

Support Vector Machines for QSAR Analysis



Matthew Trotter
Dept. of Computer Science
University College London
m.trotter@cs.ucl.ac.uk

Working in collaboration with Darko Butina at Glaxo Wellcome.

Overview

- What is a Support Vector Machine?
- Pros & Cons of the SVM for QSAR analysis.
- Initial Results:
 - Benchmark Test of SVM on QSAR Data.
- Work in Progress
- Conclusion
- References

SVM - History

- Initial Structural Risk Minimisation idea by Vapnik in the late 1960s.
- SVM first mentioned specifically in 1993 paper by Vapnik and others. General SVM formulation appeared in 1995.
- Since publication of Vapnik's book, 'The Nature of Statistical Learning Theory' [1], interest has mounted.
- Current explosion of interest within Computer Science, but few reports of use in industry.

What is a Support Vector Machine?

- Maximal margin classifier, based on Vapnik's *Structural Risk Minimisation* [1].
- Minimises an upper bound on the *Expected Generalisation Error*.

$$R(\mathbf{a}) \leq R_{emp}(\mathbf{a}) + \sqrt{\frac{h(\log(2l/h) + 1) - \log(h/4)}{l}}$$

- *Optimum Separating Hyperplane* provides minimum expected generalisation error.
- Only points which lie on the margin are used to construct the decision boundary.

Optimal Separating Hyperplane - I

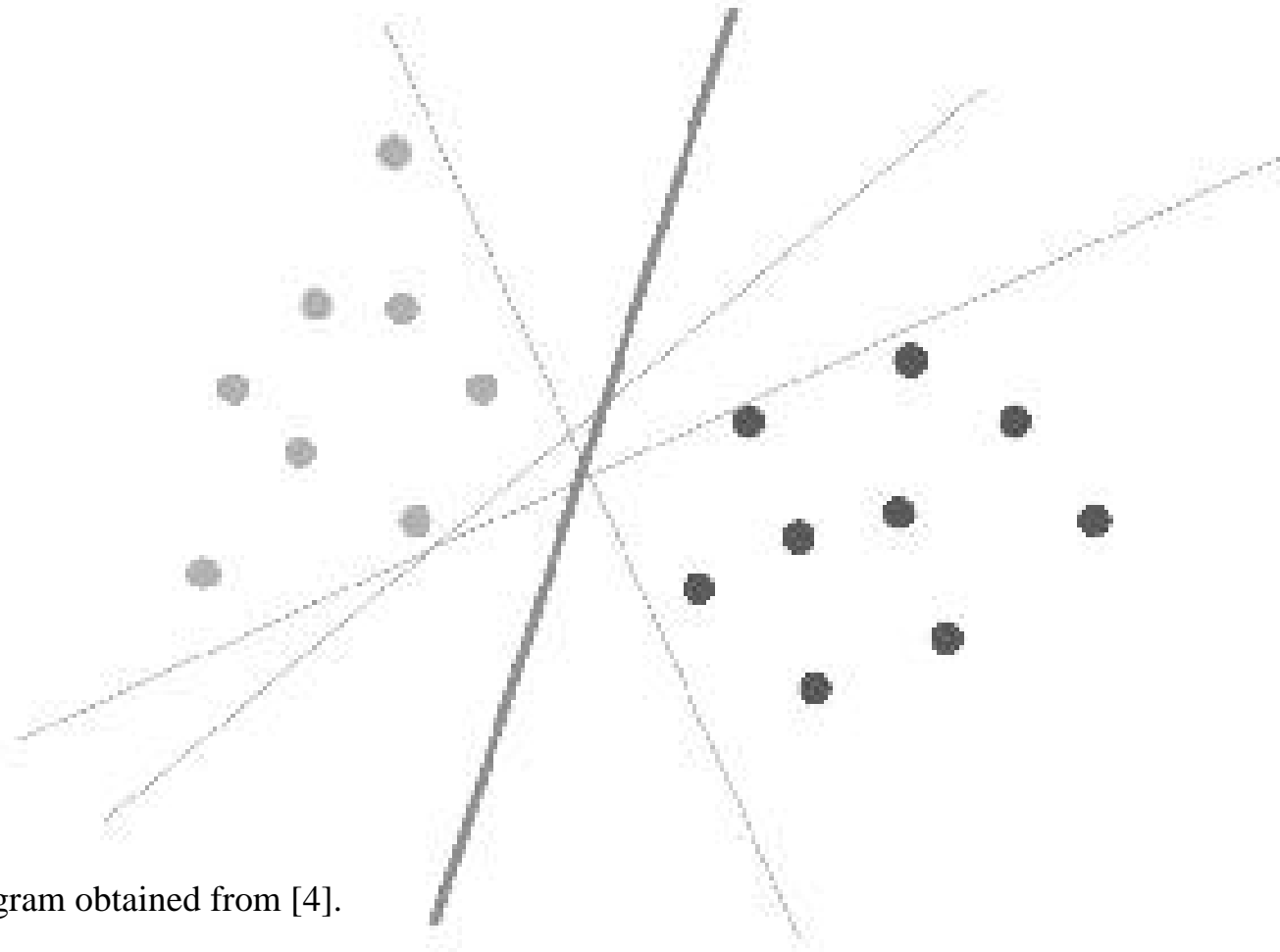


Diagram obtained from [4].

Optimal Separating Hyperplane - II

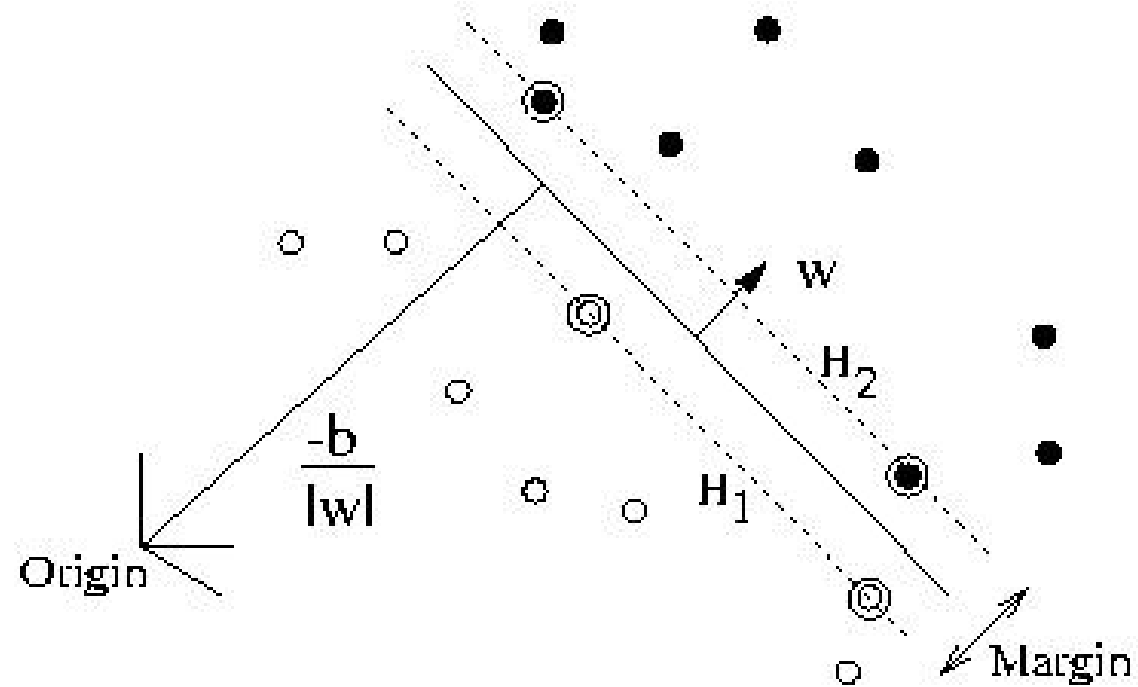


Diagram obtained from [2].

Linear Formulation

- Optimum separating hyperplane can be seen as the midpoint between two, parallel hyperplanes:

$$\mathbf{x}_i \cdot \mathbf{w} + b \geq +1 \quad y_i = +1$$

$$\mathbf{x}_i \cdot \mathbf{w} + b \leq -1 \quad y_i = -1$$

- This can be formed into a single inequality: $y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 \geq 0 \quad \forall i$

- Distance between the two margin hyperplanes is: $2/\|\mathbf{w}\|$

- Optimum separating hyperplane is found by minimising: $\|\mathbf{w}\|^2$

Full explanation and derivation of the theory can be found in [2].

Linear Formulation

- Lagrangian formulation gives:

$$L_P \equiv \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^l \mathbf{a}_i y_i (\mathbf{x}_i \cdot \mathbf{w} + b) + \sum_{i=1}^l \mathbf{a}_i$$
$$\mathbf{w} = \sum_i \mathbf{a}_i y_i \mathbf{x}_i$$
$$\sum_i \mathbf{a}_i y_i = 0$$

- Problem is best solved by maximising the Lagrangian dual formulation:

$$L_D = \sum_i \mathbf{a}_i - \frac{1}{2} \sum_{i,j} \mathbf{a}_i \mathbf{a}_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$$
$$\mathbf{w} = \sum_{i=1}^{N_S} \mathbf{a}_i y_i \mathbf{x}_i$$

- New points classified by observing the sign of: $\mathbf{y}' = \text{sgn}(\mathbf{w} \cdot \mathbf{x}' + b)$

Slack Variables for Non-Separable Data

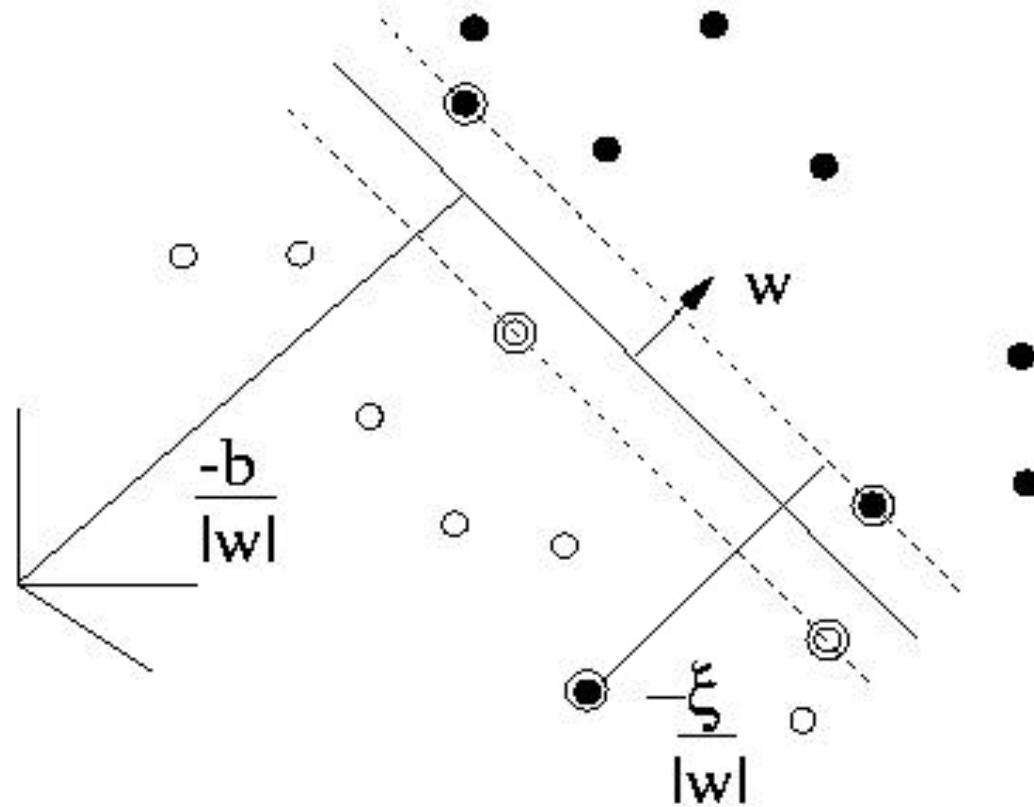


Diagram obtained from [2].

Slack Variables - Formulation

- Linear formulation changed to:

$$\mathbf{x}_i \cdot \mathbf{w} + b \geq +1 - \mathbf{x}_i \quad y_i = +1$$

$$\mathbf{x}_i \cdot \mathbf{w} + b \leq -1 + \mathbf{x}_i \quad y_i = -1$$

$$\mathbf{x} \geq 0, \forall i$$

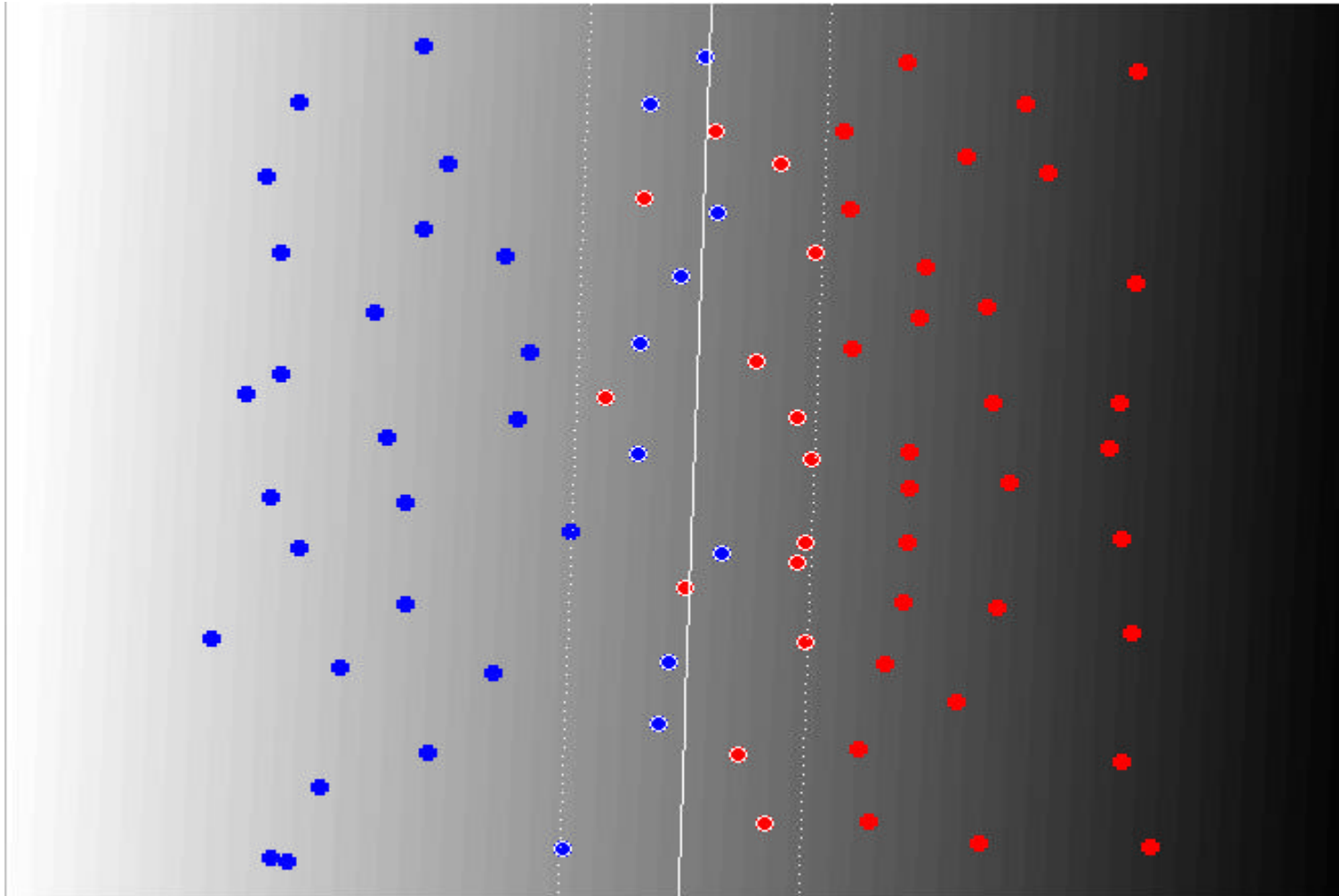
- Now, the following term is minimised:

$$\|\mathbf{w}\|^2 \rightarrow (\|\mathbf{w}\|^2 + C(\sum_i \mathbf{x}_i)^k)$$

- If 'k' is chosen as 1, Lagrangian dual remains unchanged, but alphas are bounded by C:

$$L_D \equiv \sum_i \mathbf{a}_i - \frac{1}{2} \sum_{i,j} \mathbf{a}_i \mathbf{a}_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \quad 0 \leq \mathbf{a}_i \leq C$$

Slack Variables in Action



Mapping to Feature Space for Separation

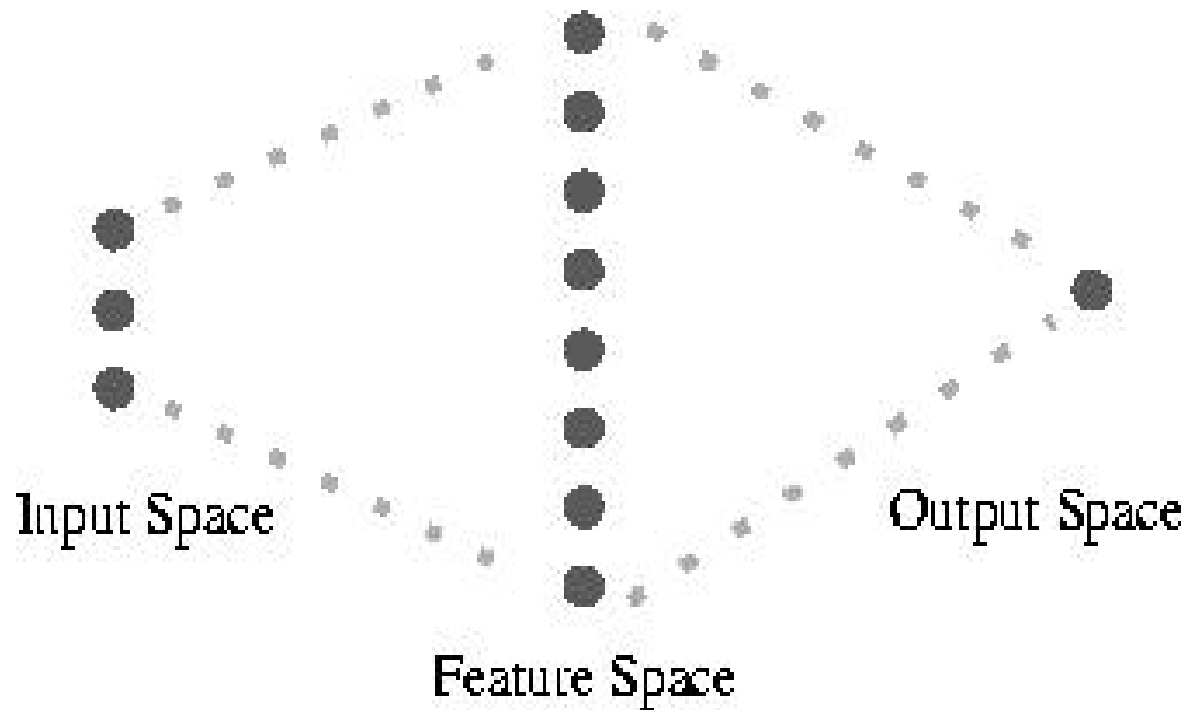


Diagram obtained from [4].

Kernels

■ Recall:

$$L_D = \sum_i \mathbf{a}_i - \frac{1}{2} \sum_{i,j} \mathbf{a}_i \mathbf{a}_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$$

■ We would like to map the training data into a high-dimensional space:

$$L_D = \sum_i \mathbf{a}_i - \frac{1}{2} \sum_{i,j} \mathbf{a}_i \mathbf{a}_j y_i y_j \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$$

■ Calculation becomes increasingly difficult, the higher the dimension. Use a Mercer's kernel to do the calculations in the high-dimensional space:

$$K(\mathbf{x}, \mathbf{y}) = \sum_i \Phi(\mathbf{x})_i \cdot \Phi(\mathbf{y})_i$$

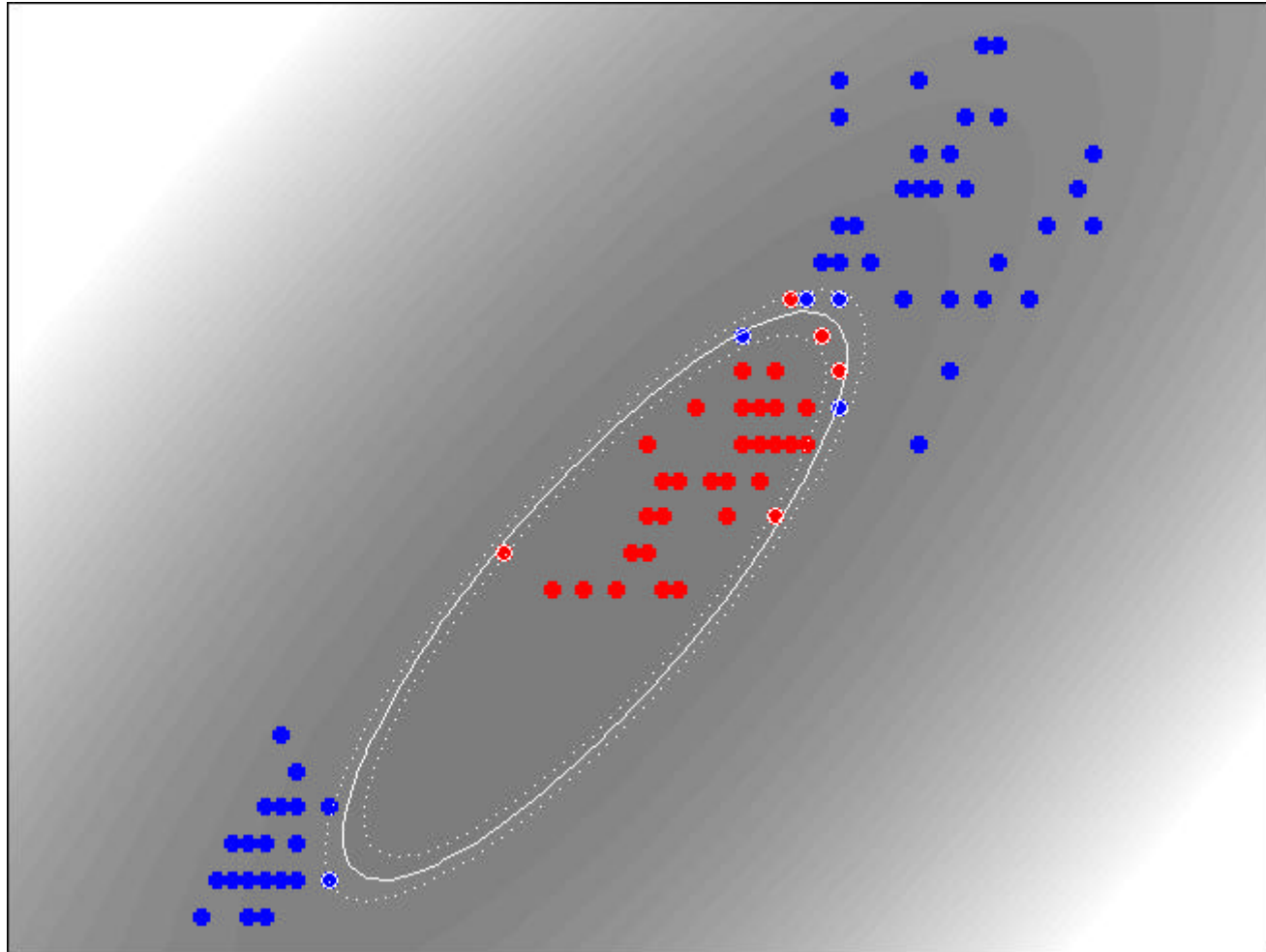
■ Formulation becomes:

$$L_D \equiv \sum_i \mathbf{a}_i - \frac{1}{2} \sum_{i,j} \mathbf{a}_i \mathbf{a}_j y_i y_j K(\mathbf{x}_i \cdot \mathbf{x}_j)$$

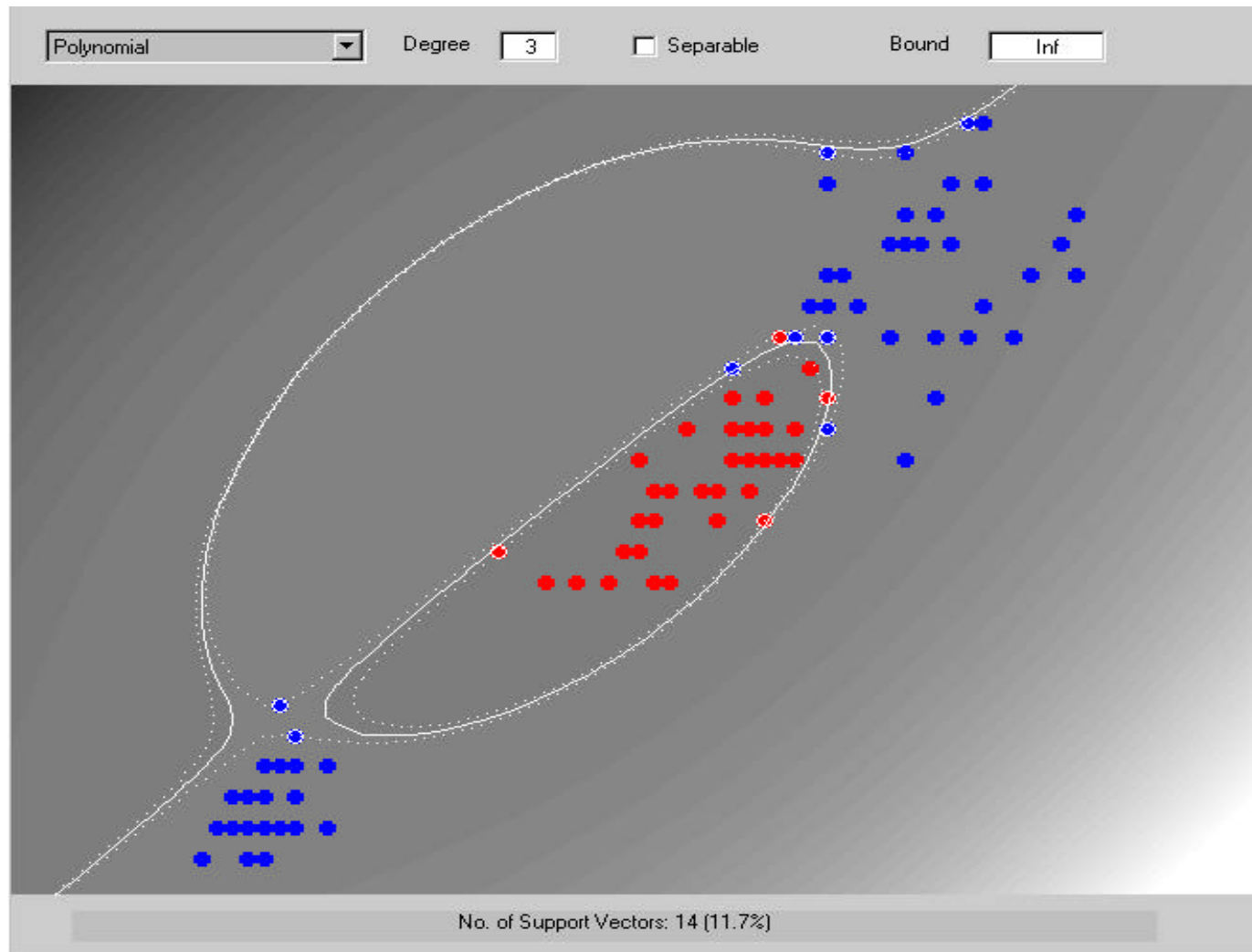
Typical Kernel Functions

- Linear: $(\mathbf{x}_i \cdot \mathbf{x}_j)$
- Polynomial: $(\mathbf{x}_i \cdot \mathbf{x}_j + 1)^d$
- Radial Basis Function: $\exp(-(\mathbf{x}_i - \mathbf{x}_j)^2 / (2\sigma^2))$
- Sigmoid (MLP): $\tanh(\text{scale}(\mathbf{x}_i \cdot \mathbf{x}_j) - \text{offset})$
- *Only positive definite functions, which satisfy Mercer's conditions can be used to represent a legitimate inner product in feature space.*

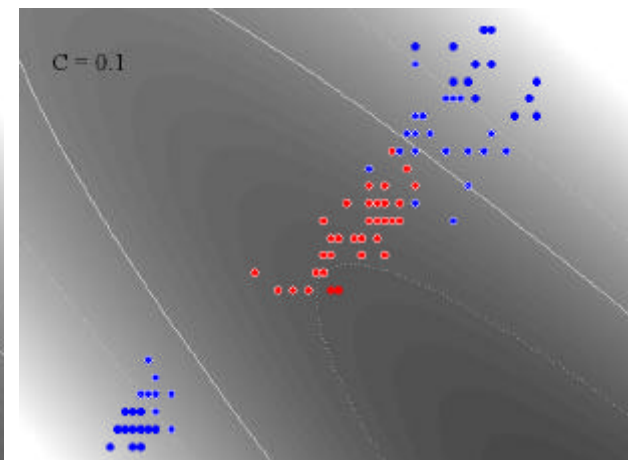
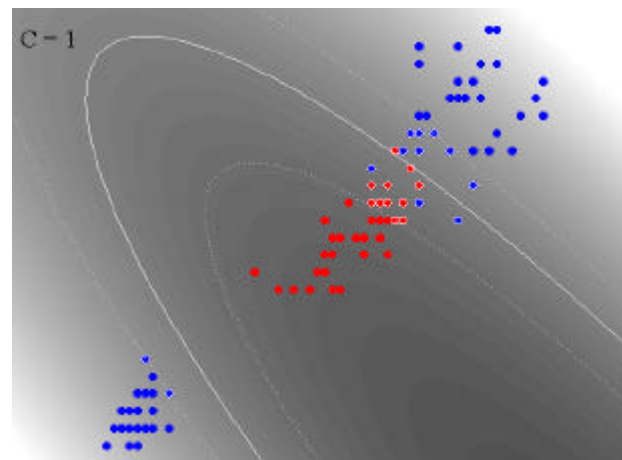
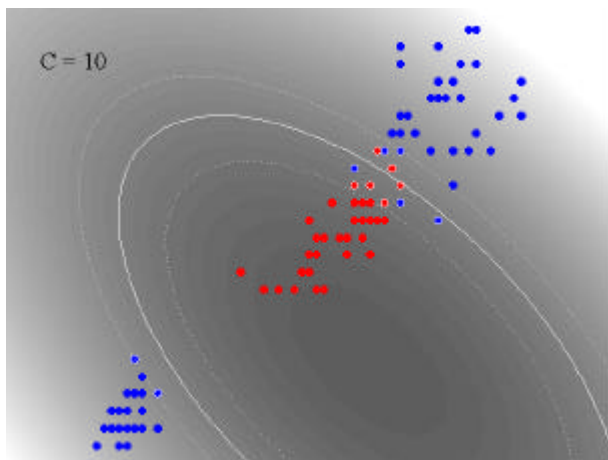
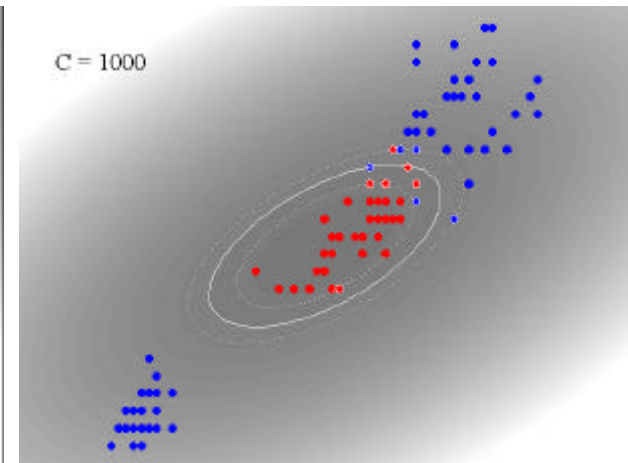
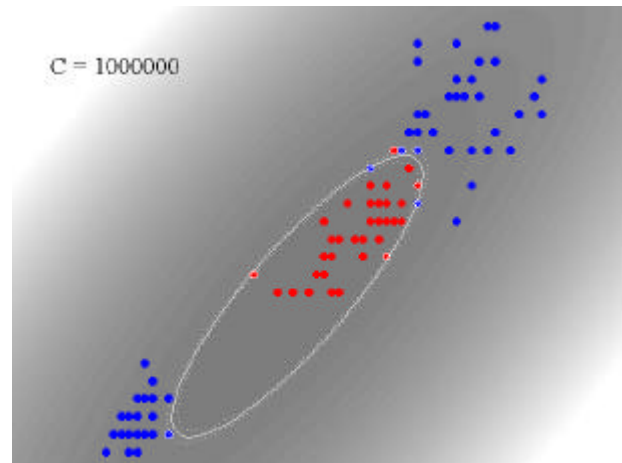
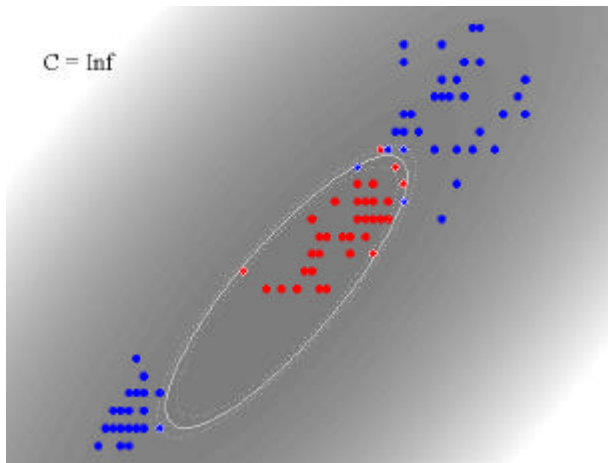
Relevance of SV Points to Separation - I



Careful Kernel Selection is Crucial



Capacity Control - The Importance of C



SVM - QSAR Pros

- Good generalisation performance.
- Fast classification of unknown compounds.
- Solution can also provide information regarding the training data.
- Potential for feature selection and outlier detection.
- Few free parameters.

SVM - QSAR Cons

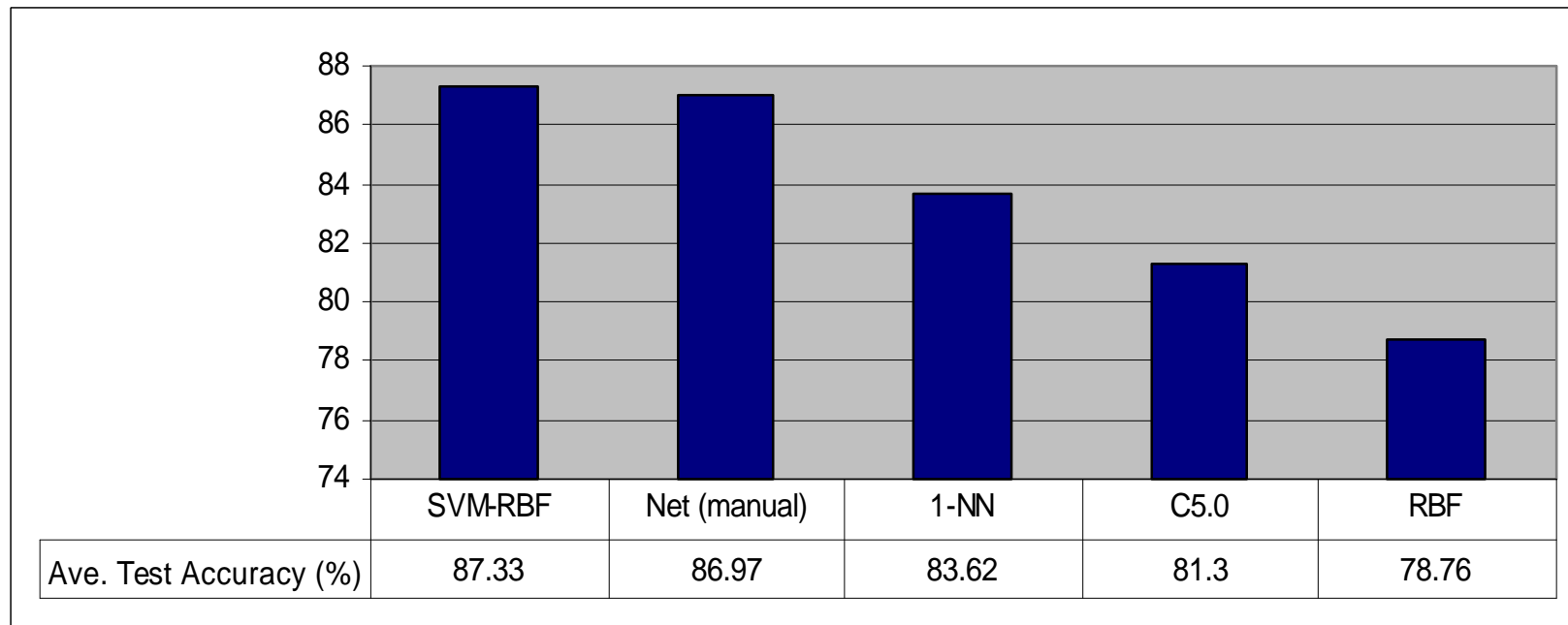
- Training time.
- Parameter selection is time consuming and data dependent.
- Opaque technique.

Performance Comparison

- Performance comparison between the SVM and Neural Net, RBF Net, C5.0 Decision Tree and One-Nearest-Neighbour classifiers on well-known, publicly available QSAR data [3].
- Data obtained from [6]. 54 attributes and binary class targets.
- Neural Net, RBF Net and Decision Tree implemented using Clementine data mining software. Nearest-Neighbour algorithm implemented using Matlab. SVM implemented using Thorsten Joachims' SVM^{light} algorithm [5].

Initial Results

- SVM significantly outperforms all techniques bar a manually capacity controlled Neural Network.



Work in Progress

- Revision of paper [3] for publication in special edition of 'Computers & Chemistry'.
- Significance of SV Points to the QSAR analyst.
- Consistent parameter set-up.
- Improved performance through use of domain knowledge.

Conclusion

- SVMs show good potential for QSAR analysis.
- Trials demonstrate an increase in predictive accuracy over a selection of current machine learning techniques.
- Possibilities exist for feature selection and outlier detection.
- Work on the practical usage of SVMs for QSAR analysis and significance of the SV points is in progress.

References

- [1] ‘The Nature of Statistical Learning Theory’
V. Vapnik; Springer-Verlag, 1995.

- [2] ‘A Tutorial on Support Vector Machines for Pattern Recognition’
C. Burges; Data Mining and Knowledge Discovery, Vol. 2, No. 2, 1998.

- [3] ‘Drug Design by Machine Learning: Support Vector Machines for Pharmaceutical Data Analysis’
R. Burbidge, M. Trotter, B. Buxton & S. Holden; Awaiting Publication.

- [4] SVM Toolbox for Matlab obtained from the University of Southampton and used to create the examples featured in this presentation: www.isis.ecs.soton.ac.uk/research/svm/

- [5] Thorsten Joachims’ SVM^{light} algorithm can be obtained from:
www-ai.informatik.uni-dortmund.de/thorsten/svm_light.html

- [6] QSAR data obtained from the UCI ML Data Repository: www.ics.uci.edu/
Data donated by Dr. Ross King, Imperial Cancer Research Fund. Further details can be found in [3].