

ALGORITHMS FOR TRAINING LARGE-SCALE LINEAR PROGRAMMING  
SUPPORT VECTOR REGRESSION AND CLASSIFICATION

PABLO RIVAS PEREA

Department of Electrical and Computer Engineering

APPROVED:

---

Jose Gerardo Rosiles, Ph.D., Chair

---

Sergio D. Cabrera, Ph.D.

---

Wei Qian, Ph.D.

---

Miguel Arguez, Ph.D.

---

Patricia Witherspoon, Ph.D.  
Dean of the Graduate School

©Copyright

by

Pablo Rivas Perea

2011

*to my beautiful wife*

*NANCY*

*with tons of love*

PREVIEW

PREVIEW

ALGORITHMS FOR TRAINING LARGE-SCALE LINEAR PROGRAMMING  
SUPPORT VECTOR REGRESSION AND CLASSIFICATION

by  
PABLO RIVAS PEREA

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

Department of Electrical and Computer Engineering

THE UNIVERSITY OF TEXAS AT EL PASO

May 2011

UMI Number: 3457758

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent on the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



UMI 3457758

Copyright 2011 by ProQuest LLC.

All rights reserved. This edition of the work is protected against unauthorized copying under Title 17, United States Code.



ProQuest LLC.  
789 East Eisenhower Parkway  
P.O. Box 1346  
Ann Arbor, MI 48106 - 1346

# Acknowledgements

I would like to express my deep-felt gratitude to my advisor, Dr. Gerardo Rosiles of the Electrical and Computer Engineering Department at The University of Texas at El Paso, for his advice, encouragement, enduring patience, and constant support. He always answered my emails, even when it was 3:00 AM; he has so much energy, (*where does he get it?!*). He was always, *always* giving me his time, in spite of anything else that was going on. He modestly co-supported my conference expenses and always tried to get me everything I needed to complete this dissertation. His protective attitude towards me, his apprentice, kept me from losing focus on my dissertation. He never allowed me to panic at any time. I wish all students the honor of working with Dr. Rosiles.

I also wish to thank the other members of my committee, Dr. Sergio D. Cabrera and Dr. Wei Qian from the Electrical and Computer Engineering Department at The University of Texas at El Paso. Their suggestions, comments, and additional guidance were invaluable to the completion of this work. Particularly, Dr. Cabrera co-sponsored some of conferences at which I presented some of my research papers. Nonetheless, Dr. Qian was always willing to collaborate and support my research proposals and research papers. Both have been extremely helpful in many different ways so that I had everything I needed to complete this work.

Dr. Miguel Arguez, from the Department of Mathematical Sciences at The University of Texas at El Paso, was the one who brought the light into the darkness I was walking through. His courses in Numerical Optimization and Interior Point Methods for Linear Programming opened my eyes in a tremendous way. This was very helpful in understanding the magnitude of the contributions of this dissertation.

Thanks to Dr. James C. Tilton, from the Computational and Information Sciences and Technology Office (CISTO) at the NASA Goddard Space Flight Center (GSFC), Excellence in Information Science and Technology Award winner. His direction during my internship at NASA really helped me to forge new skills. I am grateful for his friendship, hospitality, patience, and understanding. It has been the best time of my life.

Dr. Elaine Fredericksen from the English Department at The University of Texas at El Paso, acted many times as my editor, even though it was not part of her job description. But more importantly, she led me to understand so many things about the English language, in such a way that now I feel very confident about writing almost anything in English. I also want to acknowledge the collaboration of my proof-reading group: Mary Reyna and Letty Moreno-Brown. And, of course, the helpful discussions of my content group: Juan L. Rios and Mazin M. Al-Zoubi.

Additionally, I want to thank The University of Texas at El Paso Electrical and Computer Engineering Department professors and staff for all their hard work and dedication, providing me the means to complete my degree and prepare for a career as an electrical and computer engineer. This includes (but certainly is not limited to) the following individuals:

Sukie Quezada-Vazquez and Linda Romero

They helped me big time when I was hopelessly lost and new in town. Their help and availability made the transition from México to the US, much better than I expected. Both know everything about everything in the ECE department; many thanks to both. Sukie, Linda, your efficiency and efficacy is a great inspiration for me.



Lorenzo Solis

He was always there, showing up at the office with his big smile. This man was an inspiration for me because of his responsibility and passion for work. Besides, he was the only one who showed an interest in me for who I am as a human being, and not just to ask me about MATLAB, Linux, DSP, or Programming. We always had very nice conversations: short, funny, encouraging, motivating, and unconditional. Mr. Solis is the best janitor at our university, in my humble opinion.

Nito Gumataotao

Thanks to Nito, “the computer guy” of the Unix Lab, for his patience. I am sorry that I had to be remotely logged-in at 15 computers at the same time in order to finish my experiments on time. Also, my apologies for taking over the Parallel Computing licenses of the ECE department. Anyway, I acknowledge his total support on every little technical detail I needed help with. I recall one time I asked Nito if he could install  $\LaTeX$ , Kile, change my computer’s hostname, and upgrade the gcc compiler, and by the time I finished talking, he said, “Done.” In summary, thanks to Nito for his extraterrestrial skills in Unix security and administration.

Thanks to my friend Dr. Juan de Dios Cota (a.k.a. Juanito) for his patience while sharing an office with me. I hope the best for him and his family. I also want to acknowledge Carlos Ramirez’ kindness since he provided the naive interior point solver code that was partially used in this dissertation.

I want to thank my dear wife for putting up with me during the development of this work with continuing, loving support, and without complaint. I do not have the words to express all my feelings here, only that I love you, Nancy!

Many thanks to my parents and my in-laws for their prayers. Their support was felt every day during these long years. I am deeply grateful for that.

Finally, I want to thank Charles and Barbara Horak for always being there when I needed the most. Without their support, I would not have been able to finish my doctoral studies at UTEP, literally speaking. They “adopted” me and my wife for the time we were at El Paso city. They were the closest thing to a family we had here in El Paso. May the Lord bless and keep them.

PREVIEW

This work was supported in part by the National Council for Science and Technology (CONACyT), Mexico, under grant 193324 / 303732, by the Texas Instruments Foundation Endowed Scholarship for graduate students, by the IEEE Computational Intelligence travel scholarship, and by the SEP-DGRI complementary scholarship. The partial support of the Electrical and Computer Engineering Department via teaching assistantships is also acknowledged.

PREVIEW

NOTE: The first draft of this dissertation was submitted to my Supervising Committee on March, 2011.

# Abstract

The main contribution of this dissertation is the development of a method to train a Support Vector Regression (SVR) model for the large-scale case where the number of training samples supersedes the computational resources. The proposed scheme consists of posing the SVR problem entirely as a Linear Programming (LP) problem and on the development of a sequential optimization method based on variables decomposition, constraints decomposition, and the use of primal-dual interior point methods. Experimental results demonstrate that the proposed approach has comparable performance with other SV-based classifiers. Particularly, experiments demonstrate that as the problem size increases, the sparser the solution becomes, and more computational efficiency can be gained in comparison with other methods. To reduce the LP-SVR training time, a method is developed that takes advantage of the fact that the support vectors (SVs) are likely to lie on the convex hull of each class. The algorithm uses the Mahalanobis distance from the class sample mean in order to rank each sample in the training set; then the samples with the largest distances are used as part of the initial working set. Experimental results show a reduction in the total training time as well as a significant decrease in the total iterations percentage. Results also suggest that using the speedup strategy, the SVs are found earlier in the learning process. Also, this research introduces a method to find the set of LP-SVR hyper-parameters; experimental results show that the algorithm provides hyper-parameters that minimize an estimate of the true test generalization error.

Finally, the SVR scheme shows state-of-the-art performance in various applications such as power load prediction forecasting, texture-based image segmentation, and classification of remotely sensed imagery. This demonstrates that the proposed learning scheme and the LP-SVR model are robust and efficient when compared with other methodologies for large-scale problems.

# Table of Contents

	Page
Acknowledgements . . . . .	v
Abstract . . . . .	x
Table of Contents . . . . .	xi
List of Tables . . . . .	xvii
List of Figures . . . . .	xix
List of Symbols . . . . .	xxviii
List of Acronyms . . . . .	xxxii
<b>Chapter</b>	
1 General Introduction . . . . .	1
1.1 Statistical Pattern Recognition . . . . .	1
1.2 Support Vector Learning and Kernel Functions . . . . .	3
1.3 Large-Scale Support Vector Regression Training . . . . .	5
1.4 Scope of This Dissertation . . . . .	8
1.4.1 Problems to be Addressed . . . . .	9
1.4.2 Research Questions . . . . .	10
1.4.3 Dissertation Statement . . . . .	10
1.5 Summary of Contributions . . . . .	10
1.6 Overview of Remaining Chapters . . . . .	11
2 A Review of Support Vector Regression . . . . .	13
2.1 Introduction . . . . .	13
2.2 Support Vector Regression Fundamental Principle . . . . .	15
2.3 Support Vector Regression Learning Methods . . . . .	21
2.4 Linear Programming Support Vector Regression . . . . .	23
2.4.1 SVR Learning . . . . .	24

2.4.2	$\nu$ -LPR . . . . .	25
2.4.3	LP-SVR with Modified LPC . . . . .	27
2.4.4	Wavelet Kernel-based LP-SVR . . . . .	28
2.4.5	$\ell_1$ -norm LP-SVR . . . . .	29
2.5	Current Trends and Open Problems . . . . .	30
3	Large-Scale Linear Programming Support Vector Regression . . . . .	32
3.1	Introduction . . . . .	32
3.2	Large-Scale LP-SVR Formulation . . . . .	36
3.2.1	Primal, Dual, and KKT Conditions . . . . .	36
3.2.2	Optimality and Sparseness . . . . .	39
3.3	LP-SVR Variable Decomposition . . . . .	42
3.4	LP-SVR Constraints Decomposition . . . . .	49
3.5	Convergence and Optimality Conditions . . . . .	53
3.6	Interior Points Convergence and Optimality . . . . .	56
3.7	Experimental Evaluation of LS LP-SVR . . . . .	58
3.7.1	Large-Scale Learning . . . . .	61
3.7.2	Solution Sparseness . . . . .	66
3.7.3	Computational Concerns . . . . .	68
3.8	Conclusion . . . . .	69
3.8.1	Large-Scale Learning . . . . .	69
3.8.2	Model Performance and Sparseness . . . . .	70
4	LP-SVR Training Speedup and Hyper-Parameters Estimation . . . . .	71
4.1	Introduction . . . . .	71
4.2	Within-Class Distances for Learning Speed Up . . . . .	72
4.2.1	Within-Class Mahalanobis distance and Class-Convex Hull . . . . .	75
4.3	Large-Scale LP-SVR Hyper-Parameter Estimation . . . . .	77
4.4	Model Parameters Selection Criteria: Error Functions . . . . .	80
4.4.1	Error Functions for Multi-Class and Two-Class Problems . . . . .	81

4.4.2	Regression Problems . . . . .	83
4.5	Adaptation of Newton Method . . . . .	84
4.5.1	Newton Method Adaptation . . . . .	88
4.5.2	Algorithm Discussion . . . . .	89
4.5.3	Stopping Criteria . . . . .	91
4.6	Experimental Results . . . . .	93
4.6.1	Learning Speedup . . . . .	93
4.6.2	Model Selection . . . . .	95
4.7	Computational Concerns . . . . .	100
4.7.1	Speedup by Sample Selection . . . . .	100
4.7.2	Model Selection . . . . .	102
4.8	Conclusion . . . . .	103
4.8.1	Speedup by Sample Selection . . . . .	103
4.8.2	Model Selection . . . . .	104
5	Power Load Prediction . . . . .	105
5.1	Introduction . . . . .	105
5.2	Dataset . . . . .	106
5.3	Training the Regression Models . . . . .	107
5.3.1	Feed-Forward Neural Network . . . . .	107
5.3.2	Bagged Regression Trees . . . . .	108
5.3.3	Large-Scale Support Vector Regression . . . . .	109
5.3.4	Linear Programming Support Vector Regression . . . . .	109
5.4	Experimental Results . . . . .	109
5.4.1	Experiment Design and Procedure . . . . .	109
5.4.2	Quantitative Results . . . . .	111
5.5	Conclusion . . . . .	116
6	Texture Classification with LP-SVR and Biorthogonal Directional Filter Banks . . . . .	121

6.1	Introduction to Texture Analysis . . . . .	121
6.2	The Bamberger Directional Filter Bank . . . . .	123
6.3	Multiresolution Directional Pyramids . . . . .	127
6.3.1	Simple Lowpass-Highpass decomposition . . . . .	128
6.3.2	Directional Pyramids . . . . .	128
6.4	Feature Extraction . . . . .	133
6.5	Classification Stage . . . . .	136
6.6	Framework for Texture Segmentation with the DFB . . . . .	136
6.6.1	Proposed LP-SVR Architectures . . . . .	143
6.7	Evaluation of the Undecimated DP and LP-SVR for Texture Segmen- tation . . . . .	146
6.8	Conclusions on Texture Segmentation . . . . .	156
7	LP-SVR-Based Dust Storm Detection . . . . .	157
7.1	Introduction . . . . .	158
7.2	Feature Extraction . . . . .	159
7.3	Maximum <i>a Posteriori</i> Detection . . . . .	162
7.4	Probabilistic Neural Network Detection . . . . .	164
7.5	Feed-Forward Back-Propagation Neural Network Detection . . . . .	166
7.6	Linear Programming Support Vector Regression Detection . . . . .	168
7.7	Results . . . . .	168
7.7.1	Comparison between Methods . . . . .	169
7.7.2	Dust over Land . . . . .	171
7.7.3	Dust over Ocean . . . . .	172
7.7.4	Extension to Segmentation and Classification . . . . .	173
7.8	Conclusion . . . . .	174
8	Conclusions . . . . .	181
8.1	Contributions . . . . .	181
8.1.1	Large Scale LP-SVR . . . . .	181



8.1.2	LP-SVR Training Speedup and Hyper-Parameters Estimation	182
8.1.3	Applications . . . . .	183
8.2	Future Work . . . . .	186
8.2.1	Theoretical . . . . .	186
8.2.2	Experimental . . . . .	188
8.3	Publications . . . . .	189
8.3.1	Theoretical Development . . . . .	189
8.3.2	Applications . . . . .	190
	References . . . . .	191
<b>Appendix</b>		
A	Background in Support Vector Machines . . . . .	218
A.1	Support Vector Machines Fundamental Principle . . . . .	218
A.1.1	Soft-Margin Support Vector Machines . . . . .	221
A.1.2	Kernel Expansion and Kernel Functions . . . . .	223
A.1.3	Toy Example: XOR Problem . . . . .	226
A.2	Summary of Support Vector Machines Learning Methods . . . . .	230
A.2.1	Osuna's Decomposition Algorithm . . . . .	230
A.2.2	Joachims' SVM <sup>Light</sup> . . . . .	234
A.2.3	Platt's Sequential Minimal Optimization . . . . .	237
A.2.4	Rifkin's SVMFu . . . . .	239
A.2.5	Nystrom Method for Kernel Approximation . . . . .	241
A.2.6	Hush's QP SVM Dual to Primal Mapping . . . . .	242
A.2.7	Sra's LP to QP SVM Mapping . . . . .	244
A.3	Linear Programming Support Vector Machines . . . . .	245
B	Algorithms and Theorems Proofs . . . . .	248
B.1	Algorithms . . . . .	248
B.2	Theorems and Proofs . . . . .	249
B.2.1	Theorem on the ESV Bound . . . . .	249

B.2.2	Theorem on The Rank Bound . . . . .	252
B.2.3	KKT Conditions of Constraints Decomposition Method . . . .	253
C	Interior Point Methods, Datasets, Performance Metrics, and Additional Tables	255
C.1	Primal Dual Interior Point Methods for Linear Programming . . . . .	255
C.2	Data Sets Used in this Dissertation . . . . .	258
C.3	Performance Metrics . . . . .	271
C.3.1	Regression . . . . .	271
C.3.2	Classification . . . . .	272
C.4	Additional Tables from Chapter 3 . . . . .	278
D	Background in the Newton Method and in Kernel Functions . . . . .	287
D.1	Newton Method for Error Function Minimization . . . . .	287
D.1.1	Newton Method . . . . .	288
D.1.2	Globalization Strategy . . . . .	289
D.2	Background in Kernel Functions . . . . .	291
D.2.1	Mercer's Theorem . . . . .	291
D.2.2	Reproducing Kernel Hilbert Space . . . . .	292
D.2.3	Representer Theorem . . . . .	294
E	Remote Sensing . . . . .	295
E.1	Non-Linear Enhancement for MODIS True Color Images . . . . .	295
E.2	List of MODIS Level 1B Data Granules . . . . .	296
	Curriculum Vitae . . . . .	297

# List of Tables

3.1	Summary of the Dimensions and Properties of the Datasets. . . . .	59
3.2	Accuracy . . . . .	64
3.3	Balanced Error Rate . . . . .	65
3.4	Mean Absolute Error. . . . .	65
3.5	Support Vectors . . . . .	67
3.6	Sparse Support Vectors . . . . .	68
4.1	Total Training Time without speedup . . . . .	96
4.2	Total Training Time with Speedup. . . . .	96
4.3	Iterations reduction percent after speedup. . . . .	97
4.4	Summary of Behavior. . . . .	101
4.5	Summary of Testing Set Generalization Error Metrics and Statistical Properties of Residual Errors. . . . .	102
5.1	Variables Used for Prediction . . . . .	106
5.2	Electricity Load Prediction Errors . . . . .	111
6.1	Comparison of Segmentation Errors. ULap-UDFB system implemented with $J = 4$ and $L = 12$ . . . . .	148
7.1	Classifiers Performance. The processing time shown here is in milli- seconds. . . . .	170
A.1	Summary of XOR problem characterization. . . . .	227
C.1	Summary of the Dimensions and Properties of the Datasets. . . . .	267
C.2	Illustration of TP, FP, TN, and FN using a confusion matrix. . . . .	272

C.3	Illustration of TP, FP, TN, and FN for class 0, using a multi-class confusion matrix. . . . .	273
C.4	Illustration of TP, FP, TN, and FN for class 2, using a multi-class confusion matrix. . . . .	273
C.5	Classification targets and predicted values. Borrowed from [60]. . . .	276
C.6	Summary of Performance Metrics and Their Interpretation. . . . .	277
C.7	False Positives in Proportion to the Dataset Size. . . . .	278
C.8	False Negatives in Proportion to the Dataset Size. . . . .	279
C.9	True Positives Rate . . . . .	279
C.10	False Positives Rate . . . . .	280
C.11	Specificity . . . . .	280
C.12	Positive Predictive Value . . . . .	281
C.13	Negative Predictive Value . . . . .	281
C.14	False Discovery Rate . . . . .	282
C.15	Mathews Correlation Coefficient . . . . .	282
C.16	$F_1$ -Score . . . . .	283
C.17	Estimate of Scaled Error Rate . . . . .	283
C.18	Area Under the Receiver Operating Characteristic Curve . . . . .	284
C.19	Area Under the Receiver Operating Characteristic Curve Convex Hull	284
C.20	Root Mean Squared Error and Normalized Root Mean Squared Error.	285
C.21	Sum of Squared Error and Statistical Metrics . . . . .	285
C.22	Exact Support Vectors . . . . .	286
C.23	Saturated Support Vectors . . . . .	286

# List of Figures

1.1	Overview of a general pattern recognition process. . . . .	2
1.2	Different separating hyperplanes. The hyperplane denoted as $\mathbf{w}_B^T \mathbf{x}_i + b = 0$ correctly separates the two classes; however, $\mathbf{w}_C^T \mathbf{x}_i + b = 0$ maximizes the separation margin between classes. In contrast, $\mathbf{w}_A^T \mathbf{x}_i + b = 0$ is a poor separating hyperplane. . . . .	3
1.3	The hyperplane is defined by $\mathbf{w}^T \mathbf{x}_i + b = 0$ , and the support vectors are those that satisfy the condition $\mathbf{w}^T \mathbf{x}_i + b = d_i$ . . . . .	4
1.4	The data points (input space) are projected to a higher dimension (feature space) using a kernel function. . . . .	4
2.1	Illustration of the $\epsilon$ -insensitive loss function. . . . .	16
2.2	Linear regression: illustration of an $\epsilon$ -insensitive tube, fitted to the data points shown as circles. . . . .	17
2.3	Illustration of an SVM for non-linear regression, showing the regression curve together with the $\epsilon$ -insensitive tube. . . . .	17
2.4	Depending on the problem size, one may find three different strategies for SVR training. . . . .	25
3.1	Variable decomposition strategy from Torii, <i>et al.</i> [187], in which a subset of the variables $\alpha_i$ is considered for the solution of the LP-SVR problem, for all $i \in \mathcal{B}$ . The shaded area corresponds to an arbitrary selection of $ \mathcal{B} $ indices. . . . .	34

3.2	LP-SVR constraints decomposition strategy. This figure shows an example of the decomposition of the LP constraints. Here the coefficients matrix $\mathbf{A}$ and vector $\mathbf{b}$ are divided in $\tau$ blocks. This figure shows the linear programming chunking (LPC) approach by Bradley's, <i>et al.</i> [22,23]. . . . .	35
3.3	LP-SVR variable decomposition strategy. This illustrates a decomposition of the LP variables vector $\mathbf{z}$ and coefficients matrix $\mathbf{A}$ in which a subset of the variables $\alpha_i$ is considered for the solution of the LP-SVR problem, for all $i \in \mathcal{B}$ . The shaded area corresponds to an arbitrary selection of $ \mathcal{B} $ indices that produce a reduced LP: $\mathbf{z}_{\mathcal{B}}$ and $\mathbf{A}_{\mathcal{B}}$ . . . . .	44
3.4	Proposed LP-SVR constraints decomposition strategy. The proposed decomposition strategy exploits the LP-SVR structure to further reduce the problem size. Note that the reduced problem size is inversely proportional to $\tau$ . The notation $\mathbf{A}_{ij}$ refers to an element of the matrix $\mathbf{A}$ at the $i$ -row and $j$ -th column. . . . .	50
3.5	Behavior of the KKT conditions as the number of iterations progress. The primal, dual, and complementarity condition must converge to zero. The results shown represent the average value over several experiments on arbitrarily three-class non-separable classification problems. . . . .	58
3.6	Training Time as a Function of the Problem Size. . . . .	62
3.7	Total Number of Iterations as a Function of the Problem Size. . . . .	63
3.8	Objective Function as a Function of Iterations Number. . . . .	64
3.9	Support Vectors as a Function of Iterations. SPV SSV ESV SV. . . . .	66

4.1	Relationship between convex hull and maximum Mahalanobis distance for the two class problem. Here it is shown the original separable two class problem, class convex hull, support vectors, and $k = 8$ maximum Mahalanobis distance samples. Note that both SVs and Convex Hull match the $k = 8$ maximum Mahalanobis distance samples. . . . .	73
4.2	Mahalanobis distance-ranking of class indices using feature vectors in either the input space or the kernel-induced feature space. . . . .	76
4.3	Relationship between convex hull and maximum Mahalanobis distance for the two class problem. Here it is shown the original separable two class problem, class convex hull, support vectors, and $k$ maximum Mahalanobis distance samples. Note that both SVs and Convex Hull match the $k$ maximum Mahalanobis distance samples. . . . .	78
4.4	Response of the root mean squared error as a function of $\theta = [C, \sigma]$ . Note, how the error surface is non-smooth and has many local minima.	80
4.5	Unit step function approximation. A function $\Psi\{\cdot\}$ was used for convenience in easing computations. . . . .	82
4.6	Comparison between a logarithmic grid refinement technique (a)-(c) [6, 7] and the proposed adaptation of the newton method. . . . .	85
4.6	Continued... comparison between a logarithmic grid refinement technique (a)-(c) [6, 7] and the proposed adaptation of the newton method.	86
4.6	Continued... comparison between a logarithmic grid refinement technique (a)-(c) [6, 7] and the proposed adaptation of the newton method.	87
4.7	Sample Selection. Support Vectors as a Function of Iterations. SSV, ESV, SPV, and SV. . . . .	94
4.8	Support Vectors as a Function of Iterations. SPV SSV ESV SV. . . .	95
4.9	Total Model Selection Time and Number of Iterations as a Function of the Problem Size. . . . .	98
4.10	BER and AUC as a Function of Iterations Number. . . . .	99

4.11	RMSE, and MAE as a Function of Iterations Number. . . . .	100
5.1	Framework to build regression models for power load prediction. Blocks on the left indicate the input variables <i>i.e.</i> , attributes used to build the regression models. . . . .	108
5.2	Two-day window of true data compared with predicted for the four different methods (top). Error residuals for the four methods (bottom).	112
5.3	Christmas two-day window of true data compared with predicted for the four different methods (top). Error residuals for the four methods (bottom). Note the high prediction error between 14:00-21:00 Hrs. . .	113
5.4	Error distribution for the LS SVM, BRT, FFNN, and LP-SVR regression methods. The methods with smallest variances are FFNN and LP-SVR. . . . .	114
5.5	Absolute error distribution of the LS SVM, BRT, FFNN, and LP-SVR regression methods. The vertical lines indicate the mean absolute error for each of the four methods as reported in Table 5.2. . . . .	115
5.6	Average error by hour of day. Note the error proportional difference in early morning hours and afternoon hours. . . . .	116
5.7	Average error by day of week. Note the error proportional difference in working and non-working days. . . . .	117
5.8	Hourly breakdown of the LP-SVR mean absolute prediction error. Note that the early morning hours have smaller variability. . . . .	118
5.9	Daily breakdown of the LP-SVR mean absolute prediction error. Note that Mondays and Fridays have the largest average error. . . . .	119
5.10	Monthly breakdown of the LP-SVR mean absolute prediction error. Note that the month of December exhibits the largest average error, and the largest variability. . . . .	120
6.1	Classical segmentation system based on multichannel filtering. . . .	123