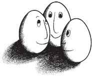
Finding all Local Models in Parallel: Multi-Objective SVM

Ingo Mierswa
AI Unit
University of Dortmund

Dagstuhl Seminar 2007

Outline



Motivation

- Model = Global Model + Local Model(s) + Noise
- SVM can find both the global and the local models
- Conflicting criteria: training error and model complexity
- Users have to specify a weighting factor C for a trade-off
- Local models: those for higher weights on training error

Solution

Embed multi-objective evolutionary algorithms instead of the quadratic programming approach into SVM.

Motivation

- Model = Global Model + Local Model(s) + Noise
- SVM can find both the global and the local models
- Conflicting criteria: training error and model complexity
- Users have to specify a weighting factor C for a trade-off
- Local models: those for higher weights on training error

Solution

Embed multi-objective evolutionary algorithms instead of the quadratic programming approach into SVM.

Motivation

- Model = Global Model + Local Model(s) + Noise
- SVM can find both the global and the local models
- Conflicting criteria: training error and model complexity
- Users have to specify a weighting factor C for a trade-off
- Local models: those for higher weights on training error

Solution

Embed multi-objective evolutionary algorithms instead of the quadratic programming approach into SVM.

Motivation

- Model = Global Model + Local Model(s) + Noise
- SVM can find both the global and the local models
- Conflicting criteria: training error and model complexity
- Users have to specify a weighting factor C for a trade-off
- Local models: those for higher weights on training error

Solution

Embed multi-objective evolutionary algorithms instead of the quadratic programming approach into SVM.

Motivation

- Model = Global Model + Local Model(s) + Noise
- SVM can find both the global and the local models
- Conflicting criteria: training error and model complexity
- Users have to specify a weighting factor C for a trade-off
- Local models: those for higher weights on training error

Solution

Embed multi-objective evolutionary algorithms instead of the quadratic programming approach into SVM.

Motivation

- Model = Global Model + Local Model(s) + Noise
- SVM can find both the global and the local models
- Conflicting criteria: training error and model complexity
- Users have to specify a weighting factor C for a trade-off
- Local models: those for higher weights on training error

Solution

Embed multi-objective evolutionary algorithms instead of the quadratic programming approach into SVM.

Desired Result

- The result of multi-objective optimization is not a single solution but a set of solutions (Pareto set)
- These solutions correspond to the optimal solutions for all possible weightings for both criteria

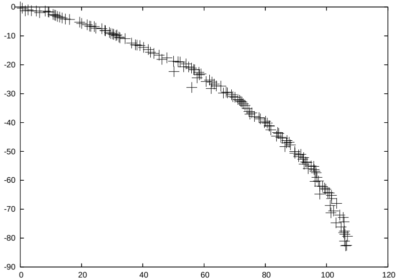
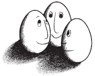


Figure: The Pareto-optimal solutions for two competing criteria



The Primal SVM Problem

Primal SVM Problem

The basic form of the primal SVM optimization problem is the following:

$$\text{minimize } \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi_i$$

$$\text{subject to } \forall i : y_i\left(\langle w, x_i\rangle + b\right) \geq 1 - \xi_i$$

$$\text{and } \forall i : \xi_i \geq 0.$$

Weighting Factor

The parameter C is a user defined weight for the both conflicting parts of the optimization criterion.

The Primal SVM Problem

Primal SVM Problem

The basic form of the primal SVM optimization problem is the following:

$$\text{minimize } \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi_i$$

$$\text{subject to } \forall i : y_i\left(\langle w, x_i\rangle + b\right) \geq 1 - \xi_i$$

$$\text{and } \forall i : \xi_i \geq 0.$$

Weighting Factor

The parameter C is a user defined weight for the both conflicting parts of the optimization criterion.

Multiple Conflicting Objectives

- EA inside SVM allows for a straightforward application of multi-objective selection schemes
- We divide the criteria of the primal SVM optimization problem into two optimization targets while the weighting factor C can be omitted

Goal

Transform both objectives into their dual form in order to allow the efficient optimization of the problems including the usage of kernel functions.

Multiple Conflicting Objectives

- EA inside SVM allows for a straightforward application of multi-objective selection schemes
- We divide the criteria of the primal SVM optimization problem into two optimization targets while the weighting factor C can be omitted

Goal

Transform both objectives into their dual form in order to allow the efficient optimization of the problems including the usage of kernel functions.

Multiple Conflicting Objectives

- EA inside SVM allows for a straightforward application of multi-objective selection schemes
- We divide the criteria of the primal SVM optimization problem into two optimization targets while the weighting factor C can be omitted

Goal

Transform both objectives into their dual form in order to allow the efficient optimization of the problems including the usage of kernel functions.

Multiple Conflicting Objectives

Primal Objective 1

$$\text{minimize } \frac{1}{2}||w||^2$$
$$\text{subject to } \forall i : y_i \left(\langle w, x_i \rangle + b\right) \geq 1 - \xi_i$$
$$\text{and } \forall i : \xi_i \geq 0$$

Primal Objective 2

$$\text{minimize } \sum_{i=1}^{n} \xi_i$$
$$\text{subject to } \forall i : y_i \left(\langle w, x_i \rangle + b\right) \geq 1 - \xi_i$$
$$\text{and } \forall i : \xi_i \geq 0.$$

Objective 1: Maximizing the Margin

- Introduce positive Lagrange multipliers α for the first set of inequality constraints and multipliers β for the second set of inequality constraints:

$$L_p^{(1)} = \frac{1}{2}||w||^2 - \sum_{i=1}^{n} \alpha_i \left(y_i \left(\langle w, x_i \rangle + b \right) + \xi_i - 1 \right) - \sum_{i=1}^{n} \beta_i \xi_i$$

- Set the derivatives to 0:

$$\frac{\partial L_p^{(1)}}{\partial w}(w, b, \xi, \alpha, \beta) = w - \sum_{i=1}^{n} y_i \alpha_i x_i = 0,$$

$$\frac{\partial L_p^{(1)}}{\partial b}(w, b, \xi, \alpha, \beta) = \sum_{i=1}^{n} \alpha_i y_i = 0,$$

$$\frac{\partial L_p^{(1)}}{\partial \xi_i}(w, b, \xi, \alpha, \beta) = -\alpha_i - \beta_i = 0$$

Plugging the Derivatives into the Primal

- Plugging the derivatives into the primal objective function $L_p^{(1)}$ delivers

$$L_p^{(1)} = \frac{1}{2}||w||^2 - \sum_{i=1}^{n} -\alpha_i y_i \left\langle \sum_{j=1}^{n} \alpha_j y_j x_j, x_i \right\rangle + \sum_{i=1}^{n} \alpha_i$$

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

- The Wolfe dual must be maximized leading to the first objective of the multi-objective SVM
- Result is very similar to the dual SVM problem stated above but without the upper bound C for the α_i

The First Objective of the MO-SVM

First Objective

The first SVM objective (maximize margin) is defined as:

$$\text{maximize} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j k(x_i, x_j)$$

$$\text{subject to } \alpha_i \geq 0 \text{ for all } i = 1, \ldots, n$$

$$\text{and } \sum_{i=1}^{n} \alpha_i y_i = 0$$

Objective 2: Minimize Training Errors

- We again add positive Lagrange multipliers α and β:

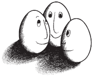
$$L_p^{(2)} = \sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i \left(\left(y_i \langle w, x_i \rangle + b \right) + \xi_i - 1 \right) - \sum_{i=1}^{n} \beta_i \xi_i$$

- Setting the derivatives to 0 leads to slightly different conditions on the derivatives of $L_p^{(2)}$:

$$\frac{\partial L_p^{(2)}}{\partial w}(w, b, \xi, \alpha, \beta) = -\sum_{i=1}^{n} y_i \alpha_i x_i = 0,$$

$$\frac{\partial L_p^{(2)}}{\partial b}(w, b, \xi, \alpha, \beta) = \sum_{i=1}^{n} \alpha_i y_i = 0,$$

$$\frac{\partial L_p^{(2)}}{\partial \xi_i}(w, b, \xi, \alpha, \beta) = 1 - \alpha_i - \beta_i = 0$$



Plugging the Derivatives into the Primal

- Plugging the derivatives into the $L_p^{(2)}$ cancels out most terms:

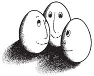
$$L_p^{(2)} = \sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i \xi_i + \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \beta_i \xi_i$$

- Together with the third derivative we can replace the β_i by $1 - \alpha_i$ leading to

$$L_p^{(2)} = \sum_{i=1}^{n} \alpha_i \xi_i - \sum_{i=1}^{n} \alpha_i \xi_i + \sum_{i=1}^{n} \alpha_i$$

$$L_p^{(2)} = \sum_{i=1}^{n} \alpha_i$$

- Maximizing the Wolfe dual leads to the second objective of the multi-objective SVM



The Second Objective of the MO-SVM

Second Objective

The second SVM objective (minimize error) is defined as:

$$\text{maximize} \sum_{i=1}^{n} \alpha_i$$

$$\text{subject to } \alpha_i \geq 0 \text{ for all } i = 1, \ldots, n$$

$$\text{and} \sum_{i=1}^{n} \alpha_i y_i = 0$$

Used Objectives

Set of all Objectives

Maximize the terms

$$-\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j k\left(x_i, x_j\right),$$

$$\text{and} \sum_{i=1}^{n} \alpha_i$$

subject to $\alpha_i \geq 0$ for all $i = 1, \ldots, n$

The result will be a Pareto front showing all models which are optimal for all possible weightings between both criteria.

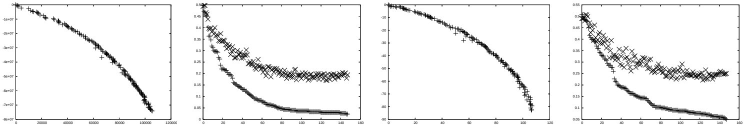
Data Sets

Data set	n	m	Source	σ	Default
Spiral	1000	2	Synthetical	1.000	50.00
Checkerboard	1000	2	Synthetical	1.000	50.00
Sonar	208	60	UCI	1.000	46.62
Diabetes	768	8	UCI	0.001	34.89
Lupus	87	3	StatLib	0.001	40.00
Crabs	200	7	StatLib	0.100	50.00

All experiments were performed with the machine learning environment YALE [1].

[1] http://yale.sf.net/

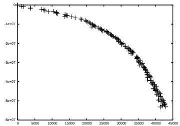
Results



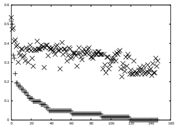
(a) Spiral Pareto (b) Spiral Generalization (c) Checkerboard Pareto (d) Checkerboard Generalization

Figure: The results for all data sets. The left plot for each dataset shows the Pareto front delivered by the multi-objective SVM proposed in this paper (x: margin size, y: training error). The right plot shows the training (+) and testing (×) errors (on a hold-out set of 20%) for all individuals of the resulting Pareto fronts (x: margin size, y: generalization error).

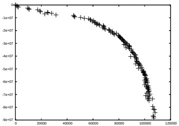
Results II



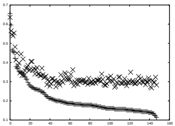
(a) Sonar Pareto



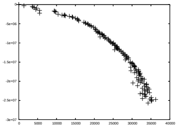
(b) Sonar Generalization



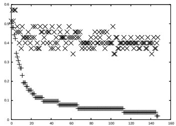
(c) Diabetes Pareto



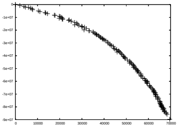
(d) Diabetes Generalization



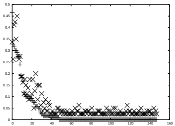
(e) Lupus Pareto



(f) Lupus Generalization

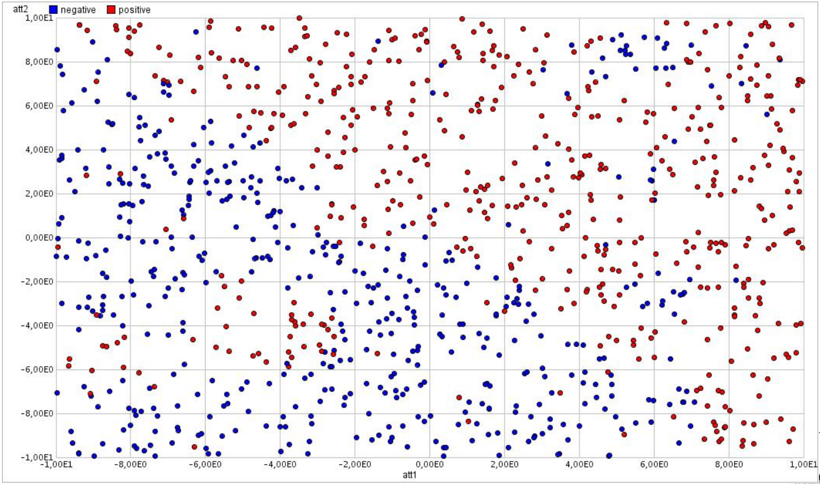


(g) Crabs Pareto

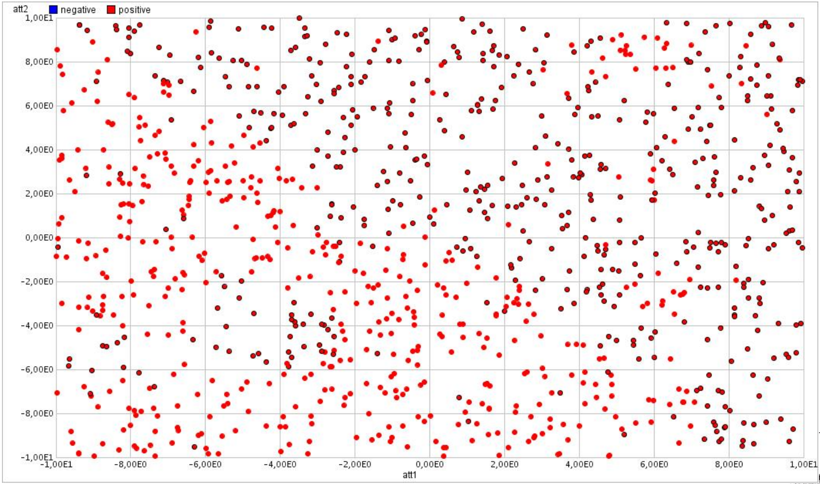


(h) Crabs Generalization

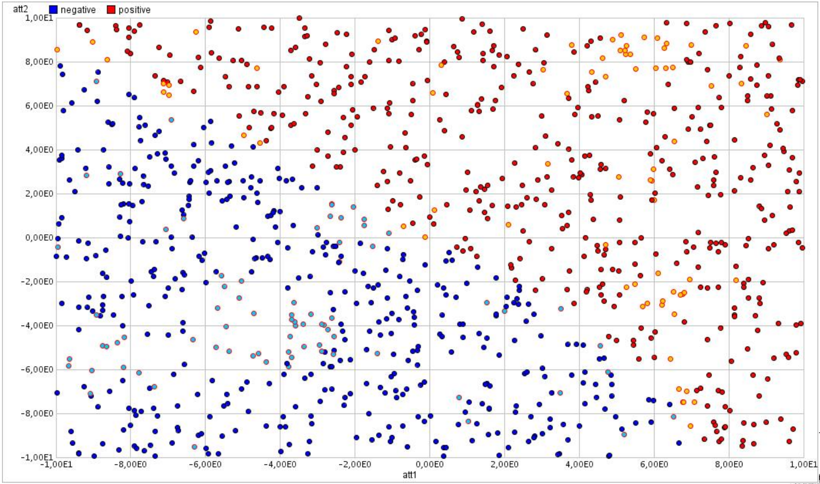
From Global to Local Models – Data

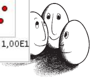


From Global to Local Models – Largest Margin

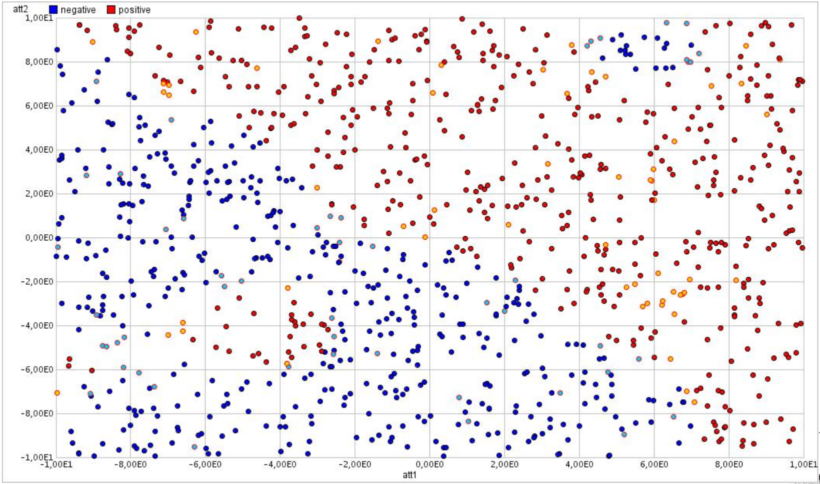


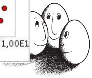
From Global to Local Models – The Global Model



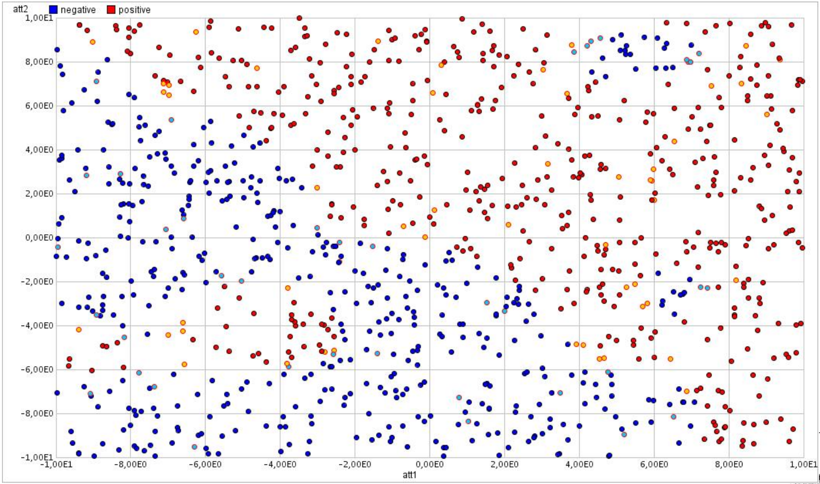


From Global to Local Models

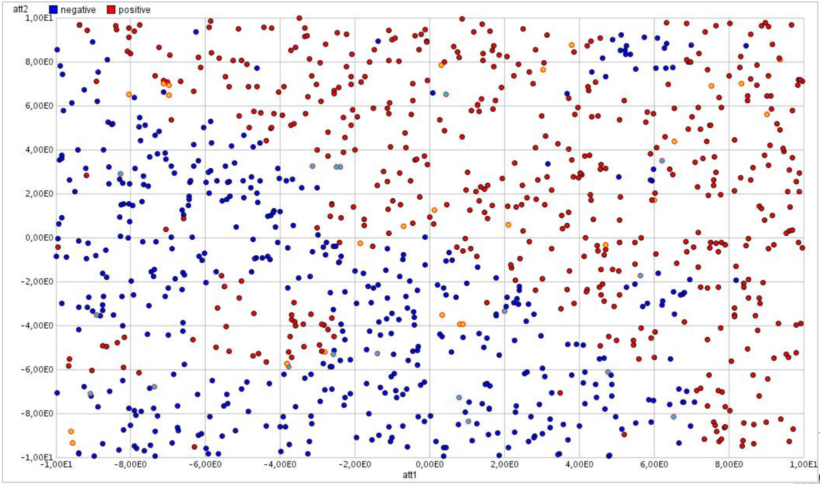




From Global to Local Models – Best Generalization



From Global to Local Models – Lowest Training Error



Main Advantage of MO-SVM

- The generalization ability plotted on the right sides clearly shows the location where overfitting occurs
- Please note that these plots could also be generated for usual SVM by iteratively applying the learner for different parameter settings but . . .
- . . . this will need one learning run for each possible value of C!

Full Knowledge in One Single Run!

The MO-SVM approach has the advantage that all models are calculated in one single run which is far less time-consuming

Main Advantage of MO-SVM

- The generalization ability plotted on the right sides clearly shows the location where overfitting occurs
- Please note that these plots could also be generated for usual SVM by iteratively applying the learner for different parameter settings but . . .
- . . . this will need one learning run for each possible value of C!

Full Knowledge in One Single Run!

The MO-SVM approach has the advantage that all models are calculated in one single run which is far less time-consuming

Main Advantage of MO-SVM

- The generalization ability plotted on the right sides clearly shows the location where overfitting occurs
- Please note that these plots could also be generated for usual SVM by iteratively applying the learner for different parameter settings but . . .
- . . . this will need one learning run for each possible value of C!

Full Knowledge in One Single Run!

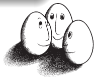
The MO-SVM approach has the advantage that all models are calculated in one single run which is far less time-consuming

Main Advantage of MO-SVM

- The generalization ability plotted on the right sides clearly shows the location where overfitting occurs
- Please note that these plots could also be generated for usual SVM by iteratively applying the learner for different parameter settings but ...
- ... this will need one learning run for each possible value of C!

Full Knowledge in One Single Run!

The MO-SVM approach has the advantage that all models are calculated in one single run which is far less time-consuming

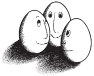


Conclusion

- Trade-off between training error and model complexity is now explicitly stated
- The optimization problem of SVM is divided in two parts and both parts are transformed into their dual form
- The optional usage of a hold-out set is suggested in order to guide the user for the final selection of a solution
- All information from the most global to the most local models is gathered in a single run!

Conclusion

- Trade-off between training error and model complexity is now explicitly stated
- The optimization problem of SVM is divided in two parts and both parts are transformed into their dual form
- The optional usage of a hold-out set is suggested in order to guide the user for the final selection of a solution
- All information from the most global to the most local models is gathered in a single run!



Conclusion

- Trade-off between training error and model complexity is now explicitly stated
- The optimization problem of SVM is divided in two parts and both parts are transformed into their dual form
- The optional usage of a hold-out set is suggested in order to guide the user for the final selection of a solution
- All information from the most global to the most local models is gathered in a single run!

Conclusion

- Trade-off between training error and model complexity is now explicitly stated
- The optimization problem of SVM is divided in two parts and both parts are transformed into their dual form
- The optional usage of a hold-out set is suggested in order to guide the user for the final selection of a solution
- All information from the most global to the most local models is gathered in a single run!