An Introduction to Machine Learning

L4: Support Vector Classification

Alexander J. Smola

Statistical Machine Learning Program
Canberra, ACT 0200 Australia
Alex.Smola@nicta.com.au

Tata Institute, Pune, January 2007

Overview

L1: Machine learning and probability theory
Introduction to pattern recognition, classification, regression, novelty detection, probability theory, Bayes rule, inference

L2: Density estimation and Parzen windows
Nearest Neighbor, Kernels density estimation, Silverman's rule, Watson Nadaraya estimator, crossvalidation

L3: Perceptron and Kernels
Hebb's rule, perceptron algorithm, convergence, kernels

L4: Support Vector estimation
Geometrical view, dual problem, convex optimization, kernels

L5: Support Vector estimation
Regression, Quantile regression, Novelty detection, ν-trick

L6: Structured Estimation
Sequence annotation, web page ranking, path planning, implementation and optimization

L4 Support Vector Classification

Support Vector Machine

- Problem definition
- Geometrical picture
- Optimization problem

Optimization Problem

- Hard margin
- Convexity
- Dual problem
- Soft margin problem

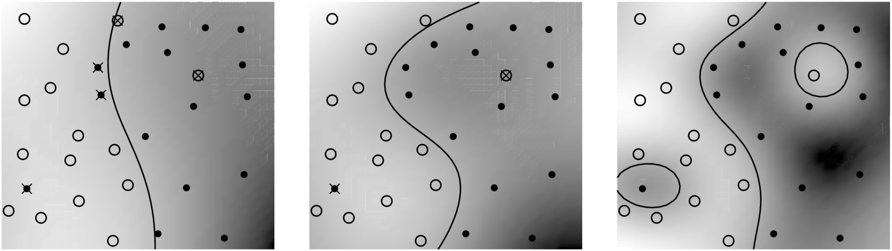
Classification

Data

Pairs of observations (x_i, y_i) generated from some distribution $\mathrm{P}(x, y)$, e.g., (blood status, cancer), (credit transaction, fraud), (profile of jet engine, defect)

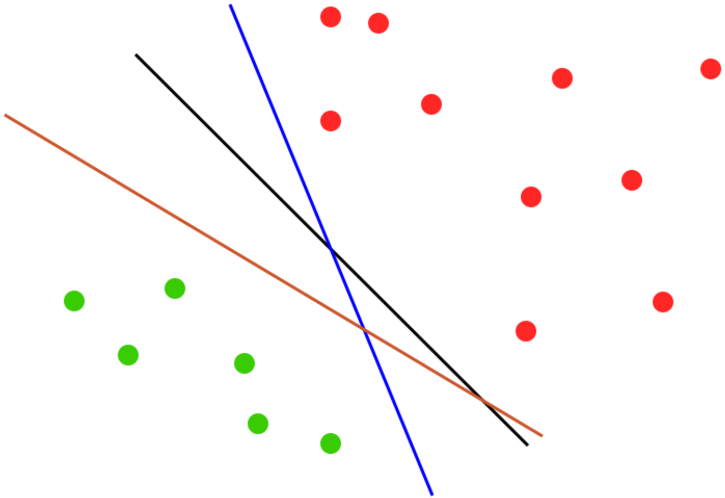
Task

- Estimate y given x at a new location.
- Modification: find a function $f(x)$ that does the task.

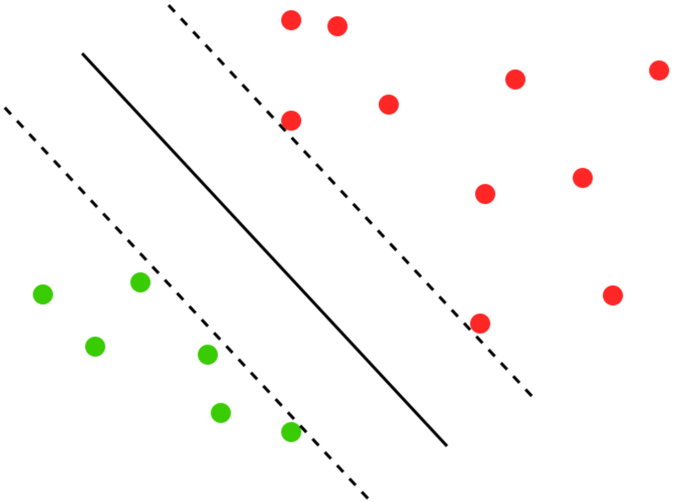


NATIONAL ICT AUSTRALIA LIMITED

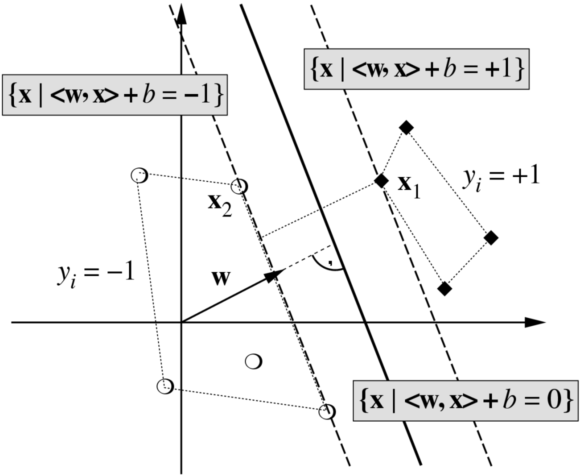
So Many Solutions



One to rule them all . . .



Optimal Separating Hyperplane



Note:

$$\langle \mathbf{w}, \mathbf{x}_1 \rangle + b = +1$$
$$\langle \mathbf{w}, \mathbf{x}_2 \rangle + b = -1$$

$$\Rightarrow \quad \langle \mathbf{w}, (\mathbf{x}_1 - \mathbf{x}_2) \rangle = 2$$

$$\Rightarrow \quad \left\langle \frac{\mathbf{w}}{\|\mathbf{w}\|}, (\mathbf{x}_1 - \mathbf{x}_2) \right\rangle = \frac{2}{\|\mathbf{w}\|}$$

Optimization Problem

Margin to Norm

- Separation of sets is given by $\frac{2}{\|w\|}$ so maximize that.
- Equivalently minimize $\frac{1}{2}\|w\|$.
- Equivalently minimize $\frac{1}{2}\|w\|^2$.

Constraints

- Separation with margin, i.e.

$$\begin{aligned} \langle w, x_i \rangle + b &\geq 1 & \text{if } y_i &= 1 \\ \langle w, x_i \rangle + b &\leq -1 & \text{if } y_i &= -1 \end{aligned}$$

- Equivalent constraint

$$y_i(\langle w, x_i \rangle + b) \geq 1$$

Optimization Problem

Mathematical Programming Setting

Combining the above requirements we obtain

$$\begin{aligned} &\text{minimize} && \frac{1}{2}\|w\|^2 \\ &\text{subject to} && y_i(\langle w, x_i \rangle + b) - 1 \geq 0 \text{ for all } 1 \leq i \leq m \end{aligned}$$

Properties

- Problem is convex
- Hence it has unique minimum
- Efficient algorithms for solving it exist

Lagrange Function

Objective Function $\frac{1}{2}\|w\|^2$.

Constraints $c_i(w, b) := 1 - y_i(\langle w, x_i \rangle + b) \leq 0$

Lagrange Function

$$\begin{aligned} L(w, b, \alpha) &= \text{PrimalObjective} + \sum_i \alpha_i c_i \\ &= \frac{1}{2}\|w\|^2 + \sum_{i=1}^{m} \alpha_i (1 - y_i(\langle w, x_i \rangle + b)) \end{aligned}$$

Saddle Point Condition

Derivatives of L with respect to w and b must vanish.

Support Vector Machines

Optimization Problem

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

$$\text{subject to } \sum_{i=1}^{m}\alpha_i y_i = 0 \text{ and } \alpha_i \geq 0$$

Support Vector Expansion

$$w = \sum_i \alpha_i y_i x_i \text{ and hence } f(x) = \sum_{i=1}^{m}\alpha_i y_i \langle x_i, x\rangle + b$$

Kuhn Tucker Conditions

$$\alpha_i(1 - y_i(\langle x_i, x\rangle + b)) = 0$$

Proof (optional)

Lagrange Function

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

Saddlepoint condition

$$\partial_w L(w, b, \alpha) = w - \sum_{i=1}^{m} \alpha_i y_i x_i \quad = 0 \iff \quad w = \sum_{i=1}^{m} \alpha_i y_i x_i$$

$$\partial_b L(w, b, \alpha) = -\sum_{i=1}^{m} \alpha_i y_i x_i \quad = 0 \iff \quad \sum_{i=1}^{m} \alpha_i y_i = 0$$

To obtain the dual optimization problem we have to substitute the values of w and b into L. Note that the dual variables α_i have the constraint $\alpha_i \geq 0$.

Proof (optional)

Dual Optimization Problem

After substituting in terms for b, w the Lagrange function becomes

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

$$\text{subject to } \sum_{i=1}^{m}\alpha_i y_i = 0 \text{ and } \alpha_i \geq 0 \text{ for all } 1 \leq i \leq m$$

Practical Modification

Need to **maximize** dual objective function. Rewrite as

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

subject to the above constraints.

Support Vector Expansion

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

- w is given by a linear combination of training patterns x_i. **Independent of the dimensionality of** x.
- w depends on the Lagrange multipliers α_i.

Kuhn-Tucker-Conditions

- At optimal solution $\mathrm{Constraint} \cdot \mathrm{Lagrange\ Multiplier} = 0$
- In our context this means

$$\alpha_i(1 - y_i(\langle w, x_i\rangle + b)) = 0.$$

Equivalently we have

$$\alpha_i \neq 0 \Longleftrightarrow y_i\left(\langle w, x_i\rangle + b\right) = 1$$

Only points at the decision boundary can contribute to the solution.

Mini Summary

Linear Classification

- Many solutions
- Optimal separating hyperplane
- Optimization problem

Support Vector Machines

- Quadratic problem
- Lagrange function
- Dual problem

Interpretation

- Dual variables and SVs
- SV expansion
- Hard margin and infinite weights

Kernels

Nonlinearity via Feature Maps

Replace x_i by $\Phi(x_i)$ in the optimization problem.

Equivalent optimization problem

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

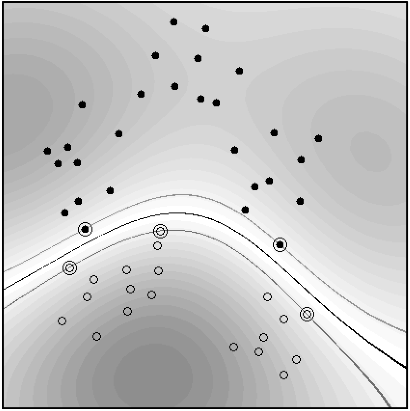
$$\text{subject to } \sum_{i=1}^{m} \alpha_i y_i = 0 \text{ and } \alpha_i \geq 0$$

Decision Function

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

$$f(x) = \langle w, \Phi(x) \rangle + b = \sum_{i=1}^{m} \alpha_i y_i k(x_i, x) + b.$$

Examples and Problems



Advantage

Works well when the data is noise free.

Problem

Already a single wrong observation can ruin everything — we require $y_i f(x_i) \geq 1$ for all i.

Idea

Limit the influence of individual observations by making the constraints less stringent (introduce slacks).

Optimization Problem (Soft Margin)

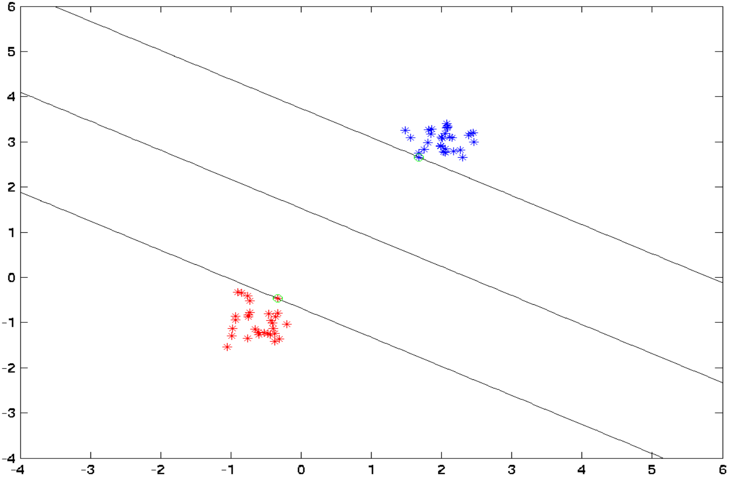
Recall: Hard Margin Problem

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|w\|^2 \\ \text{subject to} \quad & y_i(\langle w, x_i \rangle + b) - 1 \geq 0 \end{aligned}$$

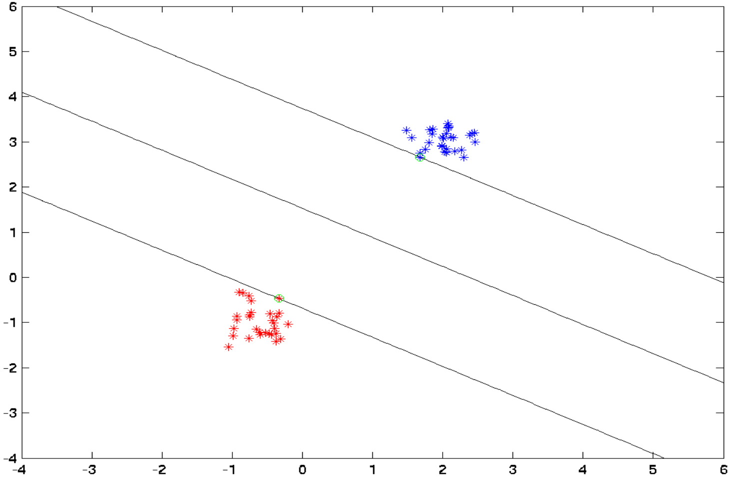
Softening the Constraints

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

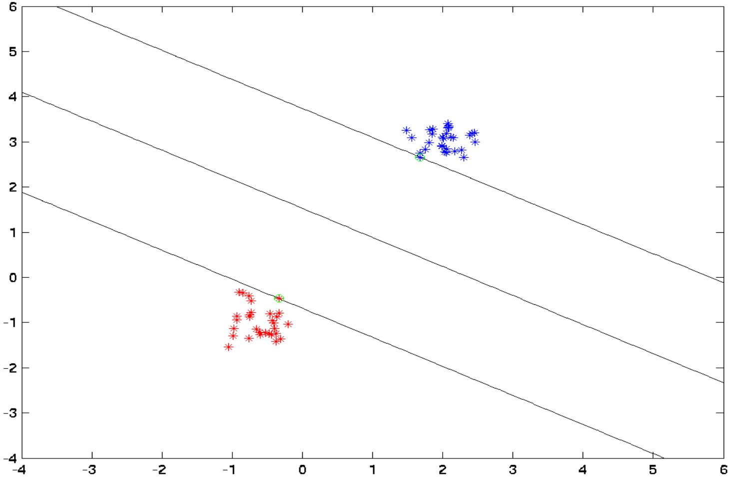
Linear SVM $C = 1$



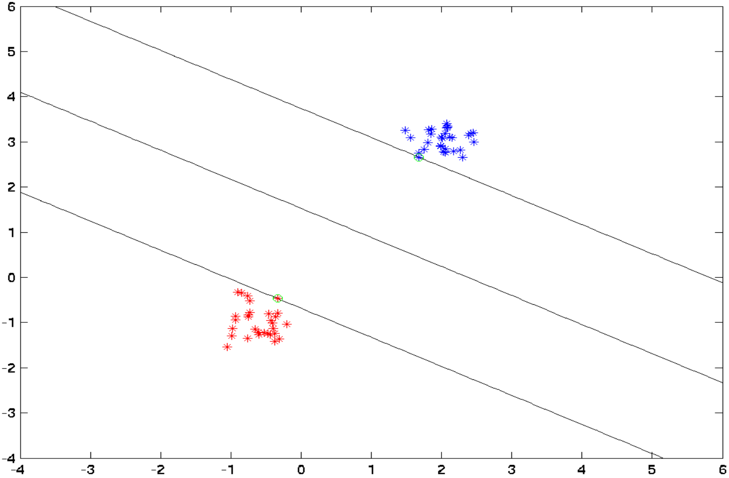
Linear SVM $C = 2$



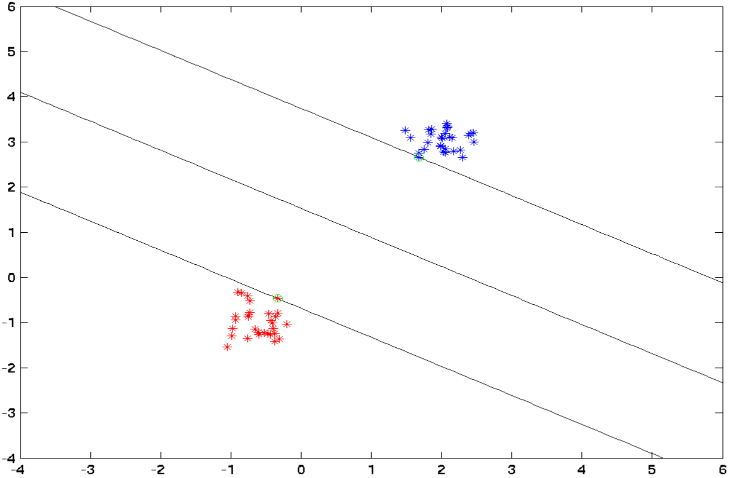
Linear SVM $C = 5$



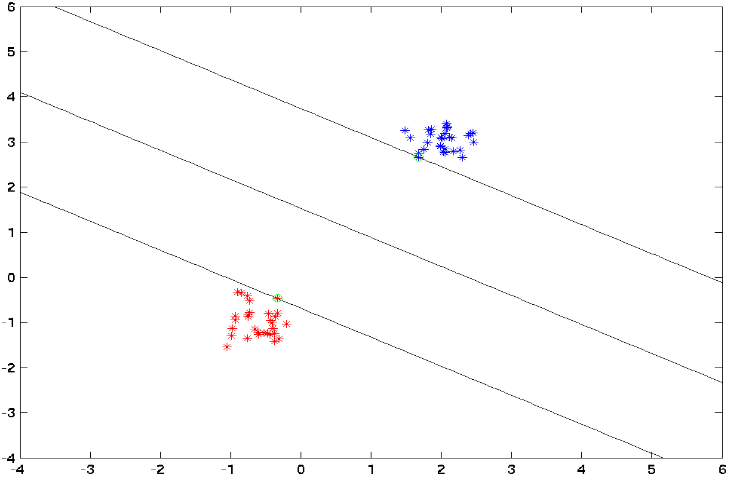
Linear SVM $C = 10$



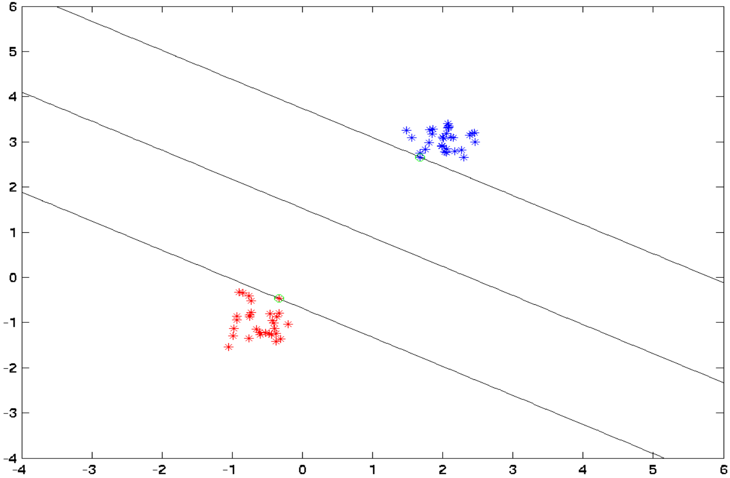
Linear SVM $C = 20$



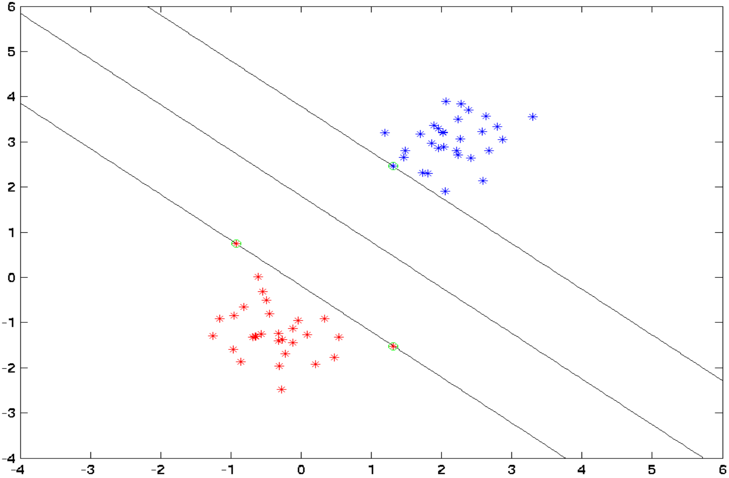
Linear SVM $C = 50$



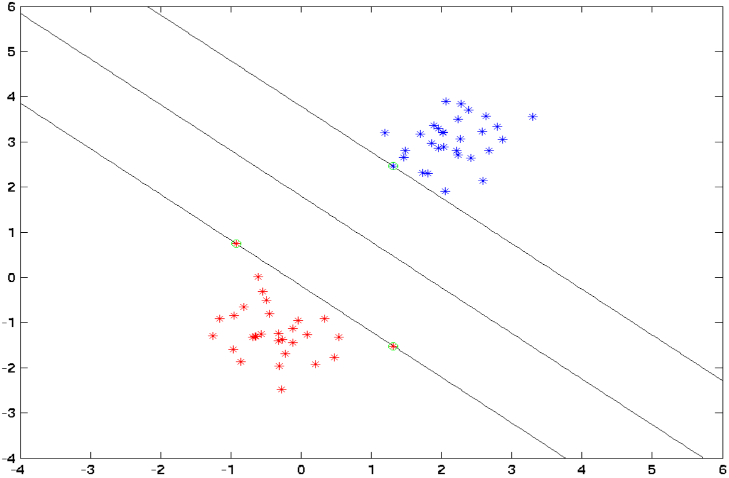
Linear SVM $C = 100$



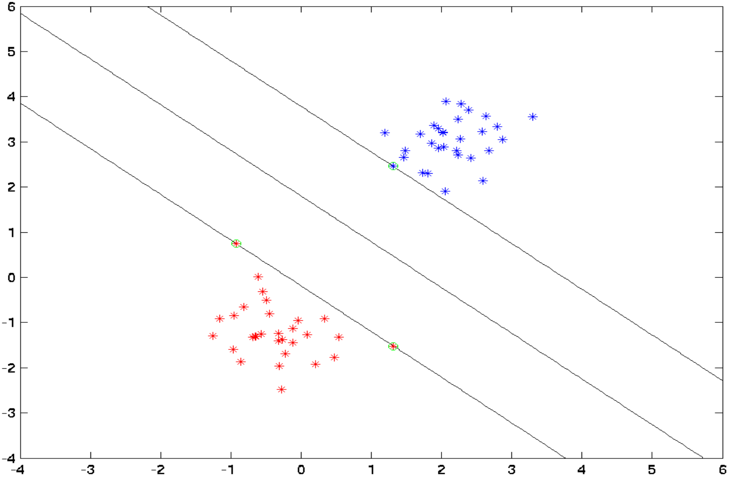
Linear SVM $C = 1$



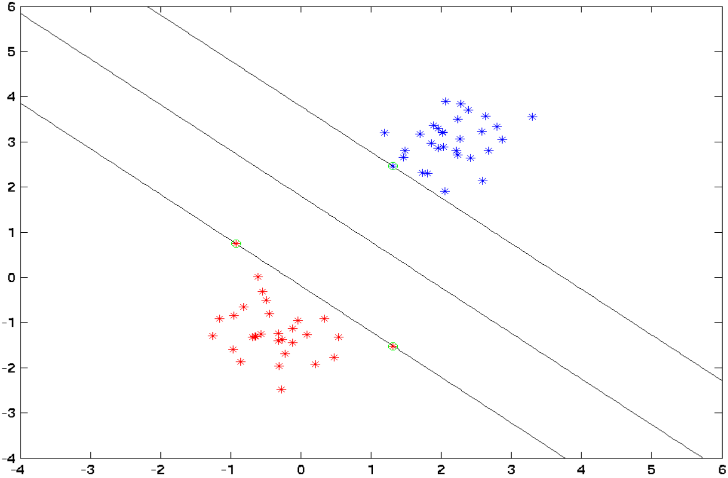
Linear SVM $C = 2$



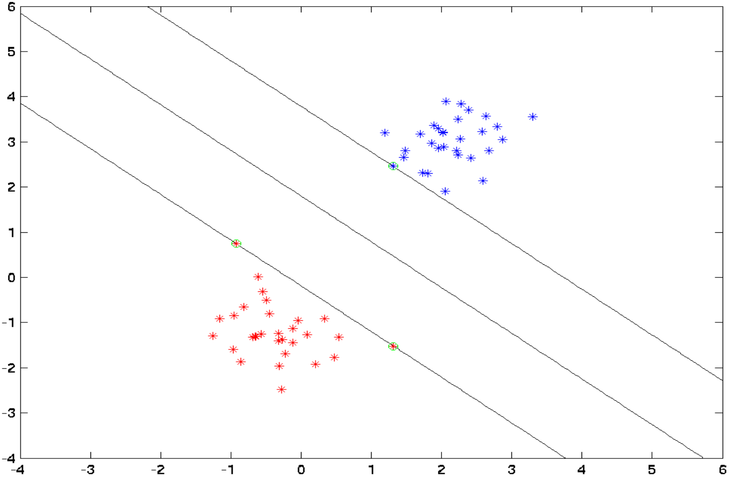
Linear SVM $C = 5$



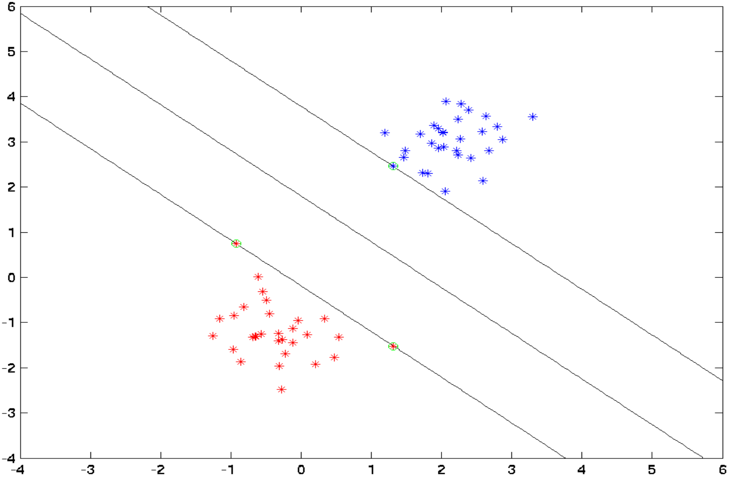
Linear SVM $C = 10$



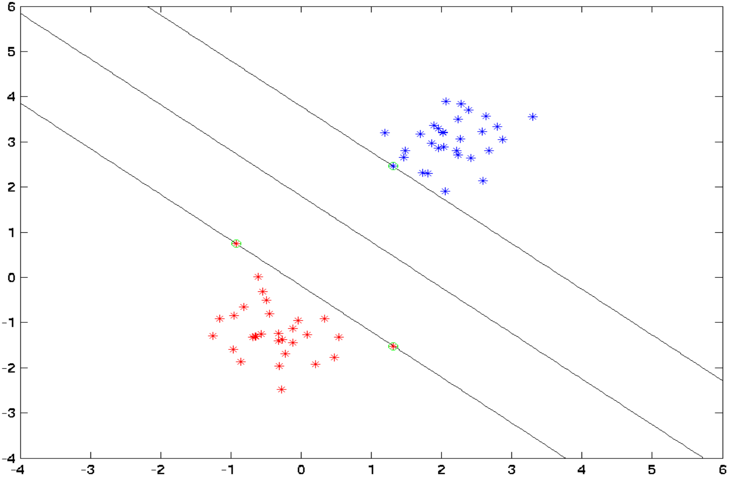
Linear SVM $C = 20$



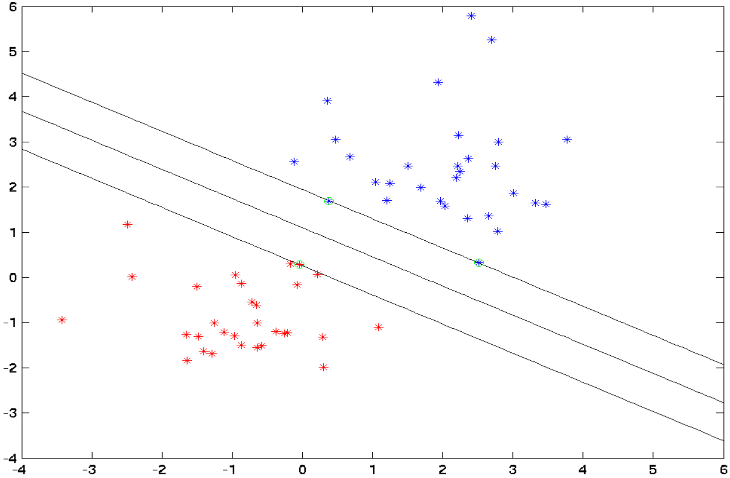
Linear SVM $C = 50$



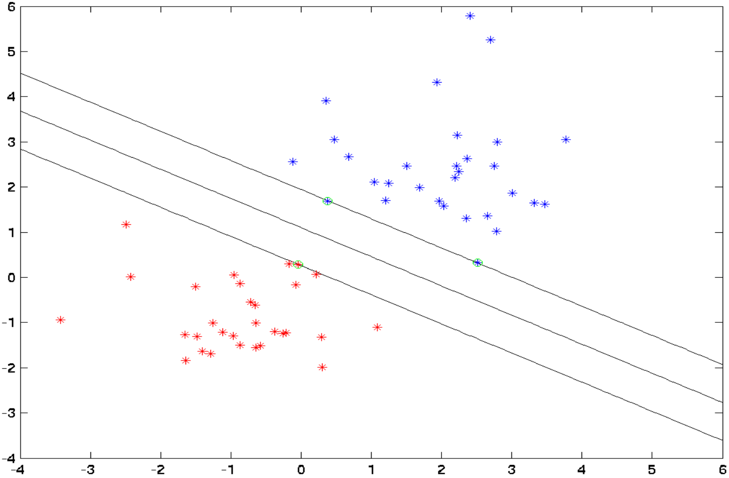
Linear SVM $C = 100$



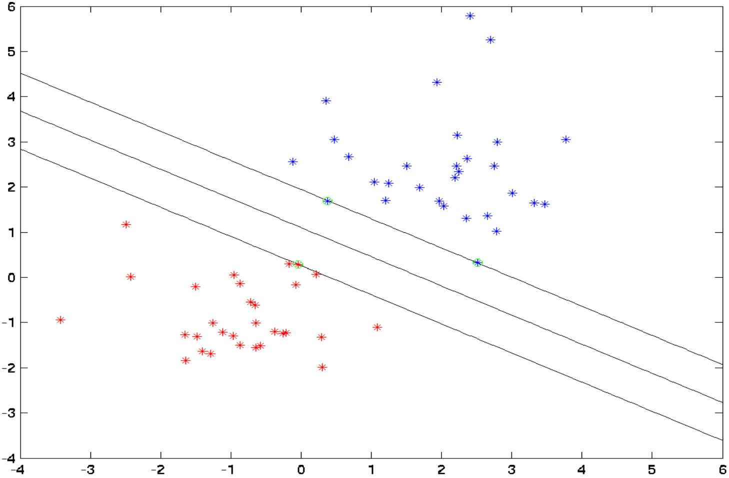
Linear SVM $C = 1$



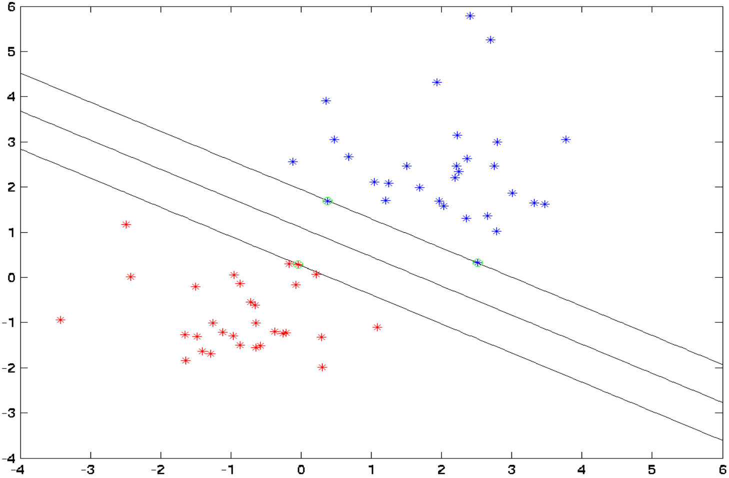
Linear SVM $C = 2$



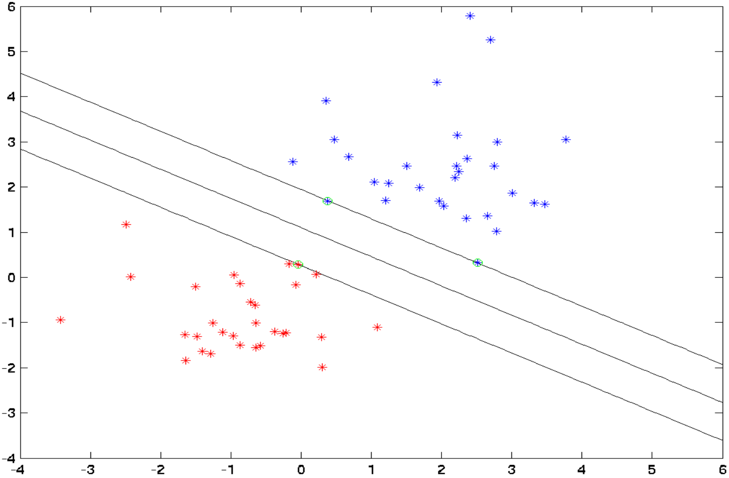
Linear SVM $C = 5$



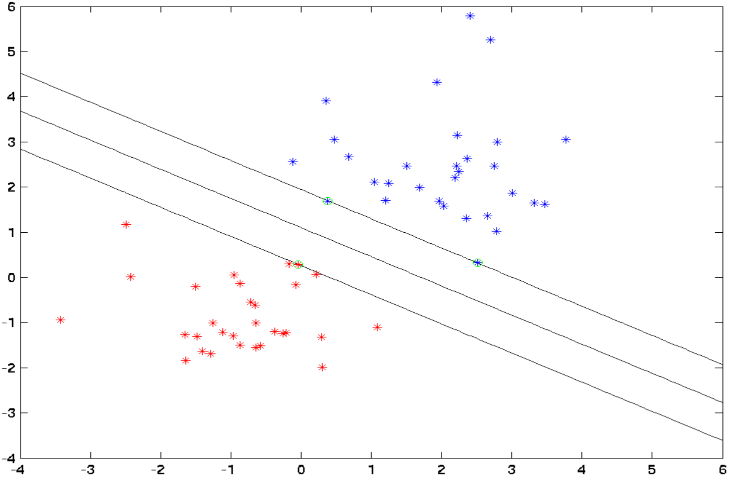
Linear SVM $C = 10$



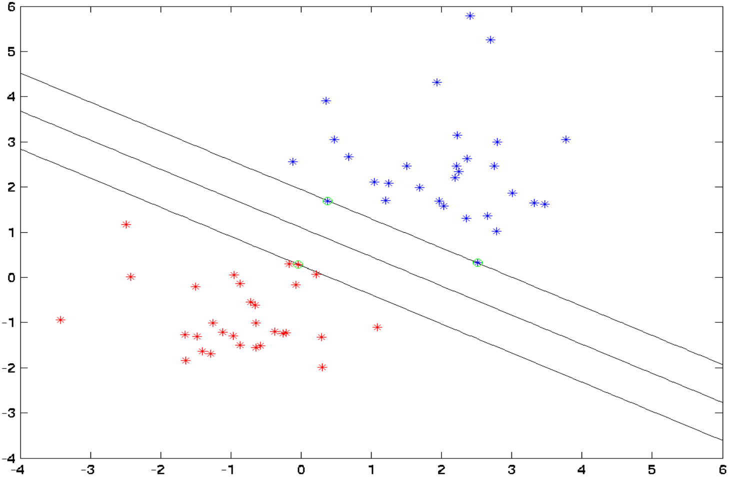
Linear SVM $C = 20$



Linear SVM $C = 50$

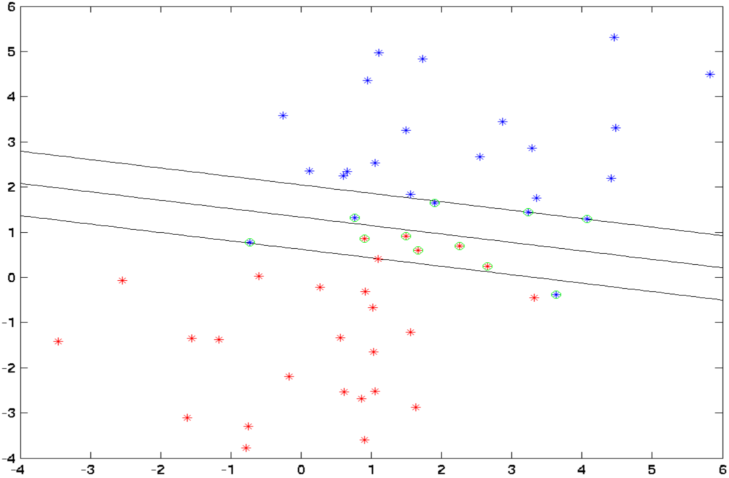


Linear SVM $C = 100$

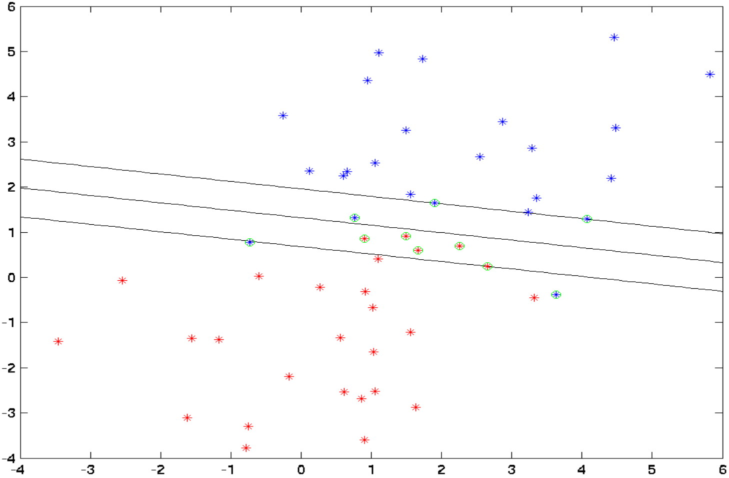


NATIONAL ICT AUSTRALIA LIMITED

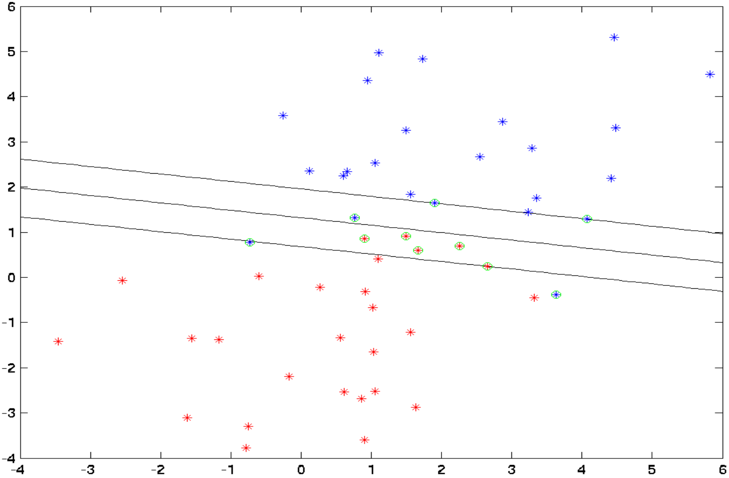
Linear SVM $C = 1$



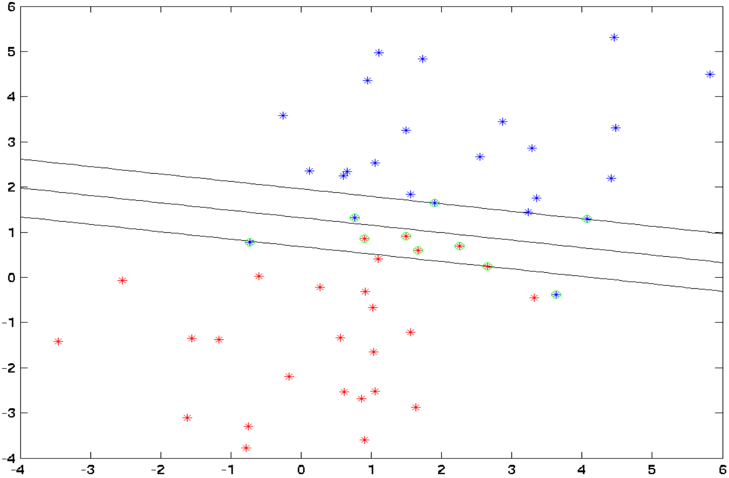
Linear SVM $C = 2$



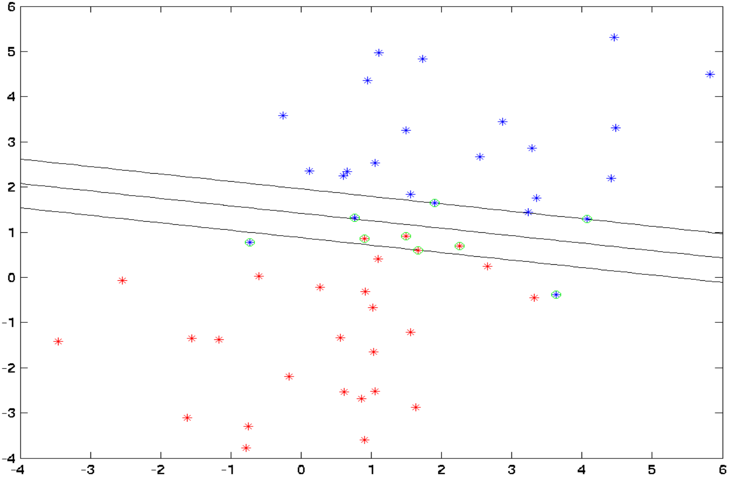
Linear SVM $C = 5$



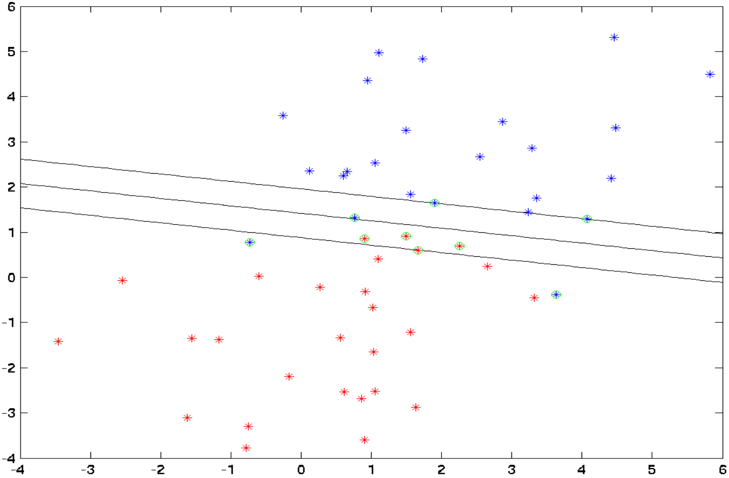
Linear SVM $C = 10$



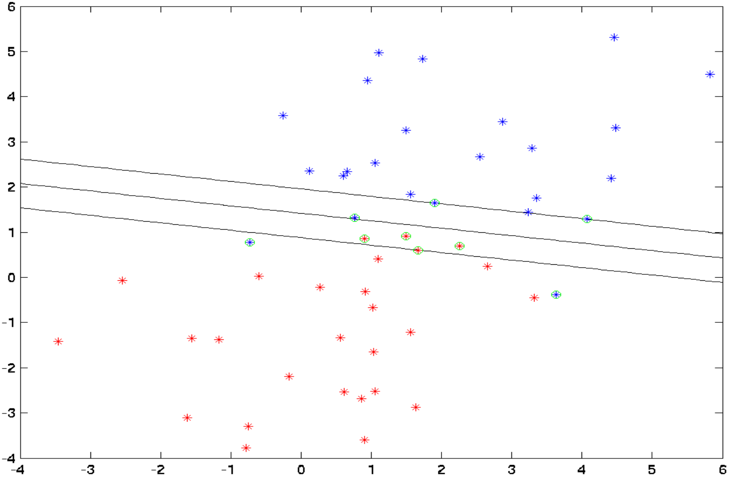
Linear SVM $C = 20$



Linear SVM $C = 50$



Linear SVM $C = 100$



Insights

Changing C

- For clean data C doesn't matter much.
- For noisy data, large C leads to narrow margin (SVM tries to do a good job at separating, even though it isn't possible)

Noisy data

- Clean data has few support vectors
- Noisy data leads to data in the margins
- More support vectors for noisy data

Python pseudocode

SVM Classification

```
import elefant.kernels.vector
# linear kernel
k = elefant.kernels.vector.CLinearKernel()
# Gaussian RBF kernel
k = elefant.kernels.vector.CGaussKernel(rbf)

import elefant.estimation.svm.svmclass as
svmclass
svm = svmclass.SVC(C, kernel=k)

alpha, b = svm.Train(x, y)
ytest = svm.Test(xtest)
```

Dual Optimization Problem

Optimization Problem

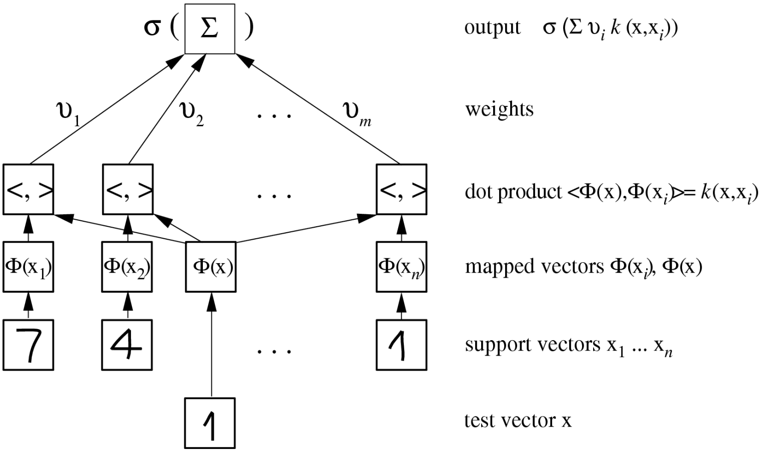
$$\text{minimize } \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j k(x_i, x_j) - \sum_{i=1}^{m} \alpha_i$$

$$\text{subject to } \sum_{i=1}^{m} \alpha_i y_i = 0 \text{ and } C \geq \alpha_i \geq 0 \text{ for all } 1 \leq i \leq m$$

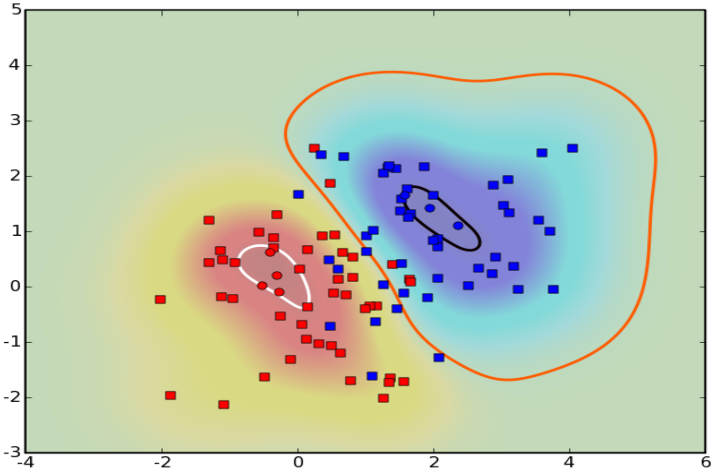
Interpretation

- Almost same optimization problem as before
- Constraint on weight of each α_i (bounds influence of pattern).
- Efficient solvers exist (more about that tomorrow).

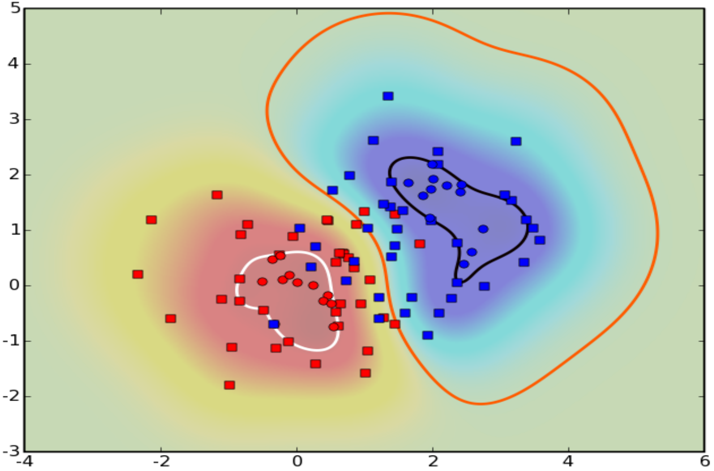
SV Classification Machine



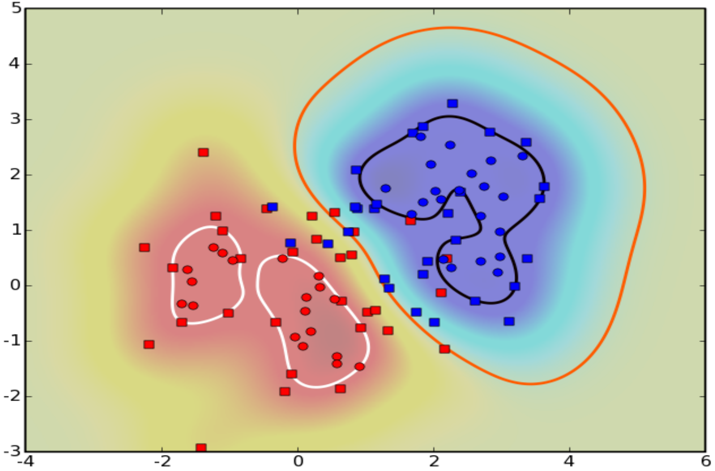
Gaussian RBF with $C = 0.1$



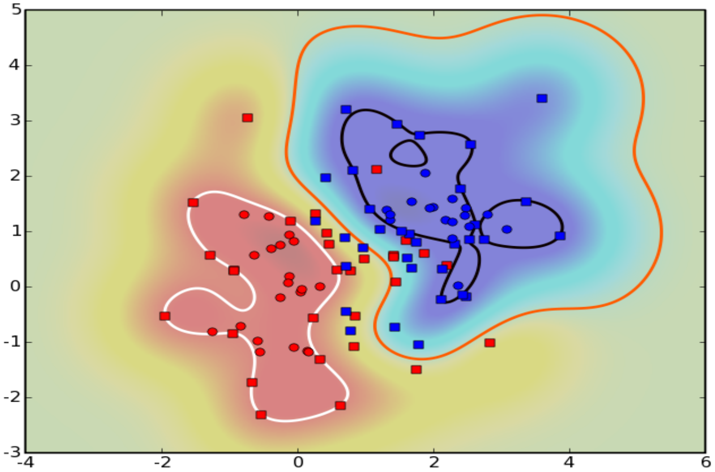
Gaussian RBF with $C = 0.2$



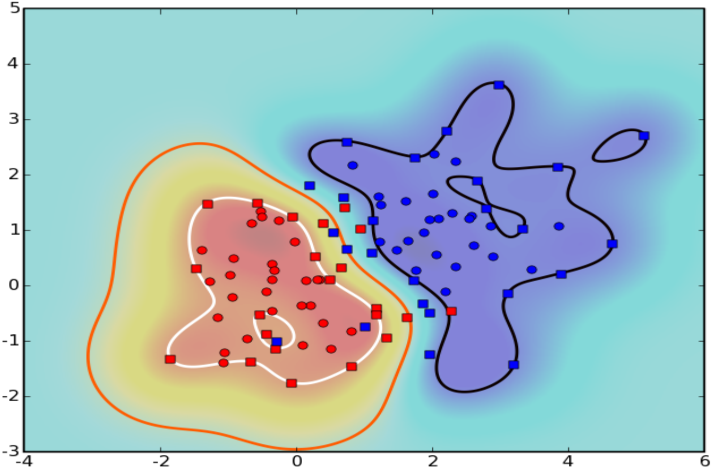
Gaussian RBF with $C = 0.4$



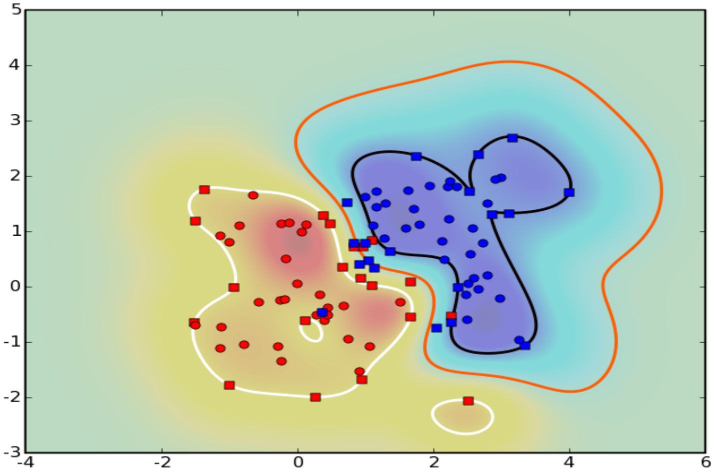
Gaussian RBF with $C = 0.8$



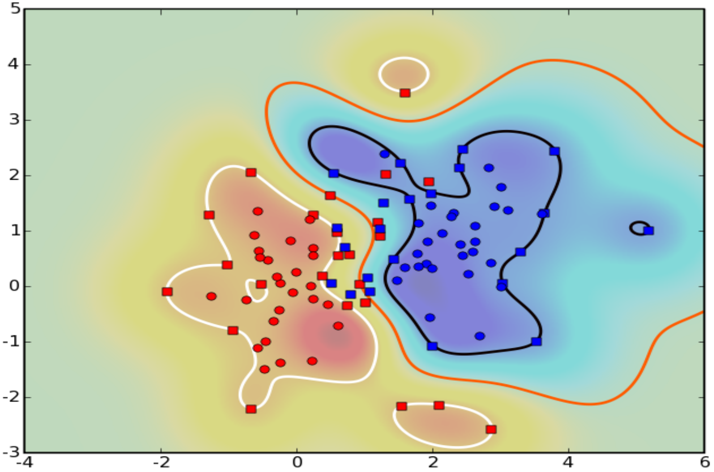
Gaussian RBF with $C = 1.6$



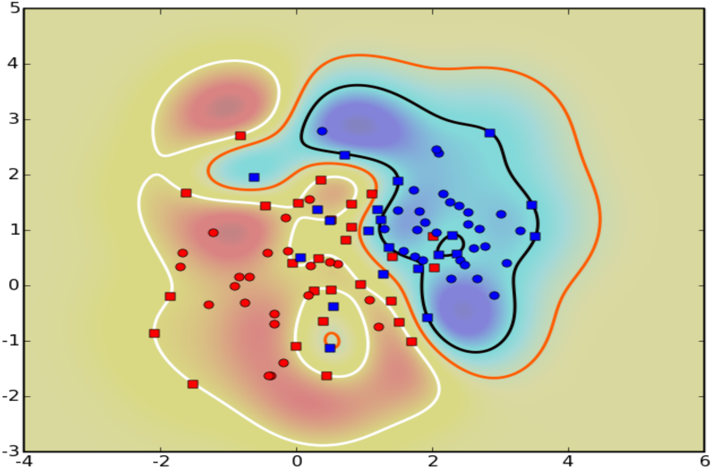
Gaussian RBF with $C = 3.2$



Gaussian RBF with $C = 6.4$



Gaussian RBF with $C = 12.8$



Insights

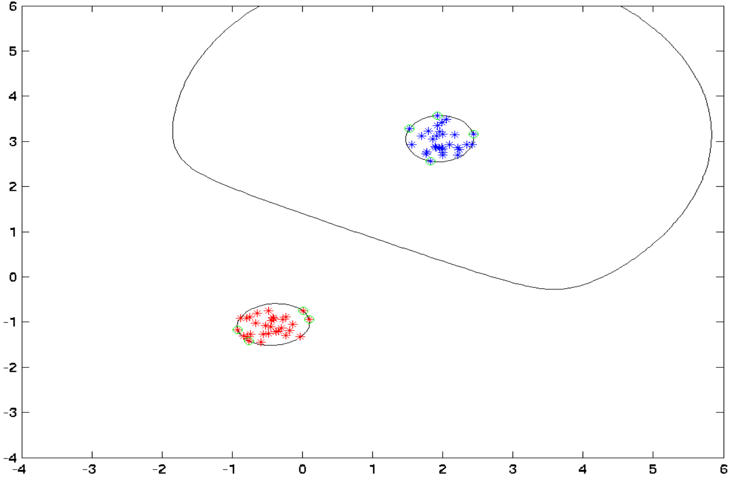
Changing C

- For clean data C doesn't matter much.
- For noisy data, large C leads to more complicated margin (SVM tries to do a good job at separating, even though it isn't possible)
- Overfitting for large C

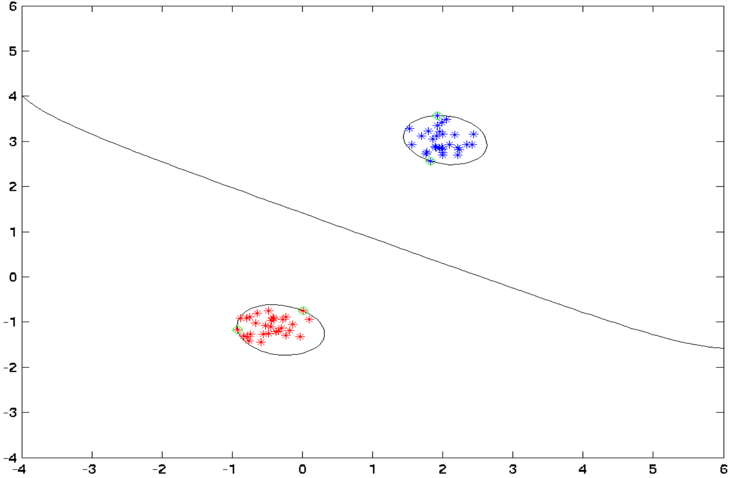
Noisy data

- Clean data has few support vectors
- Noisy data leads to data in the margins
- More support vectors for noisy data

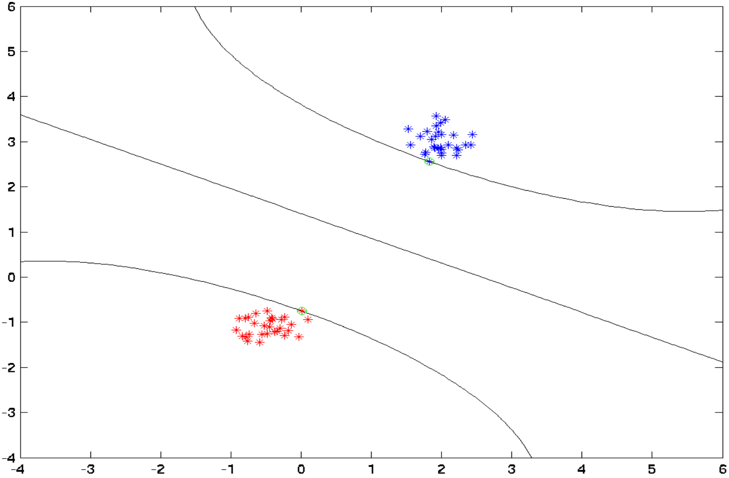
Gaussian RBF with $\sigma = 1$



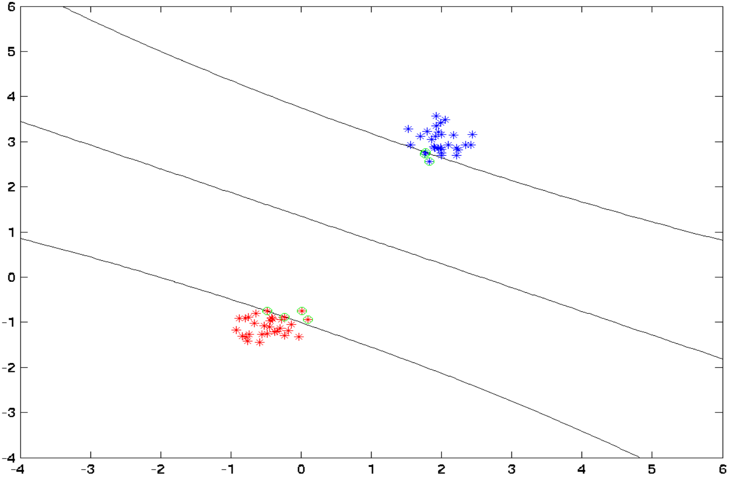
Gaussian RBF with $\sigma = 2$



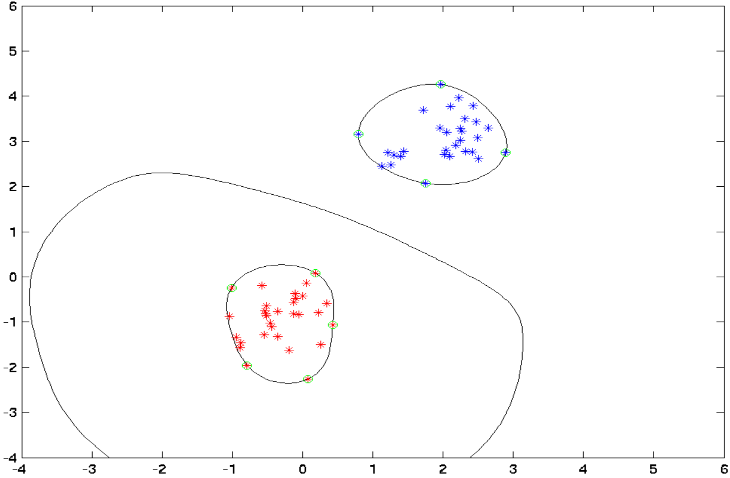
Gaussian RBF with $\sigma = 5$



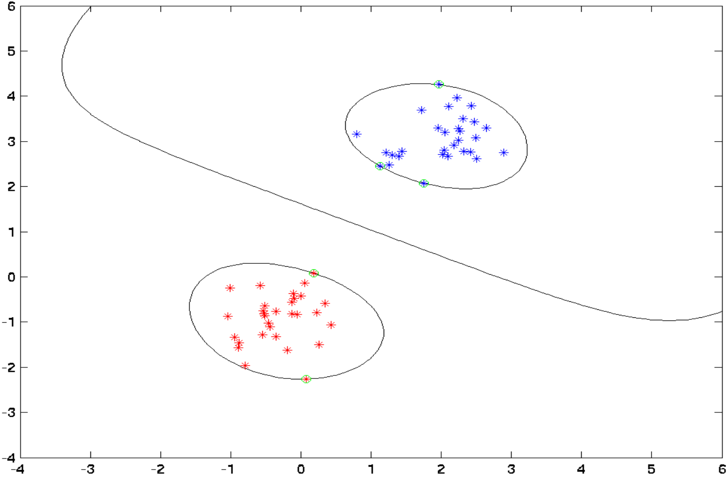
Gaussian RBF with $\sigma = 10$



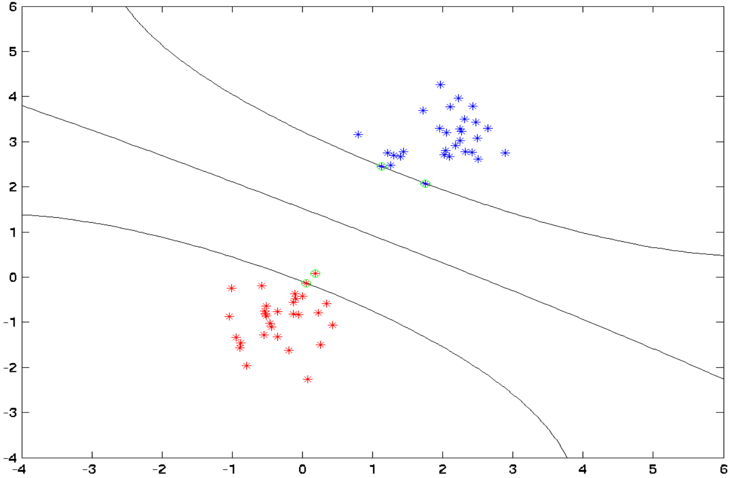
Gaussian RBF with $\sigma = 1$



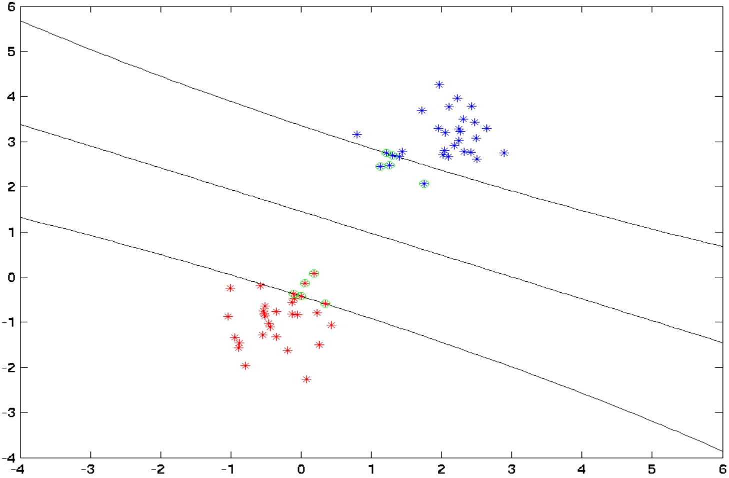
Gaussian RBF with $\sigma = 2$



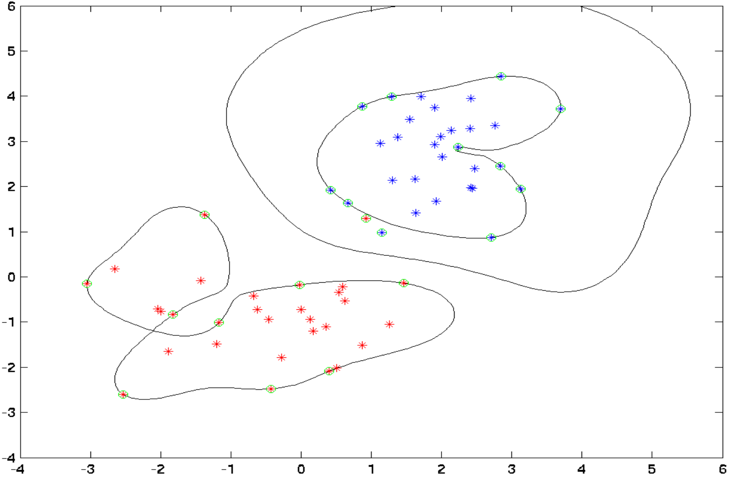
Gaussian RBF with $\sigma = 5$



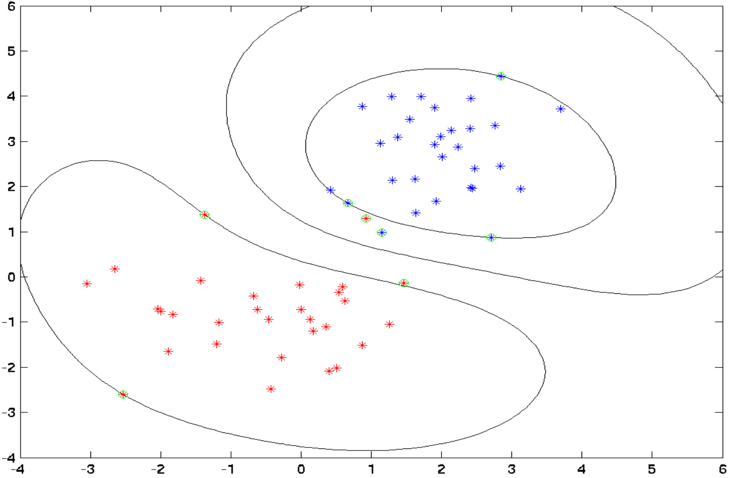
Gaussian RBF with $\sigma = 10$



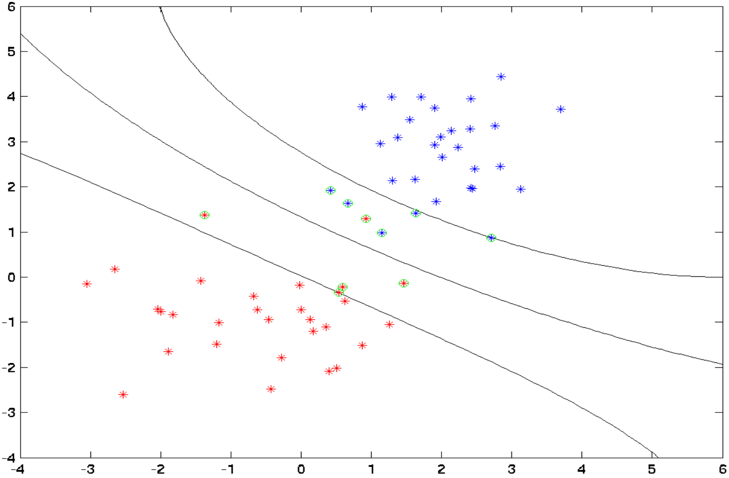
Gaussian RBF with $\sigma = 1$



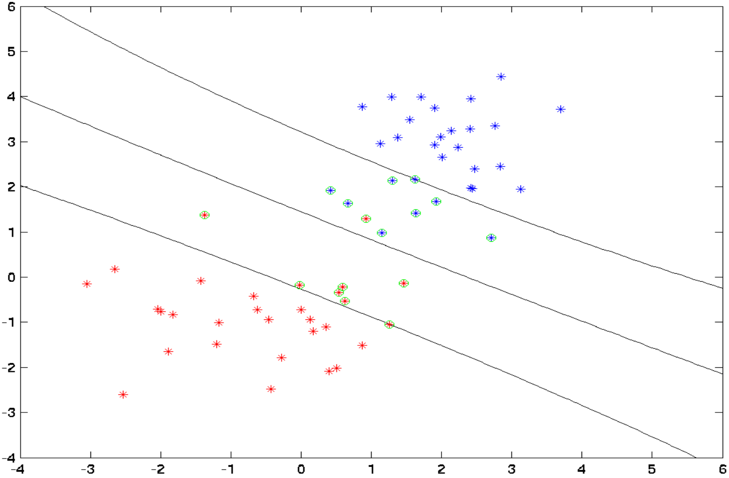
Gaussian RBF with $\sigma = 2$



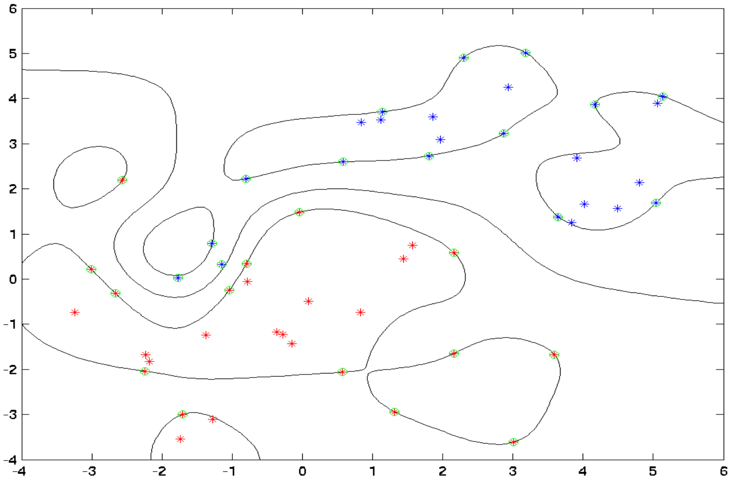
Gaussian RBF with $\sigma = 5$



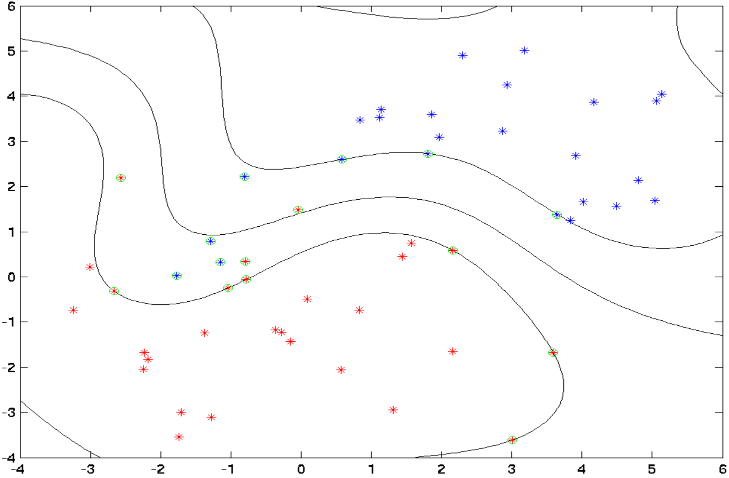
Gaussian RBF with $\sigma = 10$



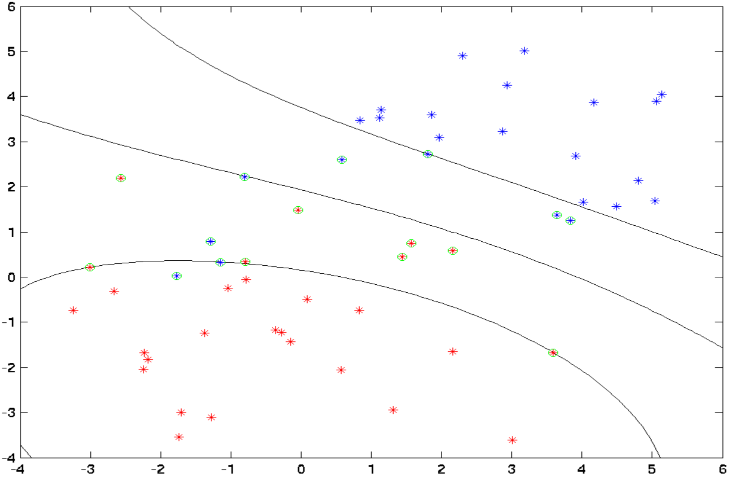
Gaussian RBF with $\sigma = 1$



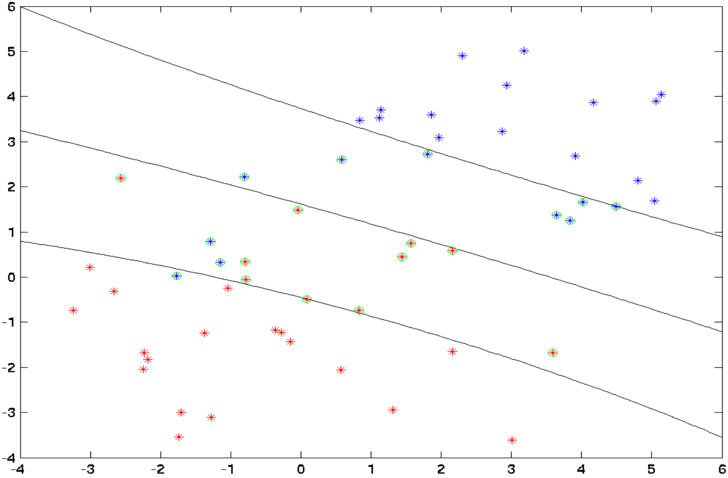
Gaussian RBF with $\sigma = 2$



Gaussian RBF with $\sigma = 5$



Gaussian RBF with $\sigma = 10$



Insights

Changing σ

- For clean data σ doesn't matter much.
- For noisy data, small σ leads to more complicated margin (SVM tries to do a good job at separating, even though it isn't possible)
- Lots of overfitting for small σ

Noisy data

- Clean data has few support vectors
- Noisy data leads to data in the margins
- More support vectors for noisy data

Summary

Support Vector Machine

- Problem definition
- Geometrical picture
- Optimization problem

Optimization Problem

- Hard margin
- Convexity
- Dual problem
- Soft margin problem