a separating surface between every two classes labelled as -1 and 1. The aim then becomes the discovery of the appropriate decision hyperplane.

The standard assignment of SVMs for regression is to find the optimal function to be fitted to the data such that it achieves at most ϵ deviation from the actual targets of samples; the purpose thus becomes to estimate the optimal regression coefficients of such a function.

3.3 Support Vector Machines for Classification

The placement of data samples to be classified triggers corresponding separating surfaces within SVM learning . The technique basically considers only the general case of binary classification and treats reductions of multi-class tasks; subsection 3.3.5 explains the possible methods to handle the latter situation.

Essentially, while targeting classification, SVMs must obey a fundamental theoretical assumption, the principle of structural risk minimization (SRM) [Vapnik and Chervonenkis, 1968], [Vapnik and Chervonenkis, 1974], [Vapnik, 1982], [Vapnik, 1995b], [Vapnik, 1995a], which is further on depicted in subsection 3.3.1.

3.3.1 Principle of Structural Risk Minimization

Recall the formulation of a learning problem in section 3.2 and consider the case of a binary classification task, $y_i \in \{-1, 1\}$. Let it additionally be given a set of functions which are parameterized by some vector t:

$$\{f_t | t \in T\}, f_t : \Re^n \rightarrow \{-1, +1\}$$

Proposition 1. *(Structural Risk Minimization principle) [Vapnik, 1982]*

For the considered learning problem, for any parameters $t \in T$ and for m > h, with a probability of at least $1 - \eta$, the following inequality

$$R(t) \leq R_{emp}(t) + \phi\left(\frac{h}{m}, \frac{\log(\eta)}{m}\right)$$

holds, where R(t) is the test error, $R_{emp}(t)$ is the training error and ϕ is called the confidence term and is defined as:

$$\phi\left(\frac{h}{m}, \frac{\log(\eta)}{m}\right) = \sqrt{\frac{h\left(\log\frac{2m}{h}+1\right) - \log\frac{\eta}{4}}{m}}.$$

Definition 1. *The parameter h is called the VC (Vapnik-Chervonenkis) – dimension of a set of functions. A given set of m input samples can be labelled in 2^m possible ways. If for each labelling, a member of the set $\{f_t|t \in T\}$ can be found to correctly assign those labels, then it is said that the set of samples is shattered by that set of functions. Now, the VC-dimension for a set of functions $\{f_t|t \in T\}$ is defined as the maximum number of training samples that can be shattered by it.*

The SRM principle basically states that, in order to achieve a high generalization ability, both the training error and the confidence term must be kept small. It is important to notice that the confidence term is minimized by reducing the VC-dimension.

Intuitively speaking, SRM states that, for a given learning task, with a certain amount of training data, generalization performance is solely achieved if the accuracy on the particular training set and the *capacity* of the machine to pursue learning on *any* other training set without error have a good balance. This request can be illustrated by a simple natural example: *A machine with too much capacity is like a botanist with photographic memory who, when presented with a new tree, concludes that it is not a tree because it has a different number of leaves from anything she has seen before; a machine with too little capacity is like the botanist's lazy brother, who declares that if it's green, then it's a tree. Neither can generalize well* [Burges, 1998].

3.3.2 Linear Support Vector Machines: The Separable Case

If training data is known to be linearly separable, then there exists a linear hyperplane of equation (3.1) :

$$\langle w, x\rangle - b = 0, \tag{3.1}$$

which separates the samples according to classes [Burges, 1998], [Haykin, 1999].

The positive data samples lie on the corresponding side of the hyperplane and their negative counterparts on the opposite side.

As a consequence, Proposition 2 arises [Burges, 1998], [Haykin, 1999]:

Proposition 2. *Two subsets of n-dimensional samples are linearly separable iff there exist* $w \in R^n$ *and* $b \in R$ *such that (3.2):*

$$\begin{cases} \langle w, x_i \rangle - b \geq 0, y_i = +1 \\ \langle w, x_i \rangle - b \leq 0, y_i = -1 \end{cases}, i = 1, 2, ..., m \tag{3.2}$$

An illustration of this geometric concept can be found in Figure 3.1.

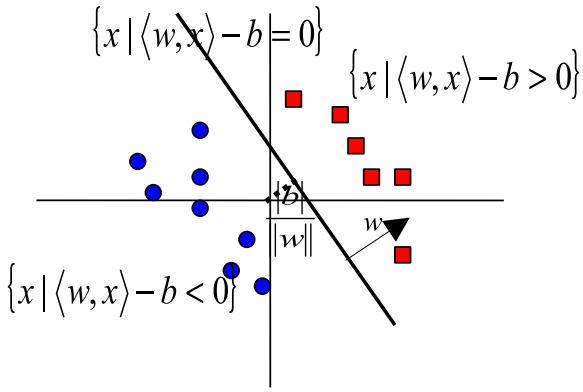

Figure 3.1: The positive and negative samples and the separating hyperplane between the two corresponding separable subsets.

As a stronger statement for linear separability, each of the positive and negative samples lies on the corresponding side of a matching supporting hyperplane for the respective class.

Proposition 3. *[Bosch and Smith, 1998] Two subsets of n-dimensional samples are linearly separable iff there exist* $w \in R^n$ *and* $b \in R$ *such that:*

$$\begin{cases} \langle w, x_i \rangle - b \geq +1, y_i = +1 \\ \langle w, x_i \rangle - b \leq -1, y_i = -1 \end{cases}, i = 1, 2, ..., m \tag{3.3}$$

Proof. " $\Leftarrow$ " The subsets given by $y_i = +1$ and $y_i = -1$, respectively, are linearly separable since all positive samples lie on one side of the hyperplane given by

$$\langle w, x \rangle - b = 0,$$

since:

$\langle w, x_i \rangle - b \geq 1 > 0$ for $y_i = +1$,

and simultaneously:

$\langle w, x_i \rangle - b \leq -1 < 0$ for $y_i = -1$,

so all negative samples lie on the other side of this hyperplane.

" $\Rightarrow$ " Suppose the two subsets are linearly separable.
Then, there exist $w \in R^n$ and $b \in R$ such that:

$$\begin{cases} \langle w, x_i \rangle - b \geq 0, y_i = +1 \\ \langle w, x_i \rangle - b \leq 0, y_i = -1 \end{cases} \text{, for } i = 1, 2, ..., m$$

Since:

$$\min \{\langle w, x_i \rangle | y_i = +1\} > \max \{\langle w, x_i \rangle | y_i = -1\},$$

if one sets:

$$p = \min \{\langle w, x_i \rangle | y_i = +1\} - \max \{\langle w, x_i \rangle | y_i = -1\}$$

and makes:

$$w' = \frac{2}{p} w$$

and

$b' = \frac{1}{p} (\min \{\langle w, x_i \rangle | y_i = +1\} + \max \{\langle w, x_i \rangle | y_i = -1\})$,
then:

$$\min\{\langle w', x_i\rangle \,|y_i = +1\} =$$
$$= \tfrac{2}{p}\min\{\langle w, x_i\rangle \,|y_i = +1\} =$$
$$= \tfrac{1}{p}(\min\{\langle w, x_i\rangle \,|y_i = +1\} + \max\{\langle w, x_i\rangle \,|y_i = -1\} +$$
$$\min\{\langle w, x_i\rangle \,|y_i = +1\} - \max\{\langle w, x_i\rangle \,|y_i = -1\}) =$$
$$= \tfrac{1}{p}(\min\{\langle w, x_i\rangle \,|y_i = +1\} + \max\{\langle w, x_i\rangle \,|y_i = -1\} + p) =$$
$$= b' + 1$$

and

$$\max\{\langle w', x_i\rangle \,|y_i = -1\} =$$
$$= \tfrac{2}{p}\max\{\langle w, x_i\rangle \,|y_i = -1\} =$$
$$= \tfrac{1}{p}(\min\{\langle w, x_i\rangle \,|y_i = +1\} + \max\{\langle w, x_i\rangle \,|y_i = -1\} - p) =$$
$$= b' - 1$$

Consequently, there exist $w \in R^n$ and $b \in R$ such that:

$\langle w, x_i\rangle \geq b + 1 \Rightarrow \langle w, x_i\rangle - b \geq 1$ when $y_i = +1$

and $\langle w, x_i\rangle \leq b - 1 \Rightarrow \langle w, x_i\rangle - b \leq 1$ when $y_i = -1$

□

Remark 1. *Basically, in the procedure above, a scaling of the parameters for the separating hyperplane was performed.*

Remark 2. *Consider the data points (x_i, y_i) for which either the first or the second line of equation (3.3) holds with the equality sign. They are called support vectors . They are data points that lie closest to the decision surface. Their removal would change the found solution.*

An illustration of the stronger separation is given in Figure 3.2. The separating hyperplane is the one that lies in the middle of the two parallel supporting hyperplanes for the two classes .

At this moment, SVMs must determine the optimal values for the coefficients *w* and *b* of the decision hyperplane that separates the training data with as few exceptions as possible. Following (3.3), the optimal w and b must then satisfy :

$$y_i(\langle w, x_i\rangle - b) - 1 \geq 0, i = 1, 2, ..., m \tag{3.4}$$

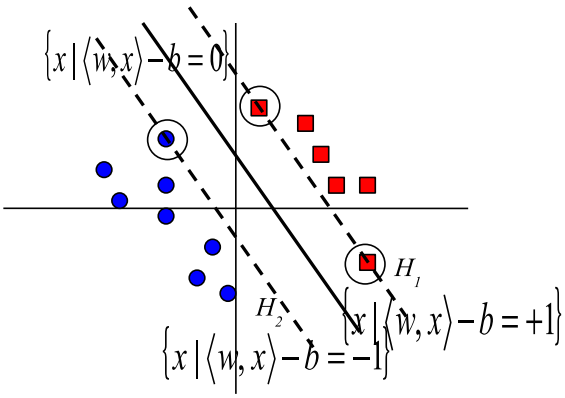

Figure 3.2: The separating and supporting hyperplanes for the separable subsets. The support vectors are circled.

In addition, according to the SRM principle, separation must be performed with a high generalization capacity.

Let the margin of separation between classes be determined.

The distance from one random sample *z* to the separating hyperplane is given by $\frac{|\langle w,z\rangle - b|}{\|w\|}$.

It results that the distance from the samples z_i that lie closest to the separating hyperplane, on either side of it, is:

$\frac{|\langle w,z_i\rangle - b|}{\|w\|} = \frac{1}{\|w\|}$ for all *i* ($|\langle w, z_i\rangle - b| = 1$, as, lying closest to the separating hyperplane, either $z_i \in H_1$ or $z_i \in H_2$ for all *i*).

This results in the fact that the margin of separation is [Vapnik, 2003]:

$$\frac{2}{\|w\|}. \tag{3.5}$$

The following proposition can be enunciated [Burges, 1998]:

Proposition 4. *Let R be the radius of the smallest ball*

$$B_R(a) = \{x \in \Re^n | \, \|x - a\| < R\}, a \in \Re^n$$

containing the samples $x_1, ..., x_m$ *and let*

$$f_{w,b} = sgn(\langle w, x\rangle - b)$$

be hyperplane decision functions defined on these samples.

Then the set $\{f_{w,b} | \|w\| \leq A\}$ *has a VC-dimension satisfying*

$$h < R^2 A^2 + 1$$

Intuitively, Proposition 4 states that, due to the fact that, following (3.5), $\|w\|$ is inversely proportional to the margin of separation, by requiring a large margin (*i.e.* a small *A*), a small VC-dimension is obtained. Conversely, by allowing for separations with small margin, a much larger class of problems can be potentially separated (*i.e.* a larger class of possible labelling for the training samples, following Definition 1 of VC-dimension) .

But, as the SRM principle requests that, in order to achieve high generalization of the classifier, training error and VC-dimension must be both kept small, hyperplane decision functions must be therefore constrained such that they maximize the margin , *i.e.*

$$minimize \frac{\|w\|^2}{2}, \tag{3.6}$$

and separate the training data with as few exceptions as possible.

From (3.4) and (3.6), it follows that the optimization problem is (3.7) :

$$\begin{cases} \text{find } w \text{ and } b \text{ as to minimize } \frac{\|w\|^2}{2} \\ \text{subject to } y_i(\langle w, x_i \rangle - b) \geq 1, i = 1, 2, ..., m. \end{cases} \tag{3.7}$$

3.3.3 Linear Support Vector Machines: The Nonseparable Case

Generally, training data are not linearly separable. In the nonseparable case, it is obvious that a linear separating hyperplane is not able to build a partition without any errors. However, a linear separation that minimizes training error can be tried as a solution to the classification problem [Haykin, 1999].

The idea is to relax the separability statement by introducing some so-called slack variables .

This relaxation can be achieved by observing the deviations of data samples from the corresponding supporting hyperplane, i.e. from the ideal condition of data separability. Such a deviation corresponds to a value of $\frac{\pm \xi_i}{\|w\|}$, $\xi_i \geq 0$ [Cortes and Vapnik, 1995].

These values may indicate different nuanced digressions (Figure 3.3), but only a ξ_i higher than unity signals a classification error.

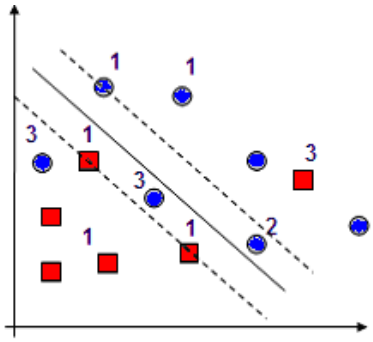

Figure 3.3: Position of data and corresponding indicators for errors - correct placement, $\xi_i = 0$ (label 1) margin position, $\xi_i < 1$ (label 2) and classification error, $\xi_i > 1$ (label 3)

Minimization of training error is achieved by adding the indicator for error (slack variable) for every training data sample into the separability statement and, at the same time, by minimizing the sum of indicators for errors.

The constraints in (3.4) subsequently become:

$$y_i(\langle w, x_i \rangle - b) \geq 1 - \xi_i, i = 1, 2, ..., m, \tag{3.8}$$

where $\xi_i \geq 0$.

And, simultaneously with (3.8), sum of misclassification situations must be minimized:

$$minimize C \sum_{i=1}^{m} \xi_i, \tag{3.9}$$

where *C* is a SVM hyperparameter, a larger C corresponding to assigning a higher penalty for errors .

Remark 3. *The data points* (x_i, y_i) *for which* (3.8) *holds with the equality sign are the support vectors.*

Therefore, the optimization problem changes to 3.10 :

$$\begin{cases} \text{find } w \text{ and } b \text{ as to minimize } \frac{\|w\|^2}{2} + C\sum_{i=1}^{m}\xi_i, \\ C > 0 \\ \text{subject to } y_i(\langle w, x_i\rangle - b) \geq 1 - \xi_i, \xi_i \geq 0, \\ i = 1, 2, ..., m. \end{cases} \tag{3.10}$$

It can be seen that this approach still obeys the SRM principle as the VC-dimension is still minimized and separation of training data with as few exceptions as possible is again achieved, both through (3.8) and (3.9).

Illustration of a linear separation for nonseparable training data is given in Figure 3.4.

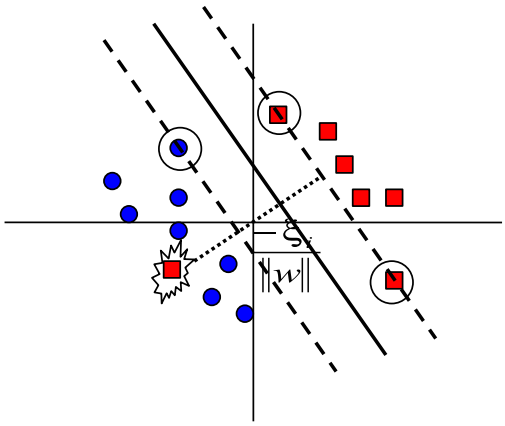

Figure 3.4: The separating and supporting linear hyperplanes for the nonseparable training subsets. The support vectors are circled and the misclassified data point is highlighted.

3.3.4 Nonlinear Support Vector Machines

If a linear hyperplane does not provide satisfactory results for the classification task, then is it possible that a nonlinear decision surface is appointed? [Burges, 1998], [Haykin, 1999]. The answer is positive and is based on the following result .

Theorem 1. *[Cover, 1965] A complex pattern classification problem cast in a high-dimensional space nonlinearly is more likely to be linearly separable than in a low-dimensional space.*

The above theorem states that an input space can be transformed into a new feature space where data is linearly separable with high probability, provided:

1. The transformation is nonlinear.

2. The dimensionality of the feature space is high enough.

The initial space of training data samples can thus be nonlinearly mapped into a higher dimensional feature space, where a linear decision hyperplane can be subsequently built. The separating hyperplane will achieve an accurate classification in the feature space which will correspond to a nonlinear decision function in the initial space (Figure 3.5).

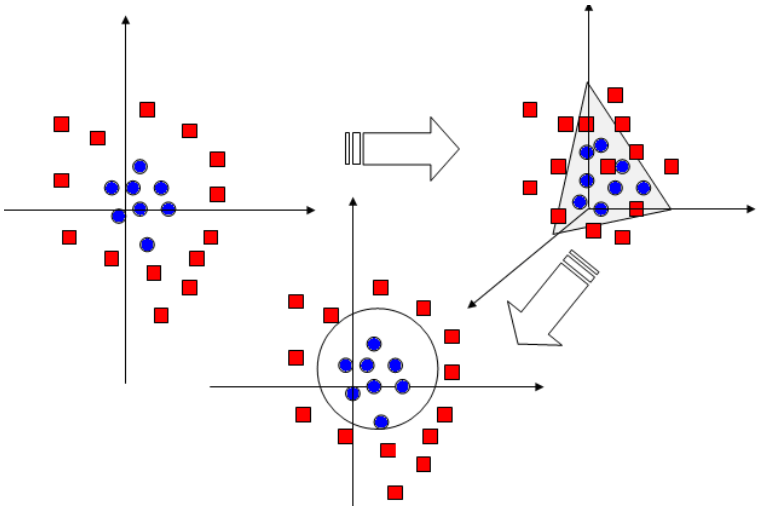

Figure 3.5: Initial data space (left), nonlinear map into the higher dimension where the objects are linearly separable/the linear separation (right), and corresponding nonlinear surface.

The procedure therefore leads to the creation of a linear separating hyperplane that would, as before, minimize training error, only this time it would perform in the feature space. Accordingly, a nonlinear map $\Phi : R^n \rightarrow H$ is considered and data samples from the initial space are mapped into H.

As it will be seen in chapter 4, in the training SVM mechanism, vectors appear only as part of scalar products; the issue can be thus further simplified by substituting the scalar product by a kernel, which is a function with the property that (3.11) :

$$K(x, y) = \langle \Phi(x), \Phi(y) \rangle \,, \tag{3.11}$$

where $x, y \in \Re^n$.

The kernel can be perceived as to express the similarity between samples. SVMs require that the kernel is a positive (semi-)definite function in order for the standard solving approach

to find a solution to the optimization problem [Mierswa, 2006b]. Such a kernel is one that satisfies Mercer's theorem from functional analysis and is, therefore, a scalar product in some space [Burges, 1998].

Theorem 2. *[Boser et al., 1992], [Courant and Hilbert, 1970], [Haykin, 1999], [Mercer, 1908]*

Let K(x,y) be a continuous symmetric kernel that is defined in the closed interval $a \leq x \leq b$ *and likewise for y. The kernel K(x,y) can be expanded in the series*

$$K(x,y) = \sum_{i=1}^{\infty} \lambda_i \Phi(x)_i \Phi(y)_i$$

with positive coefficients, $\lambda_i > 0$ *for all i. For this expansion to be valid and for it to converge absolutely and uniformly, it is necessary that the condition*

$$\int_a^b \int_a^b K(x,y)\psi(x)\psi(y)dxdy \geq 0$$

holds for all $\psi(\cdot)$ *for which*

$$\int_a^b \psi^2(x)dx < \infty$$

The problem with this restriction is twofold [Mierswa, 2006b]. On the one hand, Mercer's condition is very difficult to check for a newly constructed kernel. On the other hand, kernels that fail the theorem could prove to achieve a better separation of the training samples.

Applied SVMs consequently use a couple of classical kernels that had been demonstrated to meet Mercer's condition [Boser et al., 1992], [Vapnik, 1995b]:

- the polynomial kernel of degree p : $K(x,y) = \langle x,y \rangle^p$
- the radial basis function kernel : $K(x,y) = e^{\frac{\|x-y\|^2}{\sigma}}$, where p and σ are also hyperparameters of SVMs.

However, as a substitute for the original solving, a direct search technique does not depend on the condition whether the kernel is positive (semi-)definite or not.

Illustration of the construction of a linear separating hyperplane in the feature space and its correspondent nonlinear hyperplane in the input space is shown in Figure 3.6 where a polynomial kernel of odd degree is considered. Figures 3.7 and 3.8 show the creation of nonlinear hyperplanes when the kernels are, in turn, even polynomial and radial, respectively.

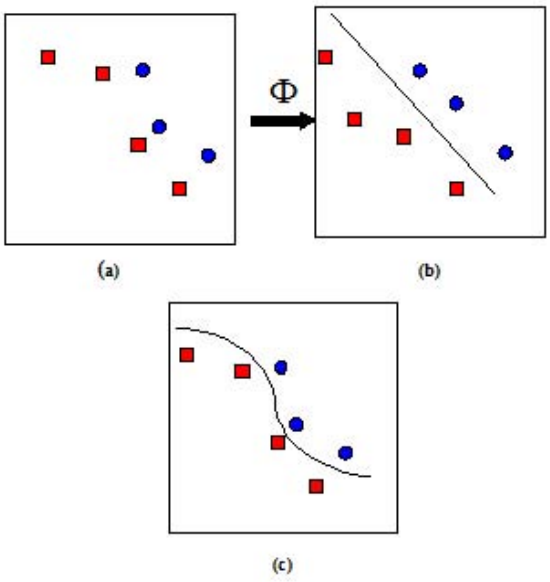

Figure 3.6: The mapping of input data (a) nonlinearly (via Φ) into a higher-dimensional feature space and the construction of the separating hyperplane there (b) corresponding to a nonlinear separating hyperplane in the input space (c). An odd polynomial kernel was used.

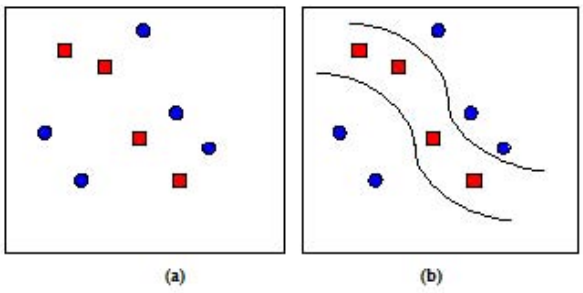

Figure 3.7: Construction of the separating hyperplane for input data (a) using an even polynomial kernel (b).

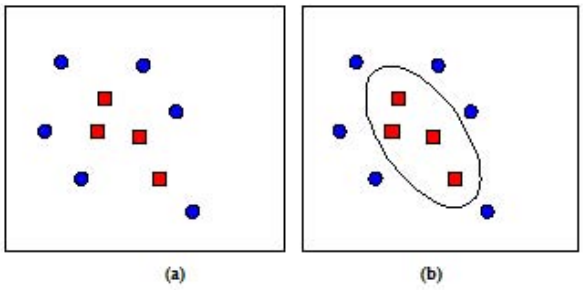

Figure 3.8: Construction of the separating hyperplane for input data (a) using a radial kernel (b).

3.3.5 Design of Multi-class Support Vector Machines

Multi-class SVMs build several two-class classifiers that separately solve the matching tasks . The translation from multi-class to two-class is performed through different systems, among which one-against-all, one-against-one or decision directed acyclic graph are the most commonly employed.

One-against-all Approach

The one-against-all (1aa) technique [Hsu and Lin, 2004] builds k classifiers. Every i^{th} SVM considers all training samples labelled with i as positive and all the remaining as negative.

Consequently, by placing the problem in the initial space, the aim of every i^{th} SVM is to determine the optimal coefficients w and b of the decision hyperplane which best separates the samples with outcome i from all the other samples in the training set, such that (3.12) :

$$\begin{cases} \text{minimize } \frac{\|w^i\|^2}{2} + C\sum_{j=1}^{m}\xi_j^i, \\ \text{subject to } y_j(\langle w^i, x_j\rangle - b) \geq 1 - \xi_j^i, \\ \xi_j^i \geq 0 \\ j = 1, 2, ..., m, i = 1, 2, ..., k. \end{cases} \tag{3.12}$$

One-against-one Approach

The one-against-one (1a1) technique [Hsu and Lin, 2004] builds $\frac{k(k-1)}{2}$ SVMs. Every i^{th} machine is trained on data from every two classes, *i* and *j*, where samples labelled with *i* are considered positive while those in class j are taken as negative.

Accordingly, living once more in the initial space, the aim of every SVM is to determine the optimal coefficients w and b of the decision hyperplane which best separates the samples with outcome i from the samples with outcome j, such that (3.13) :

$$\begin{cases} \text{minimize } \frac{\|w^{ij}\|^2}{2} + C\sum_{l=1}^{m}\xi_l^{ij}, \\ \text{subject to } y_l(\langle w^{ij}, x_l\rangle - b) \geq 1 - \xi_l^{ij}, \\ \xi_l^{ij} \geq 0 \\ l = 1, 2, ..., m, i, j = 1, 2, ..., k, i \neq j. \end{cases} \tag{3.13}$$

Decision Directed Acyclic Graph

Learning within the decision directed acyclic graph (DDAG) technique [Platt et al., 2000] proceeds in an identical manner to that of 1a1.

3.4 Support Vector Regression

Recall the learning formulation at the beginning of section 3.2 and consider the regression case, $y_i \in R$. Such a data set could represent exchange rates of a currency measured in subsequent days together with econometric attributes [Smola and Scholkopf, 1998] or a medical indicator registered in multiple patients along with personal and medical information [Altman, 1991].

The task of ϵ-support vector regression (SVR) [Vapnik, 1995b] is to find a function $f(x)$ that has at most ϵ deviation from the actual targets of data samples and, simultaneously, is as flat as possible [Smola and Scholkopf, 1998]. In other words, the aim is to estimate the regression coefficients of $f(x)$ with these requirements.

While the former condition for ϵ-SVR is straightforward, errors are allowed as long as they are less than ϵ, the latter one needs some further explanation [Rosipal, 2001]. Resulting values of the regression coefficients may affect the model in the sense that it fits current training data but has low generalization ability. In order to avoid this situation, it is required to choose the flattest function in the definition space.

Another way to interpret ϵ-SVR is that training data are constrained to lie on a hyperplane that allows for some error and, at the same time, has high generalization ability.

3.4.1 Linear Support Vector Machines for Regression

Suppose a linear regression model can fit the training data. Consequently, function f has the form 3.14 :

$$f(x) = \langle w, x \rangle - b. \tag{3.14}$$

The task of ϵ-SVR is then mathematically translated as follows. On the one hand, the condition that f approximates training data with ϵ precision is written as:

$$\begin{cases} y_i - \langle w, x_i \rangle + b \leq \epsilon \\ \langle w, x_i \rangle - b - y_i \leq \epsilon \end{cases}, i = 1, 2, ..., m \tag{3.15}$$

On the other hand, flattest function means smallest slope (w) which leads to condition :

$$\text{minimize } \|w\|^2 \tag{3.16}$$

Summing (3.15) and (3.16) up, the optimization problem that is reached in the case of linear ϵ-SVR is stated as (3.17) :

$$\begin{cases} \text{find } w \text{ and } b \text{ as to minimize } \|w\|^2 \\ \text{subject to } \begin{cases} y_i - \langle w, x_i \rangle + b \leq \epsilon \\ \langle w, x_i \rangle - b - y_i \leq \epsilon. \end{cases} , i = 1, 2, ..., m \end{cases} \tag{3.17}$$

3.4.2 Linear Support Vector Machines for Regression with Indicators for Errors

It may happen that the linear function f is not able to fit all training data and consequently ϵ-SVR will also allow for some errors, analogously to the corresponding situation in SVMs for classification [Cortes and Vapnik, 1995], [Haykin, 1999] .

Therefore, the positive slack variables ξ_i and ξ_i^*, both attached to each sample, are introduced into the condition for approximation of training data:

$$\begin{cases} y_i - \langle w, x_i \rangle + b \leq \epsilon + \xi_i, \\ \langle w, x_i \rangle - b - y_i \leq \epsilon + \xi_i^*. \end{cases} , i = 1, 2, ..., m \tag{3.18}$$

Simultaneously, the sum of these indicators for errors is minimized:

$$C \sum_{i=1}^{m} (\xi_i + \xi_i^*), \tag{3.19}$$

where C denotes again the penalty for errors.

Adding (3.18) and (3.19) up, the optimization problem in the case of linear ϵ-SVR with indicators for errors is written as (3.20) :

$$\begin{cases} \text{find } w \text{ and } b \text{ as to minimize } \|w\|^2 + C\sum_{i=1}^{m}(\xi_i + \xi_i^*) \\ \text{subject to } \begin{cases} y_i - \langle w, x_i\rangle + b \leq \epsilon + \xi_i \\ \langle w, x_i\rangle - b - y_i \leq \epsilon + \xi_i^* \quad , i = 1, 2, ..., m \\ \xi_i, \xi_i^* \geq 0 \end{cases} \end{cases} \tag{3.20}$$

3.4.3 Nonlinear Support Vector Machines for Regression

If a linear function is not at all able to fit training data, a nonlinear one has to be chosen. The procedure follows the same steps as in SVMs for classification [Cover, 1965]. Data is mapped via a nonlinear function into a high enough dimensional space and linearly modelled there as in the previous section. This corresponds to a nonlinear regression hyperplane in the initial space .

3.5 Remarks

SVMs provide a very efficient vision upon learning. They pursue a geometrical interpretation of the relationship between samples and decision surfaces and thus manage to formulate a simple and natural optimization task. However, the method SVMs employ for the subsequent training is constrained and intricate, as it can be seen in the following chapter.

Chapter 4

Training Within Support Vector Machines

4.1 Aims of This Chapter

This chapter addresses the theoretical aspects and mechanism of the classical approach to solving the constrained optimization problem within SVMs. As before, discussion of the concepts behind the classical solving addresses classification and regression in turn and pursues SVM training from the existence of a linear decision function to the creation of a nonlinear surface .

4.2 Linear Support Vector Classification: The Separable Case

Let the constrained optimization problem be stated again [Haykin, 1999]:

$$\begin{cases} \text{find } w \text{ and } b \text{ as to minimize } \frac{\|w\|^2}{2} \\ \text{subject to } y_i(\langle w, x_i \rangle - b) \geq 1, i = 1, 2, ..., m. \end{cases}$$

Remark 4. *The scaling factor $\frac{1}{2}$ is included for convenience of presentation.*

The constrained optimization problem at hand is called the primal problem (PP) .

4.2.1 Properties of the Primal Problem

A few properties of PP have to be discussed prior to solving it.

Definition 2. *A function $f : C \to \Re$ is said to be convex if*

$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y)$*, for all $x, y \in C$ and $\alpha \in [0, 1]$.*

The following propositions are well-known results in fundamental mathematics.

Proposition 5. *For a function $f : (a, b) \to \Re$, $(a, b) \subseteq \Re$, that has a second derivative in (a,b), a necessary and sufficient condition for its convexity on that interval is that the second derivative $f''(x) \geq 0$ for all $x \in (a, b)$.*

Proposition 6. *If two functions are convex, the composition of the functions is convex.*

Proposition 7. *prop:convexity2 The objective function in PP is convex [Haykin, 1999].*

Proof. Let $h = f \circ g$, where $f : \Re \to \Re$, $f(x) = x^2$ and $g : \Re^n \to \Re$, $g(w) = \|w\|$.

Let the two functions be addressed in turn.

1. $f : \Re \to \Re$, $f(x) = x^2 \Rightarrow f'(x) = 2x \Rightarrow f''(x) = 2 \geq 0 \Rightarrow f$ is convex.

2. $g : \Re^n \to \Re$, $g(w) = \|w\|$

Two properties of a norm are:

1. $\|\alpha v\| = |\alpha| \, \|v\|$

2. $\|v + w\| \leq \|v\| + \|w\|$

Let $v, w \in \Re^n$ and $\alpha \in [0, 1]$.

$$g(\alpha v + (1 - \alpha)w) = \|\alpha v + (1 - \alpha)w\| \leq |\alpha| \, \|v\| + |1 - \alpha| \, \|w\| = \alpha \, \|v\| + (1 - \alpha) \, \|w\| =$$

$$= \alpha g(v) + (1 - \alpha)g(w)$$

$\Rightarrow g$ is convex.

Following Proposition 6 $\Rightarrow h$ is convex.

□

Remark 5. *Constraints in PP are linear in w.*

Proposition 8. *The feasible region for a constrained optimization problem is convex if the constraints are linear.*

The standard training method of finding the optimal solution with respect to defined constraints resorts to an extension of the Lagrange multipliers method.

4.2.2 The Karush-Kuhn-Tucker-Lagrange Conditions

Given the fact that the objective function is convex and that constraints are linear, the Karush-Kuhn-Tucker-Lagrange (KKTL) conditions can be stated for PP [Haykin, 1999] .

This is based on the fact that since constraints are linear, the KKTL conditions are guaranteed to be necessary. Also, since PP is convex (convex objective function + convex feasible region), the KKTL conditions are at the same time sufficient for global optimality [Fletcher, 1987].

First, the Lagrangian function is constructed:

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^{m} \alpha_i \left[y_i(\langle w, x_i \rangle - b) - 1 \right], \tag{4.1}$$

where the variables α_i are the Lagrange multipliers.

The solution to the problem is determined by the KKTL conditions [Burges, 1998]:

$$\begin{cases} \frac{\partial L(w,b,\alpha)}{\partial w} = 0 \\ \frac{\partial L(w,b,\alpha)}{\partial b} = 0 \\ \alpha_i \left[y_i(\langle w, x_i \rangle - b) - 1 \right] = 0, i = 1, 2, .., m \end{cases}$$

Differentiation with respect to a vector

Let *f(w)* denote a real-valued function of parameter vector *w*. The derivative of *f* with respect to *w* is defined by the vector [Haykin, 1999]:

$$\frac{\partial f}{\partial w} = \left[\frac{\partial f}{\partial w_1}, \frac{\partial f}{\partial w_2}, ..., \frac{\partial f}{\partial w_n} \right]^T,$$

where *n* is the dimension of the vector.

In the particular case when $f(w) = w^T w = \sum_{i=1}^{n} w_i w_i$ then $\frac{\partial f}{\partial w} = w$.

Application of the KKTL conditions yields:

$$\frac{\partial L(w, b, \alpha)}{\partial w} = w - \sum_{i=1}^{m} \alpha_i y_i x_i = 0 \Rightarrow w = \sum_{i=1}^{m} \alpha_i y_i x_i \tag{4.2}$$

$$\frac{\partial L(w, b, \alpha)}{\partial b} = \sum_{i=1}^{m} \alpha_i y_i = 0 \tag{4.3}$$

$$y_i(\langle w, x_i\rangle - b) - 1 \geq 0, i = 1, 2, .., m$$

$$\alpha_i \left[y_i(\langle w, x_i\rangle - b) - 1\right] = 0, i = 1, 2, .., m \tag{4.4}$$

$$\alpha_i \geq 0, i = 1, 2, .., m$$

4.2.3 Lagrange Multipliers and Duality

Generally, given the primal problem:

$$\text{minimize } f(x)$$
$$\text{subject to } \begin{cases} g_1(x) \geq 0 \\ \ldots \\ g_m(x) \geq 0 \end{cases},$$

the Lagrange multipliers are $\alpha = (\alpha_1^*, ..., \alpha_m^*)$, $\alpha_i^* \geq 0$, such that:

$$\inf_{g_1(x)\geq 0,\ldots,g_m(x)\geq 0} f(x) = \inf_{x\in\Re^n} L(x, \alpha^*),$$

where *L* is the Lagrangian function,

$$L(x, \alpha) = f(x) + \sum_{j=1}^{m} \alpha_j g_j(x), x \in \Re^n, \alpha \in \Re^m.$$

Then, one can resort to the dual function [Haykin, 1999]:

$$q(\alpha) = \inf_{x\in\Re^n} L(x, \alpha)$$

This naturally leads to the dual problem :

$$\text{maximize } q(\alpha)$$
$$\text{subject to } \alpha \geq 0$$

The optimal primal value is $f^* = \inf_{g_1(x)\geq 0,\ldots,g_r(x)\geq 0} f(x) = \inf_{x\in\Re^n} \sup_{\alpha\geq 0} L(x, \alpha)$.
The optimal dual value is $g^* = \sup_{\alpha\geq 0} q(\alpha) = \sup_{\alpha\geq 0} \inf_{x\in\Re^n} L(x, \alpha)$.
There is always that $q^* \leq f^*$. But, if there is convexity in the primal problem, then:

1. $q^* = f^*$

2. Optimal solutions of the dual problem are multipliers for the primal problem.

4.2.4 Dual Problem for the Constrained Optimization

Equation (4.1) is expanded and one obtains [Haykin, 1999]:

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^{m} \alpha_i y_i \langle w, x_i \rangle + b \sum_{i=1}^{m} \alpha_i y_i + \sum_{i=1}^{m} \alpha_i \tag{4.5}$$

The third term on the right-hand side of the expansion is zero from Equation (4.3).

Moreover, from Equation (4.2) we obtain:

$$\frac{1}{2} \|w\|^2 = \langle w, w \rangle = \sum_{i=1}^{m} \alpha_i y_i \langle w, x_i \rangle = \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle$$

Therefore, Equation (4.5) changes to:

$$L(w, b, \alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle$$

According to the duality concepts, by setting $Q(\alpha) = L(w, b, \alpha)$, one obtains the dual problem (DP):

$$\begin{gathered} \text{find } \{\alpha_i\}_{i=1,2,..,m} \\ \text{as to maximize } Q(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \\ \text{subject to the } \begin{cases} \sum_{i=1}^{m} \alpha_i y_i = 0 \\ \alpha_i \geq 0 \end{cases}, i = 1, 2, ..., m \end{gathered}$$

The optimum Lagrange multipliers are next determined by setting the gradient of *Q* to zero and solving the resulting system.

Then, the optimum vector *w* can be computed from Equation (4.2) as follows [Haykin, 1999]:

$$w = \sum_{i=1}^{m} \alpha_i y_i x_i$$

As *b* is concerned, it can be obtained from any of the equalities of Equation (4.4), when $\alpha_i \neq 0$. Then:

$$y_i(\langle w, x_i\rangle - b) - 1 = 0 \Rightarrow$$

$$y_i(\sum_{j=1}^{m} \alpha_j y_j \langle x_j, x_i\rangle - b) = 1 \Rightarrow$$

$$\sum_{j=1}^{m} \alpha_j y_j \langle x_j, x_i\rangle - b = y_i \Rightarrow$$

$$b = \sum_{j=1}^{m} \alpha_j y_j \langle x_j, x_i\rangle - y_i$$

It is nevertheless safer to take the mean value of *b* obtained after solving all such equations.

In the solution, those points for which $\alpha_i > 0$ are support vectors.

4.3 Linear Support Vector Classification: The Nonseparable Case

Recall that now the constrained optimization problem is defined as :

$$\begin{cases} \text{find } w \text{ and } b \text{ as to minimize } \frac{\|w\|^2}{2} + C\sum_{i=1}^{m} \xi_i, \\ C > 0 \\ \text{subject to } y_i(\langle w, x_i\rangle - b) \geq 1 - \xi_i, \xi_i \geq 0, \\ i = 1, 2, ..., m. \end{cases}$$

Therefore, the Lagrangian function becomes [Burges, 1998]:

$$L(w, b, \xi, \alpha, \mu) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m} \xi_i - \sum_{i=1}^{m} \alpha_i [y_i(\langle w, x_i\rangle - b) - 1 + \xi_i] - \sum_{i=1}^{m} \mu_i \xi_i,$$

where the variables α_i and μ_i are the Lagrange multipliers.

Application of the KKTL conditions to this new constrained optimization problem leads to [Burges, 1998]:

$$\frac{\partial L(w, b, \xi, \alpha, \mu)}{\partial w} = w - \sum_{i=1}^{m} \alpha_i y_i x_i = 0 \Rightarrow w = \sum_{i=1}^{m} \alpha_i y_i x_i$$

$$\frac{\partial L(w, b, \xi, \alpha, \mu)}{\partial b} = \sum_{i=1}^{m} \alpha_i y_i = 0$$

$$\frac{\partial L(w,b,\xi,\alpha,\mu)}{\partial \xi_i} = C - \alpha_i - \mu_i = 0 \Rightarrow \alpha_i + \mu_i = C, i = 1,2,..,m \tag{4.6}$$

$$y_i(\langle w, x_i\rangle - b) - 1 + \xi_i \geq 0, i = 1,2,..,m$$

$$\xi_i \geq 0, i = 1,2,..,m$$

$$\alpha_i \left[y_i(\langle w, x_i\rangle - b) - 1 + \xi_i\right] = 0, i = 1,2,..,m \tag{4.7}$$

$$\mu_i \xi_i = 0, i = 1,2,..,m \tag{4.8}$$

$$\alpha_i \geq 0, i = 1,2,..,m$$

$$\mu_i \geq 0, i = 1,2,..,m$$

The Lagrangian function then becomes, after expanding it term by term:

$$L(w,b,\xi,\alpha,\mu) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{i=1}^{m} \alpha_i\alpha_j y_i y_j \langle x_i, x_j\rangle + C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i\xi_i - \sum_{i=1}^{m}\mu_i\xi_i.$$

From Equation (4.8), the last term of the Lagrangian becomes zero and following Equation (4.6) and expanding the third term, one obtains:

$$L(w,b,\xi,\alpha,\mu) = \textstyle\sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{i=1}^{m} \alpha_i\alpha_j y_i y_j \langle x_i, x_j\rangle + \sum_{i=1}^{m}(\alpha_i + \mu_i)\xi_i -$$

$$\textstyle\sum_{i=1}^{m}\alpha_i\xi_i = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{i=1}^{m} \alpha_i\alpha_j y_i y_j \langle x_i, x_j\rangle + \sum_{i=1}^{m}\alpha_i\xi_i + \sum_{i=1}^{m}\mu_i\xi_i$$

$$\textstyle -\sum_{i=1}^{m}\alpha_i\xi_i = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{i=1}^{m} \alpha_i\alpha_j y_i y_j \langle x_i, x_j\rangle$$

Consequently, the following corresponding dual problem is obtained:

$$\text{find the } \{\alpha_i\}_{i=1,2,..,m}$$
$$\text{as to maximize } Q(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle$$
$$\text{subject to} \begin{cases} \sum_{i=1}^{m} \alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq C \end{cases}, i = 1, 2, ..., m, C > 0$$

The second constraint is obtained from Equation (4.6) and the condition that $\mu_i \geq 0$, $i = 1, 2, .., m$.

The optimum value for *w* is again computed as:

$$w = \sum_{i=1}^{m} \alpha_i y_i x_i$$

As for the computation of *b*, it can be determined as follows [Haykin, 1999]: If the values α_i obeying the condition $\alpha_i < C$ are considered, then from Equation (4.6) $\Rightarrow \mu_i \neq 0$. Subsequently, from Equation (4.8) $\Rightarrow \xi_i = 0$. Under these circumstances, from Equations (4.7) and (4.2) one obtains the same equation as in the separable case:

$$y_i(\langle w, x_i \rangle - b) - 1 = 0 \Rightarrow b = \sum_{j=1}^{m} \alpha_j y_j \langle x_j, x_i \rangle - y_i$$

Again it is better to take *b* as the mean resulting from all such equations.

Those points that have $0 < \alpha_i < C$ are support vectors.

4.4 Nonlinear Support Vector Classification

One may state the dual problem in this new case by simply replacing the scalar product between data points with the chosen kernel, as below :

$$\text{find } \{\alpha_i\}_{i=1,2,..,m}$$
$$\text{as to maximize } Q(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$
$$\text{subject to} \begin{cases} \sum_{i=1}^{m} \alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq C \end{cases}, i = 1, 2, ..., m, C > 0$$

As generally one is not able to construct the mapping Φ from the kernel *K*, the value for the optimum vector *w* cannot always be determined explicitly from the equation:

$$w = \sum_{i=1}^{m} \alpha_i y_i \Phi(x_i)$$

Consequently, one usually has to directly determine the class for a new data sample, as follows :

$$class(x) = sgn(\langle w, \Phi(x)\rangle - b)$$

Therefore, by replacing *w* with $\sum_{i=1}^{m} \alpha_i y_i \Phi(x_i)$, one gets:

$$f(x) = sgn(\langle w, \Phi(x)\rangle - b) =$$

$$sgn(\sum_{i=1}^{m} \alpha_i y_i \langle \Phi(x), \Phi(x_i)\rangle - b) =$$

$$sgn(\sum_{i=1}^{m} \alpha_i y_i K(x, x_i) - b)$$

One is left to determine the value of *b*. This is done by replacing the scalar product by the kernel in the same equations as in the linear case, *i.e.* when $0 < \alpha_i < C$:

$$b = \sum_{j=1}^{m} \alpha_j y_j K(x_j, x_i) - y_i,$$

and taking the mean of all the values obtained for *b*.

The classification accuracy is finally defined as the number of correctly labelled cases over the total number of test samples.

4.5 Multi-class Support Vector Machines

Resulting SVM decision functions are considered as a whole and the class for each sample in the test set is decided by corresponding systems [Hsu and Lin, 2004] .

One-against-all Approach

Once the all hyperplanes are determined following the classical SVM training as above, the class for a test sample x is given by the category that has the maximum value for the learning function, as in (4.9):

$$class(x) = argmax_{i=1,2,\ldots,k}(\langle w^i, \Phi(x)\rangle) - b^i). \tag{4.9}$$

One-against-one Approach

The moment the hyperplanes of the $\frac{k(k-1)}{2}$ SVMs are found, a *voting* method is used to determine the class for a test sample *x*. For every SVM, the class of each sample *x* is computed by following the sign of the corresponding decision function applied to x. Subsequently, if the sign says *x* is in class *i*, the vote for the *i*-th class is incremented by one; conversely, the vote for class *j* is increased by unity. Finally, *x* is taken to belong to the class with the largest vote. In case two classes have an identical number of votes, the one with the smaller index is selected.

Decision Directed Acyclic Graph

After the hyperplanes of the $\frac{k(k-1)}{2}$ SVMs are discovered, the following graph system is used to determine the class for a test sample *x* (Figure 4.1). Each node of the graph has a list of classes attached and considers the first and last elements of the list. The list that corresponds to the root node contains all k classes. When a test instance x is evaluated, one descends from node to node, in other words, eliminates one class from each corresponding list, until the leaves are reached.

The mechanism starts at the root node which considers the first and last classes. At each node, i vs j, we refer to the SVM that was trained on data from classes i and j. The class of *x* is computed by following the sign of the corresponding decision function applied to x. Subsequently, if the sign says *x* is in class *i*, the node is exited via the right edge; conversely, we exit through the left edge. We thus eliminate the wrong class from the list and proceed via the corresponding edge to test the first and last classes of the new list and node. The class is given by the leaf that x eventually reaches.

4.6 Support Vector Regression

For reasons of generality and similarity to training within the classification case, we will directly consider the situation of linear support vector regression with slack variables. Additionally, from the same striking resemblance to support vector classification, computations are only sketched .

Recall that the primal optimization problem is defined as (4.10) :

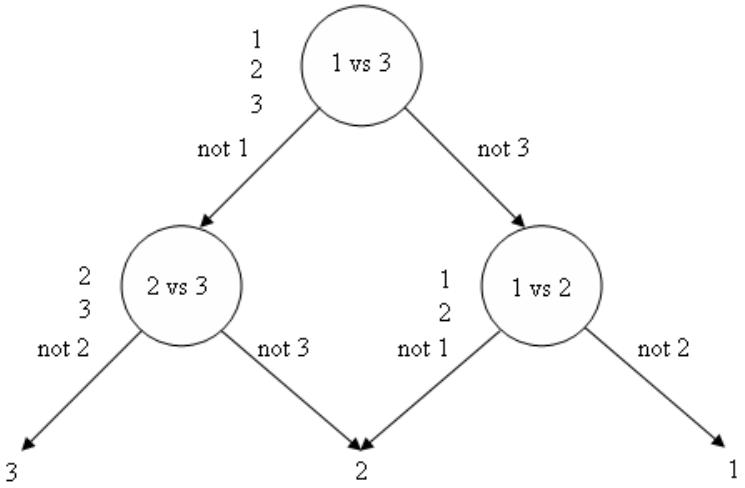

Figure 4.1: DDAG for labelling a test sample in three-class problems.

$$\begin{cases} \text{find } w \text{ and } b \text{ as to minimize } \|w\|^2 + C\sum_{i=1}^{m}(\xi_i + \xi_i^*) \\ \text{subject to } \begin{cases} y_i - \langle w, x_i\rangle + b \leq \epsilon + \xi_i \\ \langle w, x_i\rangle - b - y_i \leq \epsilon + \xi_i^* \quad , i = 1, 2, ..., m \\ \xi_i, \xi_i^* \geq 0 \end{cases} \end{cases} \tag{4.10}$$

The Lagrangian function is formulated as [Smola and Scholkopf, 1998]:

$$L(w, b, \xi, \alpha, \mu) = \tfrac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}(\xi_i + \xi_i^*) - \sum_{i=1}^{m}\alpha_i\left[\epsilon - y_i + \langle w, x_i\rangle - b + \xi_i\right] -$$

$$\sum_{i=1}^{m}\alpha_i^*\left[\epsilon + y_i - \langle w, x_i\rangle + b + \xi_i^*\right] - \sum_{i=1}^{m}(\mu_i\xi_i + \mu_i^*\xi_i^*),$$

where the variables $\alpha_i(^*)$ and $\mu_i(^*)$ are the Lagrange multipliers.

Remark 6. *For simplicity reasons, we will use* $(^*)$ *to refer to both the regular and the starred notations.*

Application of the KKTL conditions yields [Smola and Scholkopf, 1998]:

$$\frac{\partial L(w, b, \xi, \alpha, \mu)}{\partial w} = w - \sum_{i=1}^{m}(\alpha_i - \alpha_i^*)x_i = 0$$

$$\frac{\partial L(w, b, \xi(^*), \alpha(^*), \mu(^*))}{\partial b} = \sum_{i=1}^{m}(\alpha_i^* - \alpha_i) = 0$$

$$\frac{\partial L(w,b,\xi(^*),\alpha(^*),\mu(^*))}{\partial \xi_i(^*)} = C - \alpha_i(^*) - \mu_i(^*) = 0, i = 1,2,..,m$$

$$\alpha_i(^*) \geq 0, i = 1,2,..,m$$

$$\mu_i(^*) \geq 0, i = 1,2,..,m$$

After substitutions, the following dual formulation is obtained :

$$\begin{gathered}\text{find } \{\alpha_i(^*)\}_{i=1,2,..,m} \text{ as to maximize} \\ Q(\alpha(^*)) = \sum_{i,j=1}^{m}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\left\langle x_i, x_j\right\rangle \\ -\epsilon\sum_{i=1}^{m}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{m} y_i(\alpha_i - \alpha_i^*) \\ \text{subject to } \begin{cases} \sum_{i=1}^{m}(\alpha_i - \alpha_i^*) = 0 \\ 0 \leq \alpha_i(^*) \leq C, i = 1,2,...,m \end{cases}\end{gathered}$$

The optimum value for *w* is computed as before and is:

$$w = \sum_{i=1}^{m}(\alpha_i - \alpha_i^*)x_i,$$

while b can be determined as the mean of:

$$b = \epsilon - y_i + \sum_{j=1}^{m}(\alpha_j - \alpha_j^*)\left\langle x_j, x_i\right\rangle + \xi_i, i = 1,2,...,m,$$

when $0 < \alpha_i(^*) < C$.

Finally, in the most general case of nonlinear regression, the dual problem is restated as:

$$\begin{gathered}\text{find } \{\alpha_i(^*)\}_{i=1,2,..,m} \text{ as to maximize} \\ Q(\alpha(^*)) = \sum_{i,j=1}^{m}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)K(x_i, x_j) \\ -\epsilon\sum_{i=1}^{m}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{m} y_i(\alpha_i - \alpha_i^*) \\ \text{subject to } \begin{cases} \sum_{i=1}^{m}(\alpha_i - \alpha_i^*) = 0 \\ 0 \leq \alpha_i(^*) \leq C, i = 1,2,...,m \end{cases}\end{gathered}$$

and the predicted target for a test sample follows from:

$$y = f(x) = \sum_{i=1}^{m}(\alpha_i - \alpha_i^*)K(x, x_i) - b,$$

where b is computed in:

$$b = \epsilon - y_i + \sum_{j=1}^{m}(\alpha_j - \alpha_j^*)K(x_j, x_i) + \xi_i, i = 1, 2, ..., m.$$

Finally, in order to verify the accuracy of the technique, the value of the root mean square error (RMSE) is computed as in (4.11) :

$$RMSE = \sqrt{\frac{1}{p}\sum_{i=1}^{p}(y_i^{(pred)} - y_i)^2} \tag{4.11}$$

4.7 Remarks

Although very efficient, the original SVM training remains too elaborated and subject to numerous constraints and convergence requirements that stand in the way of possibly better performing, although nonstandard, decision functions. It is the purpose of this thesis to bring forward a simpler and unbound alternative that uses a direct search strategy for training.

Chapter 5

An Evolutionary Resemblant of Support Vector Machines

5.1 Aims of This Chapter

Evolutionary optimization allows the adaptation of the decision hyperplane to the available training data, which therefore can be treated directly as the (primal) optimization problem . The basic geometric idea of SVMs is considered, but the proposed approach deviates from the standard mathematical treatment. Additionally, the suggested evolutionary technique opens the way for generalizations involving nonlinear, nonstandard decision surfaces .

A new resembling approach is thus proposed; learning proceeds as in standard SVMs , while the optimal values for the coefficients of the hyperplane (w and b) are directly determined by an EA with respect to the equilibrium between accuracy and generalization ability [Stoean et al., 2006d], [Stoean et al., 2007a], [Stoean et al., 2007b], [Stoean and Dumitrescu, 2005b], [Stoean et al., 2006e]. Although the evolutionary resembling support vector machines (ERSVMs) appear to be straightforward, determining several other technique details, such as interpretation, operators and parameters, is not.

The chapter is structured as follows. Section 5.2 describes the existing evolutionary approaches towards enhancing the performance of the classical SVM architecture. Section 5.3 introduces the proposed evolutionary training together with a reformulation of the primal problem for the most general case of nonlinear learning within the context of the envisaged evolution. Section 5.4 puts forward a first considered constitution of an EA inside proposed ERSVMs. Validation is

achieved on real world examples. An alternative version of the ERSVM which is endowed with a new mechanism for reducing problem size in case of large data sets is additionally presented in subsection 5.4.9.

5.2 Previous Evolutionary Interactions with Support Vector Machines

Note that this is not the first attempt to bring SVMs and EAs close together. Existing alternatives are numerous and recent, of which some are presented further on. Strict hybridization towards different purposes is envisaged through: model and feature selection, kernel evolution and evolutionary detection of the Lagrange multipliers. Model selection concerns adjustment of hyperparameters (free parameters) within SVMs, i.e. the penalty for errors C and parameters of the kernel which, in standard variants, is performed through grid search or gradient descent methods. Evolution of hyperparameters can be achieved through evolution strategies [Friedrichs and Igel, 2004]. When dealing with high dimensional problems, feature selection regards the choice of the most relevant features as input for a SVM. The optimal subset of features can be evolved using genetic algorithms [de Souza et al., 2005] and genetic programming [Eads et al., 2002]. Evolution of kernel functions to model training data is performed by means of genetic programming [Howley and Madden, 2004]. Finally, the Lagrange multipliers involved in the expression of the dual problem can be evolved by means of evolution strategies and particle swarm optimization [Mierswa, 2006a]. Inspired by the geometrical SVM learning, [Jun and Oh, 2006] reports the evolution of w and C while using erroneous learning ratio and lift values as the objective function. The thesis, however, focuses on the evolution of the coefficients of the decision function in learning resemblance to SVMs; to the best of our knowledge, this has not been accomplished yet.

5.3 Proposed Evolutionary Resembling Support Vector Machines

The alternative concept can be modeled through the following evolutionary algorithm.

5.3.1 Representation

An individual encodes the coefficients of the hyperplane, w and b .

After termination of the EA, the approximately optimal values for the coefficients of the decision hyperplane are obtained.

5.3.2 Initial population

Individuals are randomly generated such that $w_i \in [-1, 1], i = 1, 2, ..., n$, $b \in [-1, 1]$.

5.3.3 Reformulation of the Primal Optimization Problem

Prior to decide upon the manner to evaluate individuals, the objective function must be established.

Since the proposed method departs from the standard SVMs, a different way to move into a higher dimensional space must be found [Stoean et al., 2007b] . Accordingly, w is also mapped through Φ into H. As a result, the squared norm that is involved in the generalization condition is now $\|\Phi(w)\|^2$. Also, the equation of the hyperplane consequently changes to (5.1):

$$\langle \Phi(w), \Phi(x_i) \rangle - b = 0. \tag{5.1}$$

We employ the scalar product in the form (5.2):

$$\langle u, w \rangle = u^T w, \tag{5.2}$$

and we additionally force the usage of the kernel to also transform the norm in its simplistic equivalence to the scalar product.

In conclusion, we reformulate the general primal form to further consider learning in the feature space and with the use of a kernel function. The classification task then becomes (5.3) :

$$\begin{cases} \text{find } w \text{ and } b \text{ as to minimize } K(w, w) + C \sum_{i=1}^{m} \xi_i, \\ C > 0 \\ \text{subject to } y_i(K(w, x_i) - b) \geq 1 - \xi_i, \xi_i \geq 0, \\ i = 1, 2, ..., m, \end{cases} \tag{5.3}$$

while the regression one is transposed to (5.4) :

$$\begin{cases} \text{find } w \text{ and } b \text{ as to minimize } K(w,w) + C\sum_{i=1}^{m}(\xi_i + \xi_i^*) \\ \text{subject to } \begin{cases} y_i - K(w,x_i) + b \leq \epsilon + \xi_i \\ K(w,x_i) - b - y_i \leq \epsilon + \xi_i^* \quad , i = 1,2,...,m \\ \xi_i, \xi_i^* \geq 0 \end{cases} \end{cases} \tag{5.4}$$

5.3.4 Multi-class Reconsideration

The ERSVMs targeting multi-class situations must undergo similar transformation with respect to the expression of the optimization problem [Stoean et al., 2008b], [Stoean et al., 2006a], [Stoean et al., 2006f]. As 1aa is concerned, the aim of every i^{th} ERSVM is expressed as to determine the optimal coefficients, w and b, of the decision hyperplane which best separates the samples with outcome i from all the other samples in the training set, such that (5.5) :

$$\begin{cases} \text{minimize } K(w^i,w^i) + C\sum_{j=1}^{m}\xi_j^i, \\ \text{subject to } y_j(K(w^i,x_j) - b) \geq 1 - \xi_j^i, \\ \xi_j^i \geq 0 \\ j = 1,2,...,m, i = 1,2,...,k. \end{cases} \tag{5.5}$$

Within the 1a1 and DDAG approaches, the aim of every ERSVM becomes to find the optimal coefficients w and b of the decision hyperplane which best separates the samples with outcome i from the samples with outcome j, such that (5.6) :

$$\begin{cases} \text{minimize } K(w^{ij},w^{ij}) + C\sum_{l=1}^{m}\xi_l^{ij}, \\ \text{subject to } y_l(K(w^{ij},x_l) - b) \geq 1 - \xi_l^{ij}, \\ \xi_l^{ij} \geq 0 \\ l = 1,2,...,m, i,j = 1,2,...,k, i \neq j. \end{cases} \tag{5.6}$$

5.3.5 Fitness assignment

The fitness assignment derives from the objective function of the optimization problem and is minimized. Constraints are handled by penalizing the infeasible individuals through appointing

$t : R \rightarrow R$ which returns the value of the argument, if negative, while zero otherwise.

Classification and regression variants simply differ in terms of objectives and constraints, thus the expression of the fitness function for the former is as follows (5.7) [Stoean et al., 2006b], [Stoean and Dumitrescu, 2005a], [Stoean and Dumitrescu, 2006], [Stoean and Dumitrescu, 2005c] :

$$f(w, b, \xi) = K(w, w) + C \sum_{i=1}^{m} \xi_i + \sum_{i=1}^{m} [t(y_i(K(w, x_i) - b) - 1 + \xi_i)]^2, \tag{5.7}$$

while is defined for the latter in the form (5.8) [Stoean et al., 2006c], [Stoean et al., 2006d], [Stoean, 2006] :

$$\begin{aligned} f(w, b, \xi) = K(w, w) + C \sum_{i=1}^{m} (\xi_i + \xi_i^*) + \sum_{i=1}^{m} [t(\epsilon + \xi_i - y_i + \\ K(w, x_i) - b)]^2 + \sum_{i=1}^{m} [t(\epsilon + \xi_i^* + y_i - K(w, x_i) + b)]^2, \end{aligned} \tag{5.8}$$

5.3.6 Stop condition

The EA stops after a predefined number of generations .

5.3.7 Test step

As the coefficients of the hyperplane are found, the target for a new, unseen test data sample can be determined directly following (5.9) for classification :

$$class(x) = sgn(K(w, x) - b) \tag{5.9}$$

or (5.10) for regression :

$$y = f(x) = K(w, x) - b. \tag{5.10}$$

As for the multi-class tasks, the label is found by employing the same mechanisms, only this time the resulting decision function applied to the current sample takes the form :

$$f(x) = K(w^i, x) - b^i.$$

5.4 A Naïve Construction - a Proposal

We target at attaining a suitable EA for solving the general primal problem of finding the decision hyperplane .

5.4.1 Research question

In achieving the proposed task, the following issue had to be addressed. How does one treat slack variables in the optimization problem? Can one depart from the SVM geometrical strict meaning of a deviation and simply evolve the factors of indicators for errors?

5.4.2 The Naive Evolutionary Algorithm

The particularities of the proposed construction are further presented.

Representation

As already stated, the coefficients of the hyperplane, w and b, are encoded into the structure of an individual. Since indicators for errors , ξ_i, $i = 1, 2, ..., m$, appear in the conditions for hyperplane optimality, ERSVM may handle them through inclusion in the structure of an individual, as well (5.11) :

$$c = (w_1, ..., w_n, b, \xi_1, ..., \xi_m). \tag{5.11}$$

After termination of the EA, the approximately optimal values for the coefficients of the decision hyperplane are obtained.

Initial population

The genes denoting indicators for errors in the individuals of the initial population are randomly generated again such that $\xi_j \in [0, 1], j = 1, 2, ..., m$.

Fitness assignment

The evolved values of the indicators for errors can now be addressed in the proposed expression for fitness evaluation.

Selection and variation operators

Mutation of errors is constrained, preventing the ξ_is from taking negative values .

5.4.3 Preexperimental Planning

First experiments have been conducted on five data sets (with no missing values) concerning real-world problems, coming from the UCI Repository of Machine Learning Databases[1], i.e. diabetes mellitus diagnosis [Stoean et al., 2006i] , spam detection [Stoean et al., 2006g] , [Stoean et al., 2007c], iris recognition [Stoean et al., 2008b] , soybean disease diagnosis [Stoean et al., 2006h] and Boston housing [Stoean et al., 2006c], [Stoean et al., 2006d] . The motivation for the choice of test cases was manifold. Diabetes and spam are two-class problems, while soybean and iris are multi-class. Differentiating, on the one hand, diabetes diagnosis is a better-known benchmark, but spam filtering is an issue of current major concern; moreover, the latter has a lot more features and samples, which makes a huge difference for classification as well as for optimization. On the other hand, while soybean has a high number of attributes, iris has only four, but a larger number of samples. Finally, Boston housing is a representative regression task. For all reasons mentioned above, the selection of test problems certainly contains all the variety of situations that is necessary for the objective validation of the new approach of ERSVM. Brief information on the tasks is given in Table 5.1, while details are provided in chapter 7.

Table 5.1: Data set properties.

	Diabetes	**Iris**	**Soybean**	**Spam**	**Boston**
Data					
Number of samples	768	150	47	4601	506
Number of attributes	8	4	35	57	13
Number of classes	2	3	4	2	-

[1] Available at http://www.ics.uci.edu/∼mlearn/MLRepository.html

5.4.4 Task

It is desired to be evaluated whether the suggested ERSVM with evolved errors produces a viable learning machine when compared to the standard approach (results are given in section 6.11) and how to determine appropriate parameters for the EA.

5.4.5 Evolutionary Algorithm Setup

Operators were chosen experimentally . Tournament selection, intermediate crossover and mutation with normal perturbation are applied.

5.4.6 Problem Setup

For each data set, 30 runs of the ERSVM were conducted; in every run approximately 70% random cases were appointed to the training set and the remaining 30% went into test. Experiments showed the necessity for data normalization in diabetes, spam and iris. No further modification of the data was carried out and all data was used in the experiments.

SVM parameters were manually chosen and can be found in ERSVMs section of Table 5.2. The error penalty C was invariably set to 1. For certain (e.g. radial, polynomial) kernels, the optimization problem shall be relatively simple, due to Mercer's theorem, and is implicitly solved by classical SVMs [Mierswa, 2006b]. Note that ERSVMs are not restricted to using the traditional kernels, but we employ them here to enable comparison with classical SVMs.

Table 5.2: Manually tuned SVM parameter values for the evolutionary and canonical approach.

	Diabetes	**Iris 1a1/1aa**	**Soybean**	**Spam**	**Boston**
ERSVMs					
p or σ	$p = 2$	$\sigma = 1$	$p = 1$	$p = 1$	$p = 1$
SVMs					
p or σ	$p = 1$	$\sigma = 1/m$	$p = 1$	$p = 1$	$\sigma = 1/m$

For the iris data set, a radial kernel was used, for diabetes, a polynomial one was employed, while for spam, soybean and Boston, we applied a linear surface. In the regression case, ϵ was set to 0.

Manually determined EA parameter values are given in Table 5.3.

Table 5.3: Manually tuned EA parameter values for the naïve construction.

	Diabetes	**Iris 1a1/1aa**	**Soybean**	**Spam**	**Boston**
p or σ	$p = 2$	$\sigma = 1$	$p = 1$	$p = 1$	$p = 1$
Population size	100	100/100	100	100	200
Generations	250	100/100	100	250	2000
Crossover prob.	0.40	0.30/0.70	0.30	0.30	0.50
Mutation prob.	0.40	0.50/0.50	0.50	0.50	0.50
ξ mutation prob.	0.50	0.50	0.50	0.50	0.50
Mutation strength	0.10	0.10/4	0.10	0.10	0.10
ξ mutation strength	0.10	0.10/0.10	0.10	0.10	0.10

In order to validate the manually found EA parameter values, the tuning method of Sequential Parameter Optimization (SPO) [Bartz-Beielstein, 2006] was applied. Parameter bounds were set as follows:

- Population size - 5/2000
- Number of generations - 50/300
- Crossover probability - 0.01/1
- Mutation probability - 0.01/1
- Error mutation probability - 0.01/1
- Mutation strength - 0.001/5
- Error mutation strength - 0.001/5

Since the three multi-class techniques behave similarly in all our manual multi-class experiments (Table 5.5), we run automatic tuning only for the most widely used case of 1a1. The best parameter configurations as determined by SPO are depicted in Table 5.4.

Table 5.4: SPO tuned EA parameter values for the naïve representation.

	Diabetes	Iris	Soybean	Spam	Spam +Chunks	Boston
Population size	198	46	162	154	90	89
Generations	296	220	293	287	286	1755
Crossover prob.	0.87	0.77	0.04	0.84	0.11	0.36
Mutation prob.	0.21	0.57	0.39	0.20	0.08	0.5
ξ mutation prob.	0.20	0.02	0.09	0.07	0.80	0.47
Mutation strength	4.11	4.04	0.16	3.32	0.98	0.51
ξ mutation strength	0.02	3.11	3.80	0.01	0.01	0.12

5.4.7 Results/Visualization

Test accuracies/errors obtained by manual tuning are presented in Table 5.5. Differentiated (spam/non spam for spam filtering and ill/healthy for diabetes) accuracies are also depicted.

A visualization of ERSVM for classification can be exhibited for simple artificial bidimensional data sets separated by various kernels (Figures 5.1, 5.2 and 5.3).

Illustration of ERSVM for regression is depicted in Figure 5.4, where the obtained function to be fitted to bidimensional data is drawn.

Table 5.6 holds performances and standard deviations of the best configuration of an initial Latin hypersquare design (LHS) sample and of the SPO.

5.4.8 Observations

SPO indicates that for all cases, except for the soybean data, crossover probabilities were dramatically increased, while often reducing mutation probabilities, especially for errors. However, the relative quality of SPO's final best configurations against the ones found during the initial LHS phase increases with the problem size. It must be stated that in most cases, results achieved with manually determined parameter values are only improved by SPO – if at all – by increasing effort (population size or number of generations).

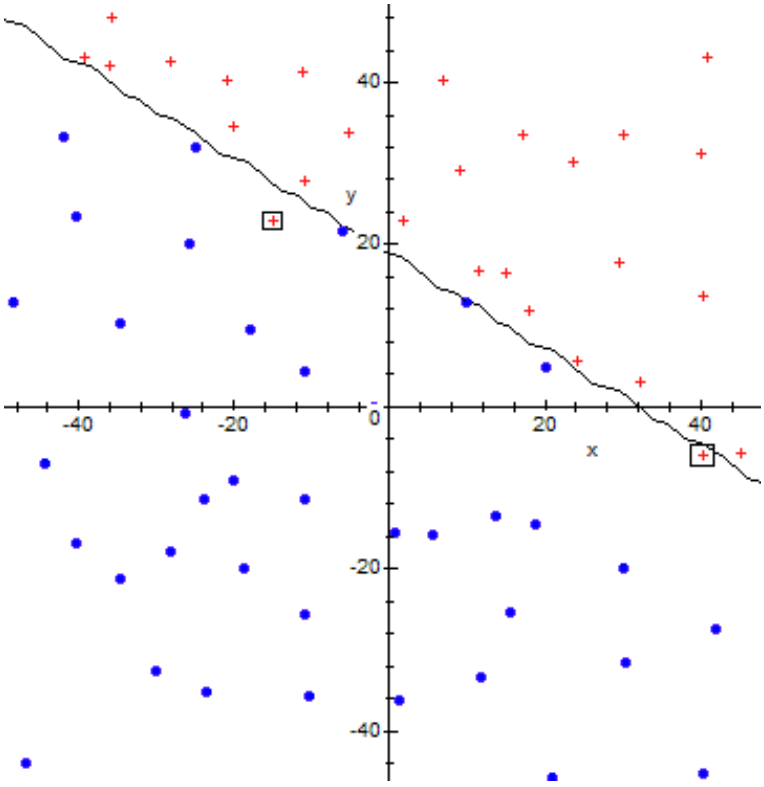

Figure 5.1: Visualization of naïve ERSVMs for classification on bidimensional data. Errors of classification are squared. An odd polynomial kernel is employed

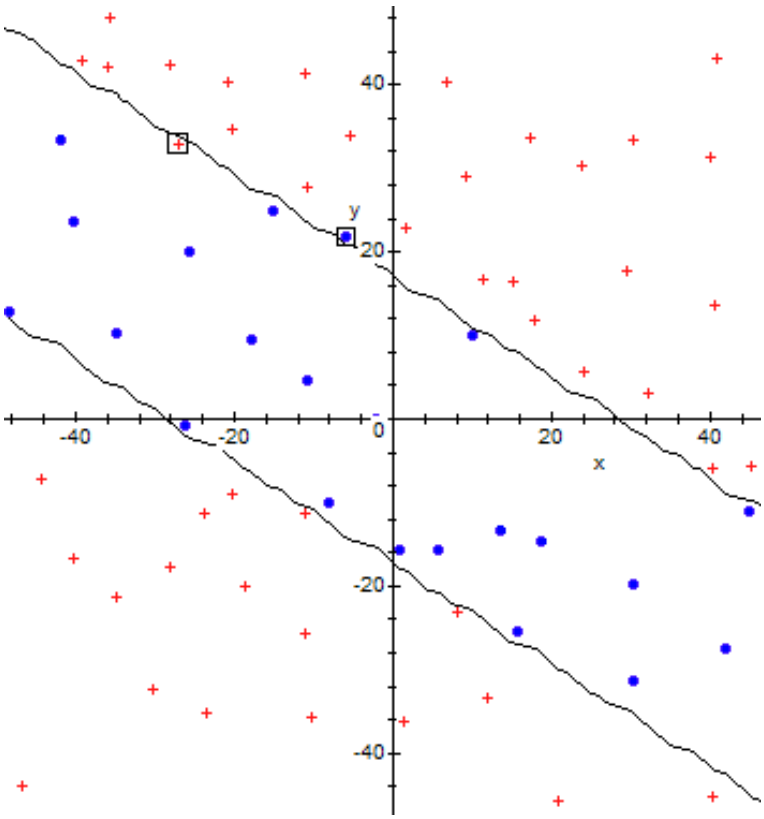

Figure 5.2: Visualization of naïve ERSVMs for classification on bidimensional data. Errors of classification are squared. An even polynomial kernel is employed

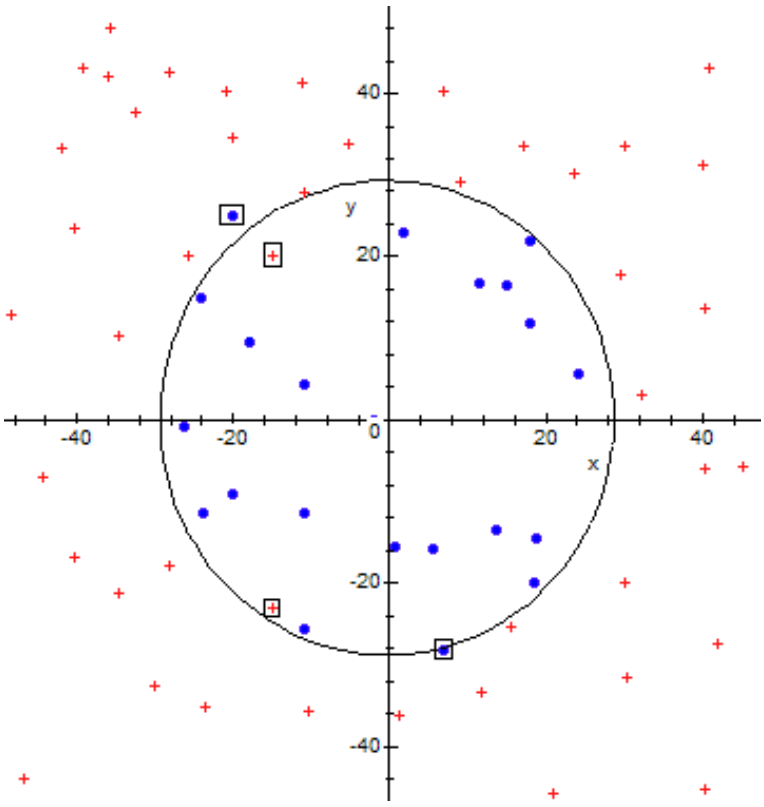

Figure 5.3: Visualization of naïve ERSVMs for classification on bidimensional data. Errors of classification are squared. A radial polynomial kernel is employed

Table 5.5: Accuracy/RMSE of the manually tuned naïve ERSVM version on the considered test sets, in percent.

	Average	Worst	Best	StD
Diabetes (overall)	**76.30**	71.35	80.73	2.24
Diabetes (ill)	50.81	39.19	60.27	4.53
Diabetes (healthy)	90.54	84.80	96.00	2.71
Iris 1aa (overall)	**95.85**	84.44	100.0	3.72
Iris 1a1 (overall)	**95.18**	91.11	100.0	2.48
Iris DDAG (overall)	**94.96**	88.89	100.0	2.79
Soybean 1aa (overall)	**99.22**	88.24	100	2.55
Soybean 1a1 (overall)	**99.02**	94.11	100.0	2.23
Soybean DDAG (overall)	**98.83**	70.58	100	5.44
Spam (overall)	**87.74**	85.74	89.83	1.06
Spam (spam)	77.48	70.31	82.50	2.77
Spam (non spam)	94.41	92.62	96.30	0.89
Boston	**4.78**	5.95	3.96	0.59

5.4.9 A Chunking Mechanism

A problem appears for large data sets, i.e. spam filtering, where the amount of runtime needed for training is very large. This stems from the large genomes employed, as indicators for errors of every sample in the training set are included in the representation. Consequently, we tackle this problem with an adaptation of a chunking procedure [Perez-Cruz et al., 2004] inside ERSVM.

A chunk of N training samples is repeatedly considered. Within each chunking cycle, the EA (with a population of half random individuals and half previously best evolved individuals) runs and determines the coefficients of the hyperplane. All training samples are tested against the obtained decision function and a new chunk is constructed based on N/2 randomly (equally distributed) incorrectly placed samples and half randomly samples from the current chunk. The chunking cycle stops when a predefined number of iterations with no improvement in training accuracy passes (Algorithm 2) .

ERSVM with chunking was applied to the spam data set. Manually tuned parameters had

Algorithm 2 ERSVM with Chunking

Require: The training samples

Ensure: Best obtained coefficients and corresponding accuracy

begin

Randomly choose N training samples set, equally distributed, to make a chunk;

while a predefined number of iterations passes with no improvement **do**

if first chunk **then**

Randomly initialize population of a new EA;

else

Use best evolved hyperplane coefficients and random indicators for errors to fill half of the population of a new EA and randomly initialize the other half;

end if

Apply EA and find coefficients of the hyperplane;

Compute side of all samples in the training set with evolved hyperplane coefficients;

From incorrectly placed, randomly choose (if exist) N/2 samples, equally distributed;

Randomly choose the rest up to N from the current chunk and add all to new;

if obtained training accuracy if higher than the best one obtained so far **then**

Update best accuracy and hyperplane coefficients; set improvement to true;

end if

end while

Apply best obtained coefficients on the test set and compute accuracy

return accuracy

end

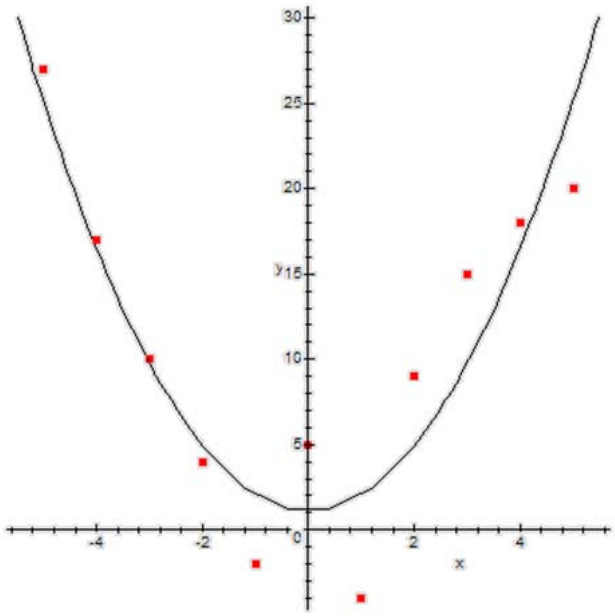

Figure 5.4: Visualization of naïve ERSVMs for regression on bidimensional data.

the same values as before, except the number of generations for each run of the EA which is now set to 100. The chunk size, i.e N, was chosen as 200 and the number of iterations with no improvement (repeats of the chunking cycle) was designated to be 5. Values derived from the SPO tuning are presented in the chunking column from Table 5.4.

Results of manual and SPO tuning are shown in Tables 5.7 and 5.8. The novel approach of ERSVM with chunking reached its goal, running 8 times faster than the previous one, at a cost of a small loss in accuracy. Besides solving the EA genome length problem, proposed mechanism additionally reduces the large number of computations that derives from the reference to the many training samples in the expression of the fitness function.

5.5 Remarks

The computational results have shown that the proposed technique produces good accuracies as compared to the canonical SVMs (see section 6.11). As concerns the difficulty in setting the EA, SPO confirms: Distinguishing the performance of different configurations is difficult even after computing a large number of repeats. Consequently, the ”parameter optimization potential”

Table 5.6: Accuracies of the SPO tuned naïve ERSVM version on the considered test sets, in percent.

	LHS$_{best}$	**StD**	**SPO**	**StD**
Diabetes (overall)	75.82	3.27	**77.31**	2.45
Diabetes (ill)	49.35	7.47	52.64	5.32
Diabetes (healthy)	89.60	2.36	90.21	2.64
Iris (overall)	**95.11**	2.95	**95.11**	2.95
Soybean (overall)	99.61	1.47	**99.80**	1.06
Spam (overall)	89.27	1.37	**91.04**	0.80
Spam (spam)	80.63	3.51	84.72	1.59
Spam (non spam)	94.82	0.94	95.10	0.81
Boston	5.41	0.65	**5.04**	0.52

Table 5.7: Accuracy/RMSE of the manually tuned ERSVM with chunking version on the considered test sets, in percent.

	Average	**Worst**	**Best**	**StD**
Spam (overall)	**87.30**	83.13	90.00	1.77
Spam (spam)	83.47	75.54	86.81	2.78
Spam (non spam)	89.78	84.22	92.52	2.11

justifies employing a tuning method only for the larger problems, diabetes, spam and Boston. Especially for the small problems, well performing parameter configurations are seemingly easy to find.

It must be stated that for the standard kernels, one cannot expect ERSVM to be better than standard SVMs, since it draws its roots from the latter. However, in future work, it can be profited from the flexibility of the EAs as optimization tools, by being able to additionally evolve kernels that achieve a better learning, regardless of whether they are positive (semi-)definite or not.

Table 5.8: Accuracies of the SPO tuned ERSVM with chunking version on the considered test sets, in percent.

	LHS$_{best}$	**StD**	**SPO**	**StD**
Spam (overall)	87.52	1.31	**88.37**	1.15
Spam (spam)	86.26	2.66	86.35	2.70
Spam (non spam)	88.33	2.48	89.68	2.06

Chapter 6

A Pruned Evolutionary Resemblant

6.1 Aims of This Chapter

Although already a viable alternative approach, the ERSVM may still be improved concerning simplicity . The current optimization problem requires to treat the error values, which in the present EA variant are included in the representation. These can be expected to severely complicate the problem by increasing the genome length (variable count) by the number of training samples. Moreover, such a methodology strongly drifts away from the canonical SVM concept. In this chapter, it is proposed to resolve this issue by investigating whether one can only represent the hyperplane coefficients and compute the indicators for errors instead of evolving them.

The chapter has the following structure. Section 6.2 brings a second, pruned representation of the proposed EA, which also speeds up runtime; a crowding variant is subsequently illustrated in subsection 6.8. An all-in-one evolution for the practical use is generalized in subsection 6.9. In order to validate the aim of this work, that is to offer a simpler, liberated and yet performing alternative to SVM training, section 6.11 exhibits the comparison with results of canonical SVMs implemented in R on the same data sets and in equivalent conditions.

6.2 The Pruned Evolutionary Algorithm

Since ERSVM directly and interactively provide hyperplane coefficients at all times, we propose to drop the indicators for errors from the EA representation and, instead, compute their values. Consequently, this time, individual representation contains only w and b (6.1) :

$$c = (w_1, ..., w_n, b). \tag{6.1}$$

Additionally, all indicators ξ_i, $i = 1, 2, ..., m$ will have to be computed in order to be referred in the fitness function.

In case of classification, the procedure follows [Bosch and Smith, 1998]. We take the current individual (which is the current separating hyperplane) and supporting hyperplanes are determined through the mechanism below. One first computes (6.2) :

$$\begin{cases} m_1 = min\{K(w, x_i)|y_i = +1\} \\ m_2 = max\{K(w, x_i)|y_i = -1\} \end{cases} \tag{6.2}$$

Then (6.3):

$$\begin{cases} p = |m_1 - m_2| \\ w' = \frac{2}{p}w \\ b' = \frac{1}{p}(m_1 + m_2) \end{cases} \tag{6.3}$$

For every training sample x_i, we obtain the deviation to its corresponding supporting hyperplane (6.4):

$$\delta(x_i) = \begin{cases} K(w', x_i) - b' - 1, y_i = +1, \\ K(w', x_i) - b' + 1, y_i = -1, \\ i = 1, 2, ..., m. \end{cases} \tag{6.4}$$

If sign of deviation equals class, corresponding $\xi_i = 0$; else, the (normalized) absolute deviation is returned as the indicator for error. Experiments showed the need for normalization of the computed deviations in the cases of diabetes, spam and iris, while, on the contrary, soybean requires without. The different behavior can be explained by the fact that the first three data sets have a larger number of training samples. The sum of deviations is subsequently added to the expression of the fitness function. As a consequence, in the early generations, when the generated coefficients lead to high deviations, their sum, considered from 1 to the number of training samples, takes over the whole fitness value and the evolutionary process is driven off the course to the optimum. The discussed diverse choices of actions concerning the normalization of data and errors and the kernel selection bring experimental evidence for the crucial importance of proper

data preparation prior to SVM learning. The form of the fitness function (5.7) remains as before, obviously without taking the ξ_is as arguments.

The method we propose for acquiring the errors for the regression situation is as follows. For every training sample, one firstly calculates the difference between the actual target and the predicted value that is obtained following the coefficients of the current individual (regression hyperplane), as in (6.5):

$$\delta_i = |K(w, x_i) - b - y_i|, i = 1, 2, ..., m \tag{6.5}$$

Secondly, one tests the difference against the ϵ threshold, following (6.6):

$$\begin{cases} \text{if } \delta_i < \epsilon \quad \text{then } \xi_i = 0, \\ \text{else} \qquad\quad \xi_i = \delta_i - \epsilon. \end{cases} i = 1, 2, ..., m \tag{6.6}$$

The newly obtained indicators for errors can now be employed in the fitness evaluation of the corresponding individual, which changes from (5.8) to (6.7):

$$f(w, b) = K(w, w) + C \sum_{i=1}^{m} \xi_i \tag{6.7}$$

The function to be fitted to the data is thus still required to be as flat as possible and to minimize the errors of regression that are higher than the permitted ϵ. Experiments on the Boston housing problem demonstrated that the specific method for computing the deviations does not require any additional normalization.

6.3 Preexperimental Planning

For all earlier mentioned reasons and for comparison between the two constructions, we keep the same data sets for application.

6.4 Task

It will be investigated if a representation without errors can perform as well as the naïve representation of section 5.4 [Stoean et al., 2007a], [Stoean et al., 2007b].

6.5 Problem Setup

The problem related settings and SVM parameters are kept the same as for naïve, except ϵ which is now set to 5 for the regression problem. This change tells that the pruned representation apparently needs a more generous ϵ allowance within training.

The EA proceeds with the manual values for parameters from Table 6.1.

Table 6.1: Manually tuned parameter values for the pruned approach.

	Diabetes	Iris	Soybean	Spam	Boston
Population size	100	100/100	100	150	200
Generations	250	100/100	100	300	2000
Crossover prob.	0.4	0.30/0.70	0.30	0.80	0.50
Mutation prob.	0.4	0.50/0.50	0.50	0.50	0.50
Mutation strength	0.1	4/4	0.1	3.5	0.1

Resulting parameter values for SPO on the pruned variant are shown in Table 6.2.

Table 6.2: SPO tuned parameter values for the pruned representation.

	Diabetes	Iris	Soybean	Spam	Boston
Population size	190	17	86	11	100
Generations	238	190	118	254	1454
Crossover prob.	0.13	0.99	0.26	0.06	0.88
Mutation prob.	0.58	0.89	0.97	0.03	0.39
Mutation strength	0.15	3.97	0.08	2.58	1.36

6.6 Results

Results obtained after manual and SPO tuning are depicted in Tables 6.3 and 6.4.

The automated performance values are generated by 30 validation runs for the best found configurations after initial design and SPO, respectively.

Table 6.3: Accuracy/RMSE of the manually tuned pruned ERSVM version on the considered test sets, in percent.

	Average	Worst	Best	StD
Diabetes (overall)	**74.60**	70.31	82.81	2.98
Diabetes(ill)	45.38	26.87	58.57	6.75
Diabetes (healthy)	89.99	86.89	96.75	2.66
Iris 1aa (overall)	**93.33**	86.67	100	3.83
Iris 1a1 (overall)	**95.11**	73.33	100	4.83
Iris DDAG (overall)	**95.11**	88.89	100	3.22
Soybean 1aa (overall)	**99.22**	88.24	100	2.98
Soybean 1a1 (overall)	**99.60**	94.12	100	1.49
Soybean DDAG (overall)	**99.60**	94.12	100	1.49
Spam (overall)	**85.68**	82	88.26	1.72
Spam (spam)	70.54	62.50	77.80	4.55
Spam (non spam)	95.39	92.66	97.44	1.09
Boston	**5.07**	6.28	3.95	0.59

6.7 Observations

Results of automated tuning came similar to those of manual regulation which brings once again evidence of the easy adjustability of the ERSVM. Additionally, we plotted the performance spectra of LHS to compare the hardness of finding good parameters for our two representations on the spam and soybean problems (Figures 6.1 and 6.2). The Y axis represents the fractions of all tried configurations; therefore the Y value corresponding to each bar denotes the percent of configurations that reached the accuracy of the X axis where the bar is positioned.

The plots illustrate the fact that naïve ERSVM is harder to parameterize than the pruned approach: When SPO finds a configuration for the latter, it is already a promising one, as it can be concluded from the higher corresponding bars.

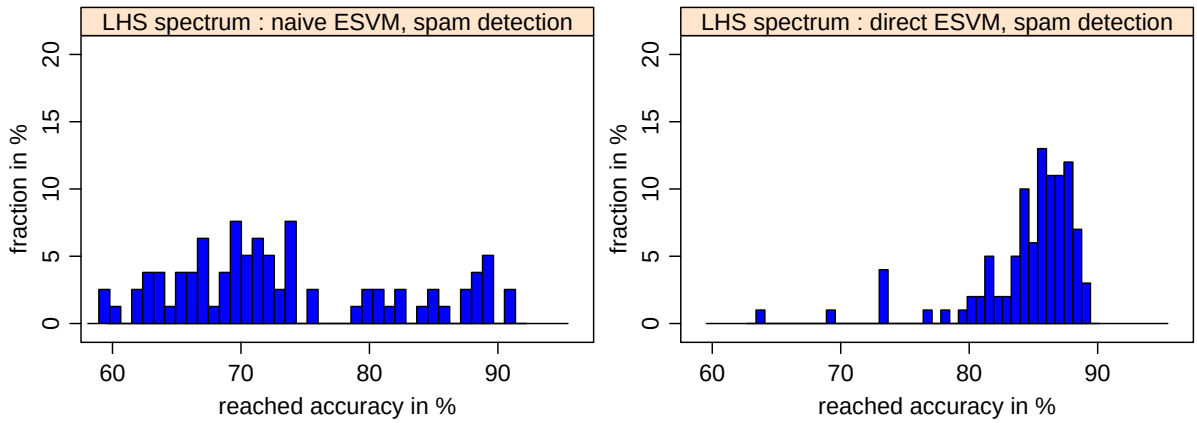

Figure 6.1: Comparison of EA parameter spectra, LHS with size 100, 4 repeats, for the naïve (left, 7 parameters) and the pruned (right, 5 parameters) representation on the spam problem.

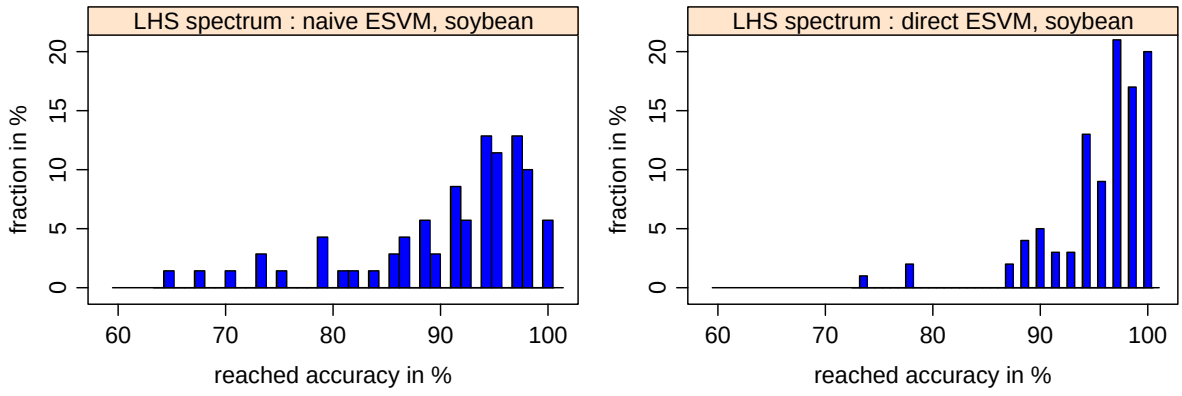

Figure 6.2: Comparison of EA parameter spectra, LHS with size 100, 4 repeats, for the naïve (left, 7 parameters) and the pruned (right, 5 parameters) representation on the soybean problem.

Table 6.4: Accuracies of the SPO tuned pruned ERSVM version on the considered test sets, in percent.

	LHS$_{best}$	**StD**	**SPO**	**StD**
Diabetes (overall)	72.50	2.64	**73.39**	2.82
Diabetes(ill)	35.50	10.14	43.20	6.53
Diabetes (healthy)	92.11	4.15	89.94	3.79
Iris (overall)	**95.41**	2.36	**95.41**	2.43
Soybean (overall)	**99.61**	1.47	99.02	4.32
Spam (overall)	89.20	1.16	**89.51**	1.17
Spam (spam)	79.19	3.13	82.02	3.85
Spam (non spam)	95.64	0.90	94.44	1.42
Boston	4.99	0.66	**4.83**	0.45

6.8 A Crowding Variant

In addition to the direct pruned representation, a crowding [DeJong, 1975] variant of the EA is also tested. Here, test for replacement is done against the most similar parent of the current population . Crowding based EAs are known to provide good global search capabilities. This is of limited value for the kernel types employed in this study, but it is important for nonstandard kernels. For now however, it is desired to investigate whether the employment of a crowding-based EA on the pruned representation would maintain the performance of the technique or not. All the other elements of the EA remain the same and the values for parameters as determined by SPO are shown in Table 6.5. The crowding experiment was chosen to be run only on the representative tasks for many samples (diabetes and spam), features (spam) and classes (iris).

Note that only automated tuning is performed for the pruned crowding ERSVM and results can be found in Table 6.6.

Automated tuning revealed that for the crowding variant, some parameter interactions dominate the best performing configurations: For larger population size, smaller mutation step sizes and larger crossover probability are better suited, and with greater run lengths, performance increases with larger mutation step sizes. For the original pruned variant, no such clear interactions can be attained. However, in both cases, many good configurations are detected.

Table 6.5: SPO tuned parameter values for the pruned representation with crowding.

	Diabetes	**Iris**	**Spam**
Population size	92	189	17
Generations	258	52	252
Crossover prob.	0.64	0.09	0.42
Mutation prob.	0.71	0.71	0.02
Mutation strength	0.20	0.20	4.05

Table 6.6: Accuracies of the SPO tuned pruned version with crowding on the considered test sets, in percent.

	LHD$_{best}$	**StD**	**SPO**	**StD**
Diabetes (overall)	74.34	2.30	**74.44**	2.98
Diabetes(ill)	43.68	6.64	45.32	7.04
Diabetes (healthy)	90.13	3.56	90.17	3.06
Iris (overall)	**95.63**	2.36	94.37	2.80
Spam (overall)	88.72	1.49	**89.45**	0.97
Spam (spam)	80.14	5.48	80.79	3.51
Spam (non spam)	94.25	1.66	95.07	1.20

6.9 An All-in-One Enhancement

In order to offer a complete solution for practical usage, it is further proposed to also include in the representation an immediate manner for a dynamic choice of model hyperparameters . Deriving from performed experiments, the parameter expressing the penalty for errors C seems of no significance within the ERSVM technique; it will consequently be dropped from the parameters pool. Further on, by simply inserting one more variable to the genome, the kernel parameter (p or $sigma$) can be also evolved. In this way, profiting from the evolutionary solving of the primal problem, model selection is actually performed at the same time.

The idea was tested through an immediate manual tuning of the EA parameters; the values are depicted in Table 6.7. For reasons of generality with respect to the new genomic variable, an

extra mutation probability and mutation strength respectively, were additionally appointed. The corresponding gene also had a continuous encoding, the hyperparameter being rounded at kernel application. The soybean task was not considered anymore for this experiment, as very good results were already provided for it.

Table 6.7: Manually tuned parameter values for all-in-one pruned representation.

	Diabetes	**Iris**	**Boston**	**Spam**
Population size	100	50	100	5
Generations	250	280	2000	480
Crossover prob.	0.4	0.9	0.5	0.1
Mutation prob.	0.4	0.9	0.5	0.1
Mutation prob. hyperparam.	0.4	0.9	0.9	0.1
Mutation strength	0.1	1	0.1	3.5
Mutation strength hyperparam.	0.5	0.1	0.1	0.1

The resulting values for the hyperparameters were identical to our previous manual choice (Table 5.2), with one exception in the diabetes task, where sometimes a linear kernel is obtained.

Results of the all-inclusive technique (Table 6.8), similar in accuracy/regression error to the prior ones and obtained at no additional cost, sustain the inclusion of model selection and point to the next step, the coevolution of nonstandard kernels.

6.10 Discussion

It is interesting to remark that the pruned representation is not that much faster. Although the genome length is drastically reduced, the runtime consequently gained is however partly lost again when computing the values for the slack variables. This draws from the extra number of dot products that must be calculated due to (6.2) and (6.5). As run length itself is a parameter in present studies, an upper bound of the necessary effort is rather obtained. Closer investigation may lead to a better understanding of suitable run lengths, e.g. in terms of fitness evaluations. However, the pruned representation has its advantages. Besides featuring smaller genomes, less parameters are needed, because the slack variables are not evolved and thus two parameters van-

Table 6.8: Accuracy/RMSE of the manually tuned all-in-one pruned ERSVM version on the considered test sets, in percent.

	Average	Worst	Best	StD
Diabetes (overall)	**74.20**	66.66	80.21	3.28
Diabetes(ill)	46.47	36.99	63.08	6.92
Diabetes (healthy)	89.23	81.40	94.62	3.46
Iris (overall)	**96.45**	93.33	100	1.71
Spam (overall)	**88.92**	85.39	91.48	1.5
Spam (spam)	79.98	68.72	94.67	5.47
Spam (non spam)	94.79	84.73	96.91	2.22
Boston	**5.06**	6.19	3.97	0.5

ish. As a consequence, it can be observed that this representation is easier to tune.

The best configurations for the pruned representation perform similarly and not significantly worse as compared to the results recorded for the naïve representation, except for the diabetes test case, where they are weaker. Parameter tuning beyond a large initial design appears to be infeasible, as performance is not significantly improved in most cases. If at all, it is successful for the larger problems of diabetes, spam and Boston. This indicates that parameter setting for the ERSVM is rather easy, because there is a large set of good performing configurations (Figure 6.1). Nevertheless, there seems to be a slight tendency towards fewer good configurations (harder tuning) for the large problems.

6.11 Evolutionary Resemblant versus Canonical Support Vector Learning

In order to make a direct comparison with the results of the canonical SVMs, we used the R environment and available related packages (e1071, mlbench and kernlab) for application to all considered data sets. The results, obtained after 30 runs, are illustrated in Table 6.9. After performing manual tuning for the SVM parameters, the best results were obtained as in the corresponding row of Table 5.2. It is worthy to note a couple of differences between our ERSVM and

the SVM implementation: In the Boston housing case, despite the employment of a linear kernel in the former, the latter produces better results for a radial function, while, in the diabetes task, the ERSVMs employ a degree two polynomial and SVMs target it linearly.

Table 6.9: Accuracy/RMSE of canonical SVMs on the considered test sets, in percent, as opposed to those obtained by ERSVM.

	∅SVM	StD	∅ERSVM	StD	p-value
Diabetes	**76.82**	1.84	**77.31**	2.45	$< 10^{-3}$
Iris	**95.33**	3.16	**95.63**	2.36	0.663
Spam	**92.67**	0.64	**91.04**	0.80	$< 10^{-3}$
Soybean	**92.22**	9.60	**99.80**	1.06	$< 10^{-3}$
Boston	**3.82**	0.7	**4.78**	0.59	$< 10^{-3}$

The results for each problem were compared via a Wilcoxon rank-sum test. The p-values (see Table 6.9) suggest to detect significant differences in all cases but the Iris data set. However, the absolute difference is not large for Diabetes (ERSVM better) and for Boston housing (SVM better), rendering each a slight advantage. It may be more relevant for the Spam problem (SVM better) and the Soybean task (ERSVM better).

Although, in terms of accuracy, our approach has not achieved better results for some of the test problems, it has many advantages: The decision surface is always transparent even when working with kernels whose underlying transformation to the feature space cannot be determined. The simplicity of the EA makes the solving process easily explained, understood, implemented and tuned for practical usage. Most importantly, any function can be used as a kernel and no additional constraints or verifications are necessary.

From the opposite perspective, the training is relatively slower than that of SVM, as the evaluation always relates to the training data. However, in practice (often, but not always), it is the test reaction that is more important. Nevertheless, by observing the relationship between each result and the corresponding size of the training data, it is clear that SVM performs better than ERSVM for larger problems; this is probably due to the fact that, in these cases, much more evolutionary effort would be necessary. The problem of handling large data sets is thus worth investigating in future work.

Chapter 7

Application of the Evolutionary Resembling Support Vector Machines to Real-World Problems

7.1 Aims of This Chapter

The intricate and vital tasks that arise in different practical domains are of great preoccupation to both practitioners and researchers . This chapter reinforces the reference to several real-world problems from various fields with the purpose of highlighting the applied side and success of the newly developed ERSVM. In order to fulfill the purpose of this chapter, comparison to other state-of-the-art techniques is conducted, even if the conditions of experimentation were not similar and thus not truly objective.

7.2 Pima Indians Diabetes

The diabetes data set was given by the Johns Hopkins University. Prior to that, the university selected cases from a larger database owned by the National Institute of Diabetes and Digestive and Kidney Diseases to create it. All patients in the data set are females of at least 21 years old, of Pima Indian heritage, living near Phoenix, Arizona, USA. There are 8 attributes (either discrete or continuous) containing personal data, e.g., age, number of pregnancies, and medical data, e.g., blood pressure, body mass index, result of glucose tolerance test etc (see Table 7.1).

Table 7.1: Description of attributes for each sample in the Pima Indians diabetes problem.

Attribute	Interval
Number of times pregnant	0-5
Plasma glucose concentration in an oral glucose tolerance test	[0,199]
Diastolic blood pressure	[0,122]
Triceps skin fold thickness	[0,99]
2-Hour serum insulin	[0,846]
Body mass index	[0,67]
Diabetes pedigree function	[0.078,2.42]
Age	21-81

The last attribute is a discrete one and it offers the diagnosis, which is either 0 (negative) or 1 (positive). 34.9% of the patients in the data set are assigned diabetes positive. The total number of cases is 768. The data is complete, according to its documentation; however, there are some 0 values of attributes that were not reported as missing data, but look peculiar.

ERSVMs resulted in 77.31% accuracy as opposed to SVMs that achieved a value of 76.82%. Obtained results were compared to others reported by literature, with respect to the diabetes diagnosis problem. In [Smith et al., 1988] a neural network algorithm to forecast the onset of diabetes mellitus was used. The mean of 5 runs of the best neural configuration was 75.12%. Using a neural networks heuristic, the mean accuracy in [Smithies et al., 2004] was obtained as 65.55%. A new evolutionary system to evolve artificial neural networks was proposed in [Yao and Liu, 1997] and the obtained mean accuracy on the test set was of 77.6%.

7.3 Spam Filtering

The data contains 4601 instances and 39.4% of them represent spam e-mails . For each e-mail in the data set there are 58 attributes, 57 of them of continuous nature while the last one is binary and represents the class. If the class of an e-mail is 0, then it is considered spam and, if the class is 1, it is seen as regular e-mail. For each e-mail, most of the attributes show the frequency with which some words or characters appear within it. The frequency of a word (or character) is computed by

the percentage of words (or characters) in the e-mail that match it exactly. Three of the attributes measure the length of sequences that contain only capital letters:

- One of them gives the average length of uninterrupted sequences of capital letters;
- Another one indicates the length of longest uninterrupted sequence of capital letters;
- The last one gives the total number of capital letters in the e-mail.

There are no missing attributes in the data set.

ERSVMs obtained an accuracy of 91.04% in comparison to SVMs that reached 92.67%. Literature reports accuracy on the UCI spam classification task in [Wang and Witten, 2002], where a SVM enhanced by random projections to reduce the dimensionality of data resulted in 88% accuracy. [Fradkin and Madigan, 2003] contains another boosting of a SVM by principal components analysis and presents a 92.3% efficiency. In [Tax, 2005], Naive Parzen achieved 76.8%, while k-Nearest Neighborhood reached only 66.5%.

7.4 Iris Classification

The Iris data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant, namely Iris Setosa, Iris Versicolour and Iris Virginica. There are 4 attributes referring to sepal length ([4.3, 7.9]), sepal width ([2, 4.4]), petal length ([1, 6.9]) and petal width ([0.1, 2.5]).

ERSVMs achieved a 95.63% accuracy value and the canonical SVMs reached 95.33%. In comparison, in [Veenman, 1996], a genetic programming model was tested on the Iris database and obtained 92.7%, while in [Yang and Honavar, 1991], some neural networks techniques (back-propagation algorithm and cascade-correlation algorithm) were applied in this direction and resulted in 93.2% and 92.6%, respectively.

7.5 Soybean Disease Diagnosis

The small soybean collection contains 47 plant instances, each recording 35 attributes plus one attribute reflecting the corresponding disease. There are no missing values. The first 35 attributes enclose information pertaining to the description of the environmental attributes (e.g. normality of air temperature, precipitation), the plant's global attributes (e.g. seed treatment, plant height)

and the local attributes (i.e. condition of leaves, stem, fruits - pods, seed and roots). A list of these attributes is given in Table 7.2 [Michalski and Chilausky, 1980]. The last attribute of every record shows the disease of the current soybean sample. There are 4 considered diseases: Canker (10 samples), rot (10 samples), Phytophtora rot (10 samples), Rhizoctonia root rot (17 samples).

Table 7.2: Description of attributes for each sample in the soybean disease problem.

Attribute	Possible Values
Time of occurrence	April, May, June, July, August, September, October
Plant stand	Normal, less than normal
Precipitation	Less than normal, normal, above normal
Temperature	Less than normal, normal, above normal
Occurrence of hail	Yes, no
Number years crop repeated	Different last year, same last year, same last 2 years,..., same last 7 years
Damaged area	Scattered, groups of plants in low areas, groups of plants in upland areas, whole fields
Severity	Minor, potentially severe, severe
Seed treatment	None, fungicide, other
Seed germination	90-100%, 80-89%, less than 80%
Plant height	Normal, abnormal
Condition of leaves	Normal, abnormal
Leaf spots - halos	Absent, with yellow hallows, without yellow hallows
Leaf spots - margin	With water soaked margin, without water soaked margin, does not apply
Leaf spot size	Less than 1/8”, greater than 1/8”, does not apply
Leaf shredding or shot holing	Absent, present

Attribute	Possible Values
Leaf malformation	Absent, present
Leaf mildew growth	Absent, upper surface, lower surface
Condition of stem	Normal, abnormal
Presence of lodging	Yes, no
Stem cankers	Absent, below soil line, at or slightly above soil line, above second node
Canker lesion color	Brown, dark brown or black, tan
Fungal fruiting body on stem	Absent, present
External decay	Absent, firm and dry, watery
Mycelium on stem	Absent, present
Internal discoloration	None, brown, black
Sclerotia - internal or external	Absent, present
Condition of fruits - pods	Normal, diseased, few present, does not apply
Fruit spots	Absent, colored, brown spots with black specks, distorted, does not apply
Condition of seed	Normal, abnormal
Mold growth	Absent, present
Seed discoloration	Absent, present
Seed size	Normal, less than normal
Seed shrivelling	Absent, present
Condition of roots	Normal, rotted, galls and cysts

While ERSVMs obtained 99.80% on the soybean task (Table 6.9), the standard SVMs reached a 92.22% accuracy. Comparison to other known techniques applied to the same problem was also conducted. Accuracy for the soybean disease diagnosis was reported by [Bailey et al., 2003], where classification with constraint emerging patterns and the Naive Bayes method were employed. The former method provided an accuracy of 95.50% (reaching 100% when a pair-wise classification strategy was employed) while the latter resulted in 98% accuracy.

7.6 Boston Housing

This regression task deals with the prediction of the median price of housing in 1970 in the Boston area based on socio-economic and environmental factors, such as crime rate, nitric oxide concentration, distance to employment centers and age of a property (Table 7.3). There are 506 samples, 13 continuous attributes (including the target attribute) and one binary-valued attribute. There are no missing values.

Table 7.3: Description of attributes for each sample in the Boston housing task.

Attribute	Interval
Crime rate	[0.00632, 88.9762]
Proportion of residential land zones	[0,100]
Proportion of non-retail business acres	[0.46,27.74]
Charles River dummy variable	[0,1]
Nitric oxides concentration	[0.385,0.871]
Average number of rooms	[3.561,8.78]
Proportion of owner occupied units built before 1940	[2.9,100]
Weighted distance to five Boston employment centers	[1.1296,12.126]
Index of accessibility to radial highways	[1,24]
Full value property tax rate	[187,711]
Pupil-teacher ratio	[12.6,22]
Proportion of African American population	[0.32,396.9]
Percentage of lower status of the population	[1.73,37.97]

ERSVMs provided a prediction error of 4.78, while SVMs exhibited a result of 3.82. Additionally, a linear regression model was implemented in R, with the same training/test set sizes and random manner of appointment, and reached an error of 4.76.

7.7 Remarks

The proposed ERSVMs prove to be very useful also from the practical point of view, as they are strong competitors to well-known and imposed techniques for classification and regression. In addition to this respect, apart from the functional perspective, the all-in-one enhancement is a simple and efficient tool in the hands of the specialists.

Chapter 8

Conclusions and Future Work

The evolutionary learning technique proposed in this thesis resembles the vision upon learning of SVMs but solves the inherent optimization problem by means of an EA.

8.1 Achievements

The thesis puts forward an easier and more flexible alternative to SVMs that undergoes several enhacements in order to truly stand as a worthy competitive substitute to the classical paradigm.

- Two possible representations for the EA (one simpler, and a little faster, and one more complicated, but also more accurate) that determines the coefficients are imagined.
- In order to boost the suitability of the new technique for any issue, a novel chunking mechanism for reducing size in large problems is also proposed; obtained results support its employment.
- The use of a crowding-based EA is inspected in relation to the preservation of the performance. Crowding would be highly necessary in the immediate coevolution of nonstandard kernels.
- Finally, an all-inclusive ERSVM construction for the practical perspective is developed and validated.
- On a different level, an additional aim was to address and solve real-work tasks of high importance, like disease diagnosis and prevention or spam filtering. The proposed ERSVM

tool manages to successfully provide a means of assisting the medical decision-making as well as a way to prevent loss of time and money in e-mail communication.

8.2 Remarks

Several conclusions can be eventually drawn and the potential of the technique can be further completed through a couple of enhancements.

- As opposed to SVMs, ERSVMs are definitely much easier to understand and use.
- ERSVMs do not impose any kind of constraints or requirements.
- Moreover, the evolutionary solving of the optimization problem enables the acquirement of function coefficients directly and at all times within a run.
- SVMs, on the other hand, are somewhat faster, as the kernel matrix is computed only once.
- Performances are comparable, for different test cases ERSVMs and SVMs take the lead, in turn.

8.3 Future Directions

Although already competitive, the novel ERSVM can still be enhanced in several ways:

- The requirement for an optimal decision function actually involves two criteria: the surface must fit to training data but simultaneously generalize well. So far, these two objectives are combined in a single fitness expression. As a better choice for handling these conditions, a multicriterial approach could be tried instead.
- Additionally, the simultaneous evolution of the hyperplane and of nonstandard kernels will be achieved. This approach is highly difficult by means of SVM standard methods for hyperplane determination, whereas it is straightforward for ERSVMs.
- The employment of the competitive ERSVM approach towards the solving of further practical problems is also desired.

Bibliography

[Altman, 1991] Altman, D. G. (1991). *Practical Statistics for Medical Research*. Chapman and Hall.

[Baeck, 1996] Baeck, T. (1996). *Evolutionary Algorithms in Theory and Practice*. Oxford University Press.

[Baeck et al., 1997] Baeck, T., Fogel, D. B., and Michalewicz, Z., editors (1997). *Handbook of Evolutionary Computation*. Institute of Physics Publishing and Oxford University Press.

[Bailey et al., 2003] Bailey, J., Manoukian, T., and Ramamohanarao, K. (2003). Classification using constrained emerging patterns. *Proc. of WAIM 2003, Lecture Notes in Computer Science Series*, pages 226–237.

[Bartz-Beielstein, 2006] Bartz-Beielstein, T. (2006). *Experimental research in evolutionary computation - the new experimentalism*. Natural Computing Series. Springer-Verlag.

[Bosch and Smith, 1998] Bosch, R. A. and Smith, J. A. (1998). Separating hyperplanes and the authorship of the disputed federalist papers. *American Mathematical Monthly*, 105(7):601–608.

[Boser et al., 1992] Boser, B. E., Guyon, I. M., and Vapnik, V. (1992). A training algorithm for optimal margin classifiers. *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 11–152.

[Burges, 1998] Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167.

[Cortes and Vapnik, 1995] Cortes, C. and Vapnik, V. (1995). Support vector networks. *Machine Learning*, pages 273–297.

[Courant and Hilbert, 1970] Courant, R. and Hilbert, D. (1970). *Methods of Mathematical Physics*. Wiley Interscience.

[Cover, 1965] Cover, T. M. (1965). Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers, vol. EC-14*, pages 326–334.

[Cristianini and Shawe-Taylor, 2000] Cristianini, N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines*. Cambridge University Press.

[de Souza et al., 2005] de Souza, B. F., de Leon, A. P., and de Carvalho, F. (2005). Gene selection based on multi-class support vector machines and genetic algorithms. *Journal of Genetics and Molecular Research*, 4(3):599–607.

[DeJong, 1975] DeJong, K. A. (1975). *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. PhD thesis, University of Michigan, Ann Arbor.

[Dumitrescu, 2000] Dumitrescu, D. (2000). *Genetic Algorithms and Evolution Strategies*. Blue Publishing House.

[Dumitrescu and Gorunescu, 2004] Dumitrescu, D. and Gorunescu, R. (2004). Evolutionary clustering using adaptive prototypes. *Studia Univ. Babes - Bolyai, Seria Informatica*, XLIX(1):15–20.

[Dumitrescu et al., 2000] Dumitrescu, D., Lazzerini, B., Jain, L. C., and Dumitrescu, A. (2000). *Evolutionary Computation*. CRC Press.

[Eads et al., 2002] Eads, D., Hill, D., Davis, S., Perkins, S., Ma, J., Porter, R., and Theiler, J. (2002). Genetic algorithms and support vector machines for time series classification. *Proc. Symposium on Optical Science and Technology*, pages 74–85.

[Eiben and Smith, 2003] Eiben, A. E. and Smith, J. E. (2003). *Introduction to Evolutionary Computing*. Springer-Verlag.

[Fletcher, 1987] Fletcher, R. (1987). *Practical Methods of Optimization*. Wiley.

[Fogel, 1995] Fogel, D. B. (1995). *Evolutionary Computation*. IEEE Press.

[Fradkin and Madigan, 2003] Fradkin, D. and Madigan, D. (2003). Experiments with random projections for machine learning. *Proc. of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 517–522.

[Friedrichs and Igel, 2004] Friedrichs, F. and Igel, C. (2004). Evolutionary tuning of multiple svm parameters. *Proc. 12th European Symposium on Artificial Neural Networks*, pages 519–524.

[Haykin, 1999] Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*. Prentice Hall.

[Holland, 1986] Holland, J. H. (1986). Escaping brittleness: The possibilities of general purpose learning algorithms applied to parallel rule-based systems. *Machine Learning*, 2:593–623.

[Howley and Madden, 2004] Howley, T. and Madden, M. G. (2004). The genetic evolution of kernels for support vector machine classifiers. *Proc. of 15th Irish Conference on Artificial Intelligence and Cognitive Science*. $http://www.it.nuigalway.ie/m\_madden/profile/pubs.html$.

[Hsu and Lin, 2004] Hsu, C.-W. and Lin, C.-J. (2004). A comparison of methods for multi-class support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425.

[Jun and Oh, 2006] Jun, S. H. and Oh, K. W. (2006). An evolutionary statistical learning theory. *International Journal of Computational Intelligence*, 3(3):249–256.

[Luchian et al., 1994] Luchian, S., Luchian, H., and Petriuc, M. (1994). Evolutionary automated classification. *Proc. of International Conference on Evolutionary Computation*, 2:585–588.

[Mercer, 1908] Mercer, J. (1908). Functions of positive and negative type and their connection with the theory of integral equations. *Transactions of the London Philosophical Society (A)*, 209:415 – 446.

[Michalewicz, 1992] Michalewicz, Z. (1992). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer - Verlag.

[Michalski and Chilausky, 1980] Michalski, R. and Chilausky, R. (1980). Learning by being told and learning from examples: an experimental comparison of the two methods of knowledge

acquisition in the context of developing an expert system for soybean disease diagnosis. *International Journal of Policy Analysis and Information Systems*, 4(2):125–160.

[Mierswa, 2006a] Mierswa, I. (2006a). Evolutionary learning with kernels: A generic solution for large margin problems. *Proc. of the Genetic and Evolutionary Computation Conference*, pages 1553–1560.

[Mierswa, 2006b] Mierswa, I. (2006b). Making indefinite kernel learning practical, technical report. Technical report, CArtificial Intelligence Unit, Department of Computer Science, University of Dortmund.

[Olafsson et al., 2006] Olafsson, S., Xiaonan, L., and Shuning, W. (in press, 2006). Operations research and data mining. *European Journal of Operational Research*. doi:10.1016/j.ejor.2006.09.023.

[Perez-Cruz et al., 2004] Perez-Cruz, F., Figueiras-Vidal, A. R., and Artes-Rodriguez, A. (2004). Double chunking for solving svms for very large datasets. *Proceedings of Learning 2004, Elche, Spain*. eprints.pascal-network.org/archive/00001184/01/learn04.pdf.

[Platt et al., 2000] Platt, J. C., Cristianini, N., and Shawe-Taylor, J. (2000). Large margin dags for multiclass classification. *Proc. of Neural Information Processing Systems*, pages 547–553.

[Rosipal, 2001] Rosipal, R. (2001). *Kernel-based Regression and Objective Nonlinear Measures to Access Brain Functioning*. PhD thesis, Applied Computational Intelligence Research Unit School of Information and Communications Technology University of Paisley, Scotland.

[Sarker et al., 2002] Sarker, R., Mohammadian, M., and Yao, X., editors (2002). *Evolutionary Optimization*. Kluwer Academic Pusblishers.

[Schwefel et al., 2003] Schwefel, H.-P., Wegener, I., and Weinert, K., editors (2003). *Advances in Computational Intelligence*. Springer-Verlag.

[Smith et al., 1988] Smith, J. W., Everhart, J. E., Dickson, W. C., Knowler, W. C., and Johannes, R. (1988). Using the adap learning algorithm to forecast the onset of diabetes mellitus. *Proc. of 12th Symposium on Computer Applications in Medical Care*, pages 261–265.

[Smithies et al., 2004] Smithies, R., Salhi, S., and Queen, N. (2004). Adaptive hybrid learning for neural networks. *Neural Computation*, 16(1):139–157.

[Smola and Scholkopf, 1998] Smola, A. J. and Scholkopf, B. (1998). A tutorial on support vector regression. Technical Report NC2-TR-1998-030, NeuroCOLT2 Technical Report Series.

[Stoean et al., 2008a] Stoean, C., Dumitrescu, D., Preuss, M., and Stoean, R. (in press, 2008a). Cooperative coevolution as a paradigm for classification. *Journal of Universal Computer Science*.

[Stoean et al., 2005] Stoean, C., Preuss, M., Gorunescu, R., and Dumitrescu, D. (2005). Elitist generational genetic chromodynamics - a new radii-based evolutionary algorithm for multimodal optimization. In *The 2005 IEEE Congress on Evolutionary Computation (CEC 2005)*, pages 1839–1846.

[Stoean, 2006] Stoean, R. (2006). An evolutionary support vector machines approach to regression. *Proc. of 6th International Conference on Artificial Intelligence and Digital Communications*, pages 54–61.

[Stoean and Dumitrescu, 2005a] Stoean, R. and Dumitrescu, D. (2005a). Evolutionary linear separating hyperplanes within support vector machines. *Scientific Bulletin, University of Pitesti, Mathematics and Computer Science Series*, 11:75–84.

[Stoean and Dumitrescu, 2005b] Stoean, R. and Dumitrescu, D. (2005b). Evolutionary support vector machines - a new hybridized learning technique. application to classification. Technical report, Department of Computer Science, Faculty of Mathematics and Computer Science, Babes-Bolyai University, http://www.cir.cs.ubbcluj.ro.

[Stoean and Dumitrescu, 2005c] Stoean, R. and Dumitrescu, D. (2005c). Evolutionary support vector machines - a new learning paradigm. the linear non-separable case. *Proc. of the Colocviul Academic Clujean de Informatica*, pages 15–20.

[Stoean and Dumitrescu, 2006] Stoean, R. and Dumitrescu, D. (2006). Linear evolutionary support vector machines for separable training data. *Annals of the University of Craiova, Mathematics and Computer Science Series*, 33:141–146.

[Stoean et al., 2006a] Stoean, R., Dumitrescu, D., Preuss, M., and Stoean, C. (2006a). Different techniques of multi-class evolutionary support vector machines. *Proc. of Bio-Inspired Computing: Theory and Applications*, pages 299–306.

[Stoean et al., 2008b] Stoean, R., Dumitrescu, D., Preuss, M., and Stoean, C. (in press, 2008b). Evolutionary support vector machines for classification with multiple outcomes. *Journal of Universal Computer Science*.

[Stoean et al., 2006b] Stoean, R., Dumitrescu, D., and Stoean, C. (2006b). Nonlinear evolutionary support vector machines. application to classification. *Studia Babes-Bolyai, Seria Informatica*, LI(1):3–12.

[Stoean et al., 2006c] Stoean, R., Preuss, M., Dumitrescu, D., and Stoean, C. (2006c). ϵ - evolutionary support vector regression. *Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2006*, pages 21–27.

[Stoean et al., 2006d] Stoean, R., Preuss, M., Dumitrescu, D., and Stoean, C. (2006d). Evolutionary support vector regression machines. *IEEE Postproc. of the 8th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, pages 330–335.

[Stoean et al., 2006e] Stoean, R., Preuss, M., Stoean, C., and Dumitrescu, D. (2006e). Evolutionary support vector machines and their application for classification. Technical Report CI-212/06, Collaborative Research Center on Computational Intelligence, University of Dortmund.

[Stoean et al., 2007a] Stoean, R., Preuss, M., Stoean, C., and Dumitrescu, D. (2007a). Concerning the potential of evolutionary support vector machines. *Proc. of the IEEE Congress on Evolutionary Computation*, pages 1436–1443.

[Stoean et al., 2007b] Stoean, R., Preuss, M., Stoean, C., El-Darzi, E., and Dumitrescu, D. (submitted, 2007b). An evolutionary resemblant to support vector machines for classification and regression. *Journal of the Operational Research Society*.

[Stoean et al., 2006f] Stoean, R., Stoean, C., Preuss, M., and Dumitrescu, D. (2006f). Evolutionary multi-class support vector machines for classification. *Proceedings of International Conference on Computers and Communications - ICCC 2006, Baile Felix Spa - Oradea, Romania*, pages 423–428.

[Stoean et al., 2006g] Stoean, R., Stoean, C., Preuss, M., and Dumitrescu, D. (2006g). Evolutionary support vector machines for spam filtering. *Proc. of RoEduNet IEEE International Conference*, pages 261–266.

[Stoean et al., 2006h] Stoean, R., Stoean, C., Preuss, M., and Dumitrescu, D. (2006h). Forecasting soybean diseases from symptoms by means of evolutionary support vector machines. *Phytologia Balcanica*, 12(3).

[Stoean et al., 2007c] Stoean, R., Stoean, C., Preuss, M., and Dumitrescu, D. (2007c). Evolutionary detection of separating hyperplanes in e-mail classification. *Acta Cibiniensis*, LV:41–46.

[Stoean et al., 2006i] Stoean, R., Stoean, C., Preuss, M., El-Darzi, E., and Dumitrescu, D. (2006i). Evolutionary support vector machines for diabetes mellitus diagnosis. *Proceedings of IEEE Intelligent Systems 2006, London, UK*, pages 182–187.

[Tax, 2005] Tax, D. M. J. (2005). Ddtools, the data description toolbox for matlab. http://www-ict.ewi.tudelft.nl/ davidt/occ/index.html.

[Trafalis and Gilbert, 2006] Trafalis, T. B. and Gilbert, R. C. (2006). Robust classification and regression using support vector machines. *European Journal of Operational Research*, 173:893–909.

[Vapnik, 1982] Vapnik, V. (1982). *Estimation of Dependences Based on Empirical Data*. Springer-Verlag.

[Vapnik, 1995a] Vapnik, V. (1995a). Inductive principles of statistics and learning theory. *Mathematical Perspectives on Neural Networks*.

[Vapnik, 1995b] Vapnik, V. (1995b). *The Nature of Statistical Learning Theory*. Springer.

[Vapnik, 2003] Vapnik, V. (2003). *Neural Networks for Intelligent Signal Processing*. World Scientific Publishing.

[Vapnik and Chervonenkis, 1968] Vapnik, V. and Chervonenkis, A. (1968). Uniform convergence of frequencies of occurence of events to their probabilities. *Dokl. Akad. Nauk SSSR*, 181:915 – 918.

[Vapnik and Chervonenkis, 1974] Vapnik, V. and Chervonenkis, A. (1974). *Theorie der Zeichenerkennung*. Akademie-Verlag.

[Veenman, 1996] Veenman, C. J. (1996). Positional genetic programming: Genetic algorithms with encoded tree structures. Master's thesis, Vrije Universiteit, Amsterdam.

[Wang and Witten, 2002] Wang, Y. and Witten, I. H. (2002). Modeling for optimal probability prediction. *Proc. of the Nineteenth International Conference on Machine Learning*, pages 650–657.

[Yang and Honavar, 1991] Yang, J. and Honavar, V. (1991). Experiments with the cascade-correlation algorithm. Technical Report 91-16, Iowa State University.

[Yao and Liu, 1997] Yao, X. and Liu, Y. (1997). A new evolutionary system for evolving artificial neural networks. *IEEE Transactions on Neural Networks*, 8(3):694–713.

Index