Support Vector Machines - SVM & RVM

Henrik I. Christensen

Robotics & Intelligent Machines @ GT
Georgia Institute of Technology,
Atlanta, GA 30332-0280
hic@cc.gatech.edu

Outline

Introduction

- Last time we talked about Kernels and Memory Based Models
- Estimate the full GRAM matrix can pose a major challenge
- Desirable to store only the “relevant” data
- Two possible solutions discussed
  1. Support Vector Machines (Vapnik, et al.)
  2. Relevance Vector Machines
- Main difference in how posterior probabilities are handled
- Small robotics example to show SVM performance
- Relevance Vector Machines is the probabilistic equivalent

Outline

Maximum Margin Classifiers - Preliminaries

- Lets initially consider a linear two-class problems

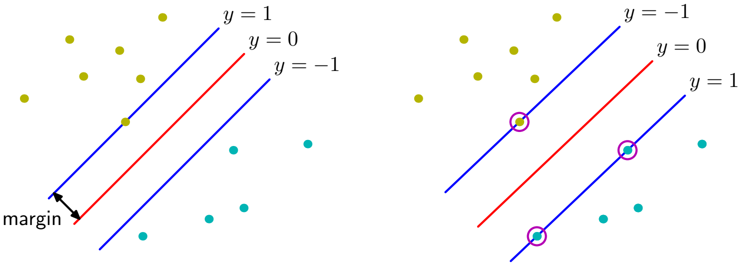
$$y(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b$$

with $\phi(.)$ being a feature space transformation and b is the bias factor

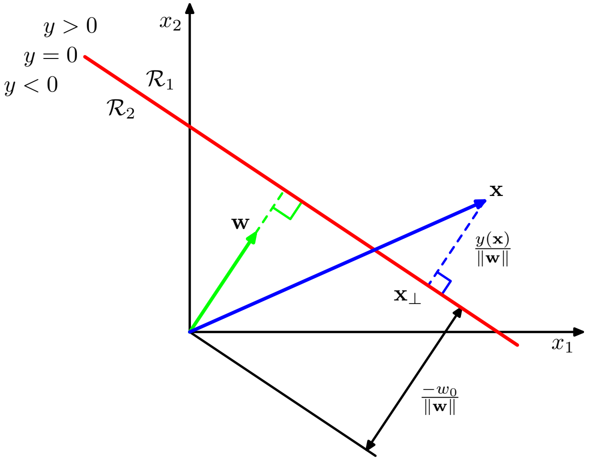
- Given a training dataset $\mathbf{x}_i, \quad i \in \{1...N\}$
- Target values $t_i, \quad i \in \{1...N\}, \quad t_i \in \{-1, 1\}$
- Assume for now that there is a linear solution to the problem

The objective

- The objective here is to optimize the margin
- Let's just keep the points at the margin



Recap distances and metrics



The objective function

- We know that $y(x)$ and t are supposed to have the same sign so that $y(x)t > 0$, i.e.

$$\frac{t_n y(\mathbf{x}_n)}{||\mathbf{w}||} = \frac{t_n(\mathbf{w}^T\phi(\mathbf{x}_n) + b)}{||\mathbf{w}||}$$

- The solution is then

$$\arg\max_{w,b}\left\{\frac{1}{||\mathbf{w}||}\min_n\left[t_n(\mathbf{w}^T\phi(\mathbf{x}_n) + b)\right]\right\}$$

- We can scale $\mathbf{w}$ and b without loss of generality.
- Scale parameters to make the key vector points

$$t_n\left(\mathbf{w}^T\phi(\mathbf{x}_n) + b\right) = 1$$

- Then for all data points it is true

$$t_n\left(\mathbf{w}^T\phi(\mathbf{x}_n) + b\right) \geq 1$$

Parameter estimation

- We need to optimize $||\mathbf{w}||^{-1}$ which can be seen as minimizing $||\mathbf{w}||^2$ subject to the margin requirements
- In Lagrange terms this is then

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{n=1}^{N} a_n \left\{ t_n \left(\mathbf{w}^T \phi(\mathbf{x}_n) + b \right) - 1 \right\}$$

- Analyzing partial derivatives gives us

$$\begin{aligned} \mathbf{w} &= \sum_{n=1}^{N} a_n t_n \phi(\mathbf{x}_n) \\ 0 &= \sum_{n=1}^{N} a_n t_n \end{aligned}$$

Parameter estimation

- Eliminating **w** and b from the objective function we have

$$L(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m)$$

- This is a quadratic optimization problem - see in a minute
- We can evaluate new points using the form

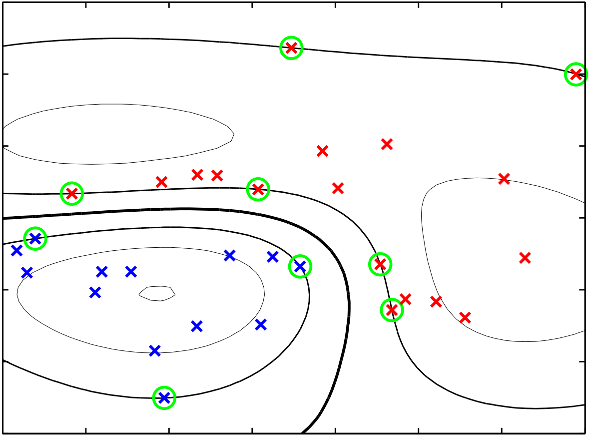
$$y(\mathbf{x}) = \sum_{n=1}^{N} a_n t_n k(\mathbf{x}, \mathbf{x}_n)$$

Estimation of the bias

- Once **w** has been estimated we can use that for estimation of the bias

$$b = \frac{1}{N_S} \sum_{n \in S} \left(t_n - \sum_{m \in S} a_m t_m k(\mathbf{x}_n, \mathbf{x}_m) \right)$$

Illustrative Synthetic Example



Status

- We have formulated the objective function
- Still not clear how we will solve it!
- We have assumed the classes are separable
- How about more messy data?

Overlapping class distributions

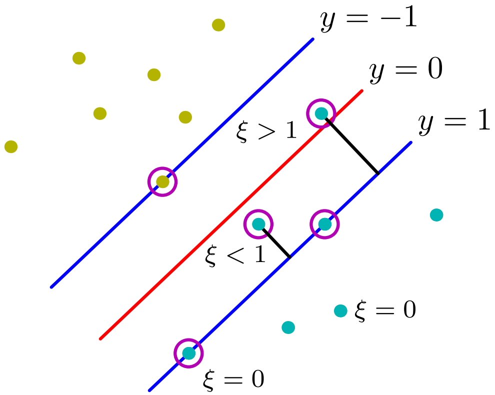
- Assume some data cannot be correctly classified
- Lets define a margin distance

$$\xi_n = |t_n - y(\mathbf{x}_n)|$$

Consider

1. $\xi < 0$ - correct classification
2. $\xi = 0$ - at the margin / decision boundary
3. $\xi \in [0; 1]$ between decision boundary and margin
4. $\xi \in [1; 2]$ between margin and other boundary
5. $\xi > 2$ - the point is definitely misclassified

Overlap in margin



Recasting the problem

- Optimizing not just for **w** but also for misclassification
- So we have

$$C\sum_{n=1}^{N}\xi_n + \frac{1}{2}||\mathbf{w}||$$

  where C is a regularization coefficient.
- We have a new objective function

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}||\mathbf{w}||^2 + C\sum_{n+1}^{N}\xi_n - \sum_{n=1}^{N} a_n\left\{t_n y(\mathbf{x}_n) - 1 + \xi_n\right\} - \sum_{n=1}^{N}\mu_n\xi_n$$

  where a and μ are Lagrange multipliers

Optimization

- As before we can derivate partial derivatives and find the extrema. The resulting objective function is then

$$L(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m)$$

  which is like before bit the constraints are a little different

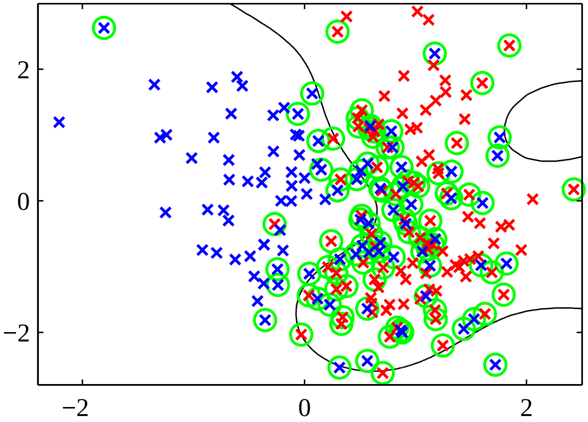
  - $0 \leq a_n \leq C$ and
  - $\sum_{n=1}^{N} a_n t_n = 0$

  which is across all training samples
- Many training samples will have $a_n = 0$ which is the same as saying they are not at the margin.

Generating a solution

- Solutions are generated through analysis of all training date
- Re-organization enable some optimization (Vapnik, 1982)
- Sequential minimal optimization is a common approach (Platt, 2000)
  - Considers pairwise interaction between Lagrange multipliers
- Complexity is somewhere between linear and quadratic

Mixed example



Outline

Multi-Class SVMs

- This far the discussion has been for the two-class problem
- How to extend to K classes?
  1. One versus the rest
  2. Hierarchical Trees - One vs One
  3. Coding the classes to generate a new problem

One versus the rest

- Training for each class with all the others serving as the non-class training samples
- Typically training is skewed - too few positives compared to negatives
- Better fit for the negatives
- The one vs all implies extra complexity in training $\approx K^2$

Tree classifier

- Organize the problem as a tree selection
- Best first elimination - select easy cases first
- Based on pairwise comparison of classes.
- Still requires extra comparison of K^2 classes

Coding new classes

- Considering optimization of an error coding
- How to minimize the criteria function to minimize errors
- Considered a generalization of voting based strategy
- Poses a larger training challenge

Outline

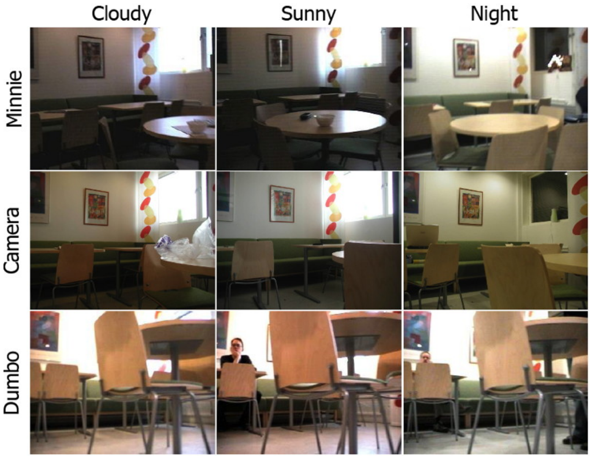
Categorization of Rooms

- Example of using SVM for room categorization
- Recognition of different types of rooms across extended periods
- Training data recorded over a period of 6 months
- Training and evaluation across 3 different settings
- Extensive evaluation

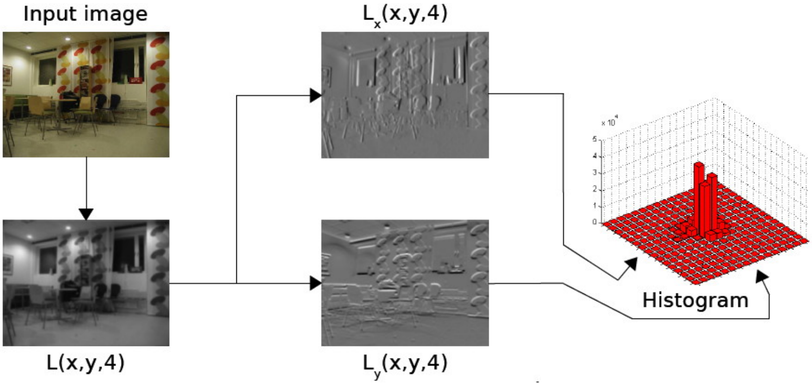
Room Categories

Training Organization

Training Organization



Preprocessing of data



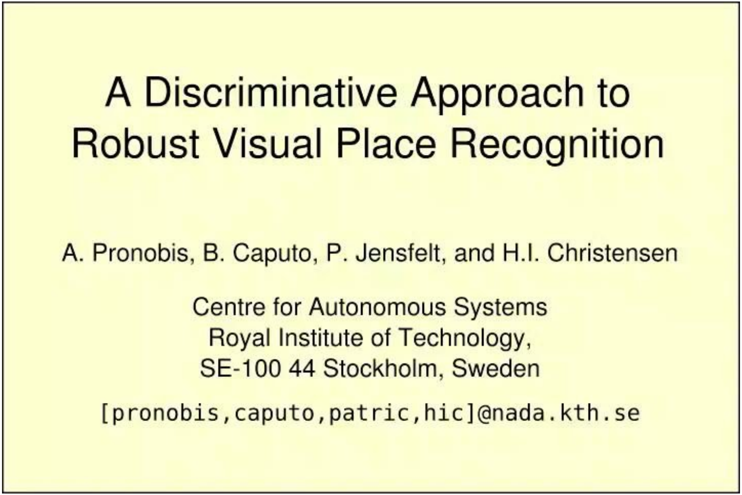
SVM details

- The system uses a χ^2 kernel.
- The kernel is widely used for histogram comparison
- The kernel is defined as

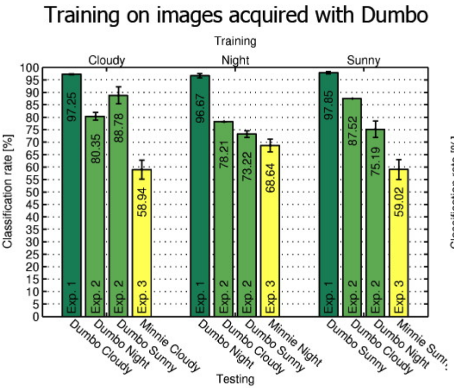
$$
\begin{aligned}
K(\mathbf{x}, \mathbf{y}) &= e^{-\gamma \chi^2(\mathbf{x}, \mathbf{y})} \\
\chi^2(\mathbf{x}, \mathbf{y}) &= \sum_i \left\{ ||x_i - y_i||^2 / ||x_i + y_i|| \right\}
\end{aligned}
$$

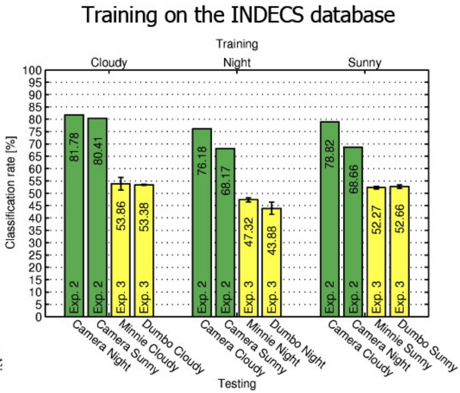
- Initially introduced by Marszalek, et al, IJCV 2007.
- Trained used "one vs the rest"

SVM results - Video

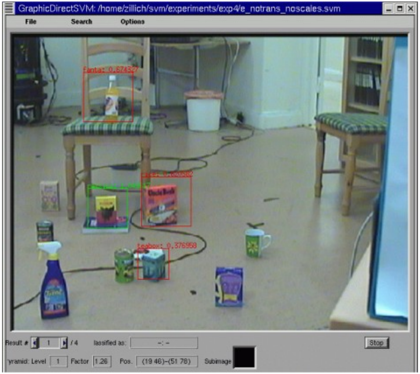


The recognition results



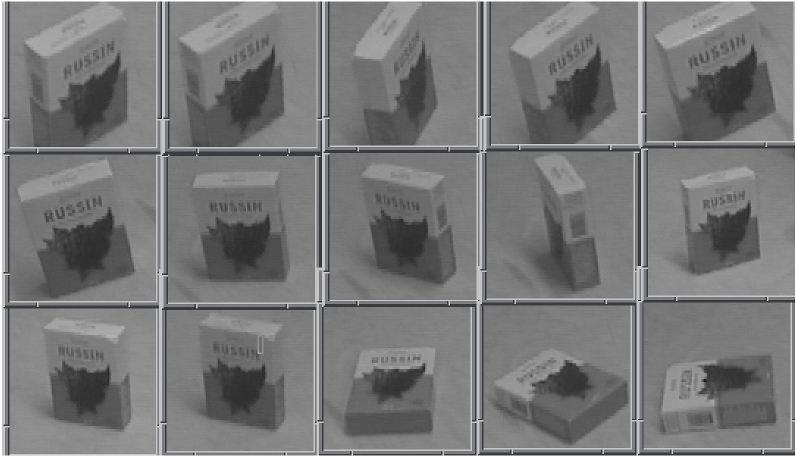


Another small example



- How to remove dependency on background? (Roobaert, 1999)

Smart use of SVMs - a "hack" with applications



Outline

RVM Introduction

- We already discussed memory based methods
- Sparse methods are directed at memory based systems with minimum (but representative) training samples
- We already discussed support vector machines
- A few challenges - ie., multi-class classification
- What if we could be more Bayesian in our formulation?

Outline

Regression model

- We are seen continuous / Bayesian regression models before

$$p(t|\mathbf{x}, \mathbf{w}, \beta) = N(t|y(\mathbf{x}), \beta^{-1})$$

- We have the linear model for fusion of data

$$y(\mathbf{x}) = \sum_{i=1}^{N} w_i \phi_i(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x})$$

- A relevance vector formulation would then be:

$$y(\mathbf{x}) = \sum_{i=1}^{N} w_i k(\mathbf{x}, \mathbf{x}_i) + b$$

The collective model

- Consider N observation vectors collected in a data matrix $\mathbf{X}$ where row i is the data vector $\mathbf{x}_i$. The corresponding target vector $\mathbf{t} = \{t_1, t_2, ..., t_N\}$ the likelihood is then:

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \prod_{i=1}^{N} p(t_i|\mathbf{x}_i, \mathbf{w}, \beta^{-1})$$

- If we consider weights to be zero-mean Gaussian we have

$$p(\mathbf{w}|\alpha) = \prod_{i=0}^{N} N(w_i|0, \alpha^{-1})$$

- ie we have different uncertainties/precision for each factor

More shuffling

- Reorganizing using the results from linear regression we get

$$p(\mathbf{w}|\mathbf{t}, \mathbf{X}, \alpha, \beta) = N(\mathbf{w}|\mathbf{m}, \mathbf{\Sigma})$$

where

$$\begin{aligned} \mathbf{m} &= \beta \mathbf{\Sigma} \mathbf{\Phi}^T \mathbf{t} \\ \mathbf{\Sigma} &= \left(\mathbf{A} + \beta \Phi^T \Phi \right)^T \end{aligned}$$

where Φ is the design matrix and $\mathbf{A} = diag(\alpha_i)$. In many cases the design matrix is the same as the GRAM matrix i.e. $\Phi_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$.

Estimation of α and β

- Using maximum likelihood we can derive estimates for α and β. We can integrate out $\mathbf{w}$

$$p(\mathbf{t}|\mathbf{X},\alpha,\beta) = \int p(\mathbf{t}|\mathbf{X},\mathbf{w},\beta)p(\mathbf{w}|\alpha)d\mathbf{w}$$

- The log likelihood is then

$$\begin{aligned} \ln p(\mathbf{t}|\mathbf{X},\alpha,\beta) &= \ln N(\mathbf{t}|0,\mathbf{C}) \\ &= -\frac{1}{2}\left\{N\ln(2\pi) + \ln|\mathbf{C}| + \mathbf{t}^T\mathbf{C}\mathbf{t}\right\} \end{aligned}$$

- where

$$\mathbf{C} = \beta^{-1}\mathbf{I} + \mathbf{\Phi}\mathbf{A}^{-1}\mathbf{\Phi}^T$$

Re-estimation of α and β

- We can then re-estimate α and β from

$$\begin{aligned} \alpha_i^{new} &= \frac{\gamma_i}{m_i^2} \\ (\beta^{new})^{-1} &= \frac{||\mathbf{t} - \mathbf{\Phi m}||^2}{N - \sum_i \gamma_i} \end{aligned}$$

- where γ_i are precision estimates defined by

$$\gamma_i = 1 - \alpha_1 \Sigma_{ii}$$

- the precision will go to zero for some of these - ie. very large uncertainty and the corresponding α values will go to zero.
- In the sense of an SVM the training data becomes irrelevant.

Regression for new data

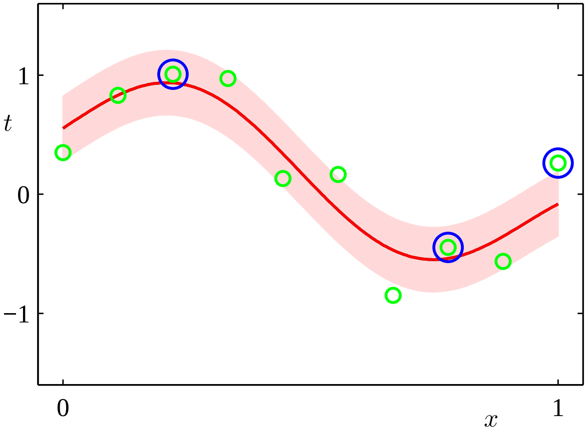
- Once hyper parameters have been estimated regression can be performed

$$p(t|\mathbf{x}, \mathbf{X}, \mathbf{t}, \alpha^*, \beta^*) = N(t|\mathbf{m}^T\phi(\mathbf{x}), \sigma^2(\mathbf{x}))$$

where

$$\sigma^2(\mathbf{x}) = (\beta^*)^{-1} + \phi(\mathbf{x})^T\mathbf{\Sigma}\phi(\mathbf{x})$$

Illustrative example



Status

- Relevance vectors are similar in style to support vectors
- Defined within a Bayesian framework
- Training requires inversion of an $(N+1) \times (N+1)$ matrix which can be (very) costly
- In general the resulting set of vectors is much smaller
- The basis functions should be chosen carefully for the training. Ie. analyze your data to fully understand what is going on.
- The criteria function is no longer a quadratic optimization problem, and convexity is not guaranteed.

Analysis of sparsity

- There is a different way to estimate the parameters that is more efficient. I.e brute force is not always optimal
- The iterative estimation of α poses a challenge, but does suggest an alternative. Consider a rewrite of the **C** matrix

$$\begin{aligned}\mathbf{C} &= \beta^{-1}\mathbf{I} + \sum_{j \neq i} \alpha_j^{-1}\phi_j\phi_j^T + \alpha_i^{-1}\phi_i\phi_i^T \\ &= C_{-i} + +\alpha_i^{-1}\phi_i\phi_i^T\end{aligned}$$

- I.e. we have made the contribution of the i'th term explicit.
- Standard linear algebra allow us to rewrite

$$\begin{aligned}det(\mathbf{c}) = |\mathbf{C}| &= |\mathbf{C}_{-i}||1 - +\alpha_i^{-1}\phi_i^T\mathbf{C}_{-i}^{-1}\phi_i| \\ \mathbf{C}^{-1} &= \mathbf{C}_{-i}^{-1} - \frac{\mathbf{C}_{-i}^{-1}\phi_i\phi_i^T\mathbf{C}_{-i}^{-1}}{\alpha_i + \phi_i^T\mathbf{C}_{-i}^{-1}\phi_i}\end{aligned}$$

The seperated log likelihood

- This allow us to rewrite the log likelihood

$$L(\alpha) = L(\alpha_{-i}) + \lambda(\alpha_i)$$

- The contribution of alpha is then

$$\lambda(\alpha_i) = \frac{1}{2}\left[\ln \alpha_i - \ln(\alpha_i + s_i) + \frac{q_i^2}{\alpha_i + s_i}\right]$$

- Here we have the complete dependency on α_i
- We have used

$$\begin{aligned} s_i &= \phi_i^T \mathbf{C}_{-i}^{-1} \phi_i \\ q_i &= \phi_i^T \mathbf{C}_{-i}^{-1} \mathbf{t} \end{aligned}$$

s_i is known as the sparsity and q_i is known as the quality of ϕ_i

Evaluation for stationary conditions

- It can be shown (see Bishop pp. 351-352)
- if $q_i^2 > s_i$ then there is a stable solution

$$\alpha_i = \frac{s_i^2}{q_i^2 - s_i}$$

- otherwise α_i goes to infinity == irrelevant

Status

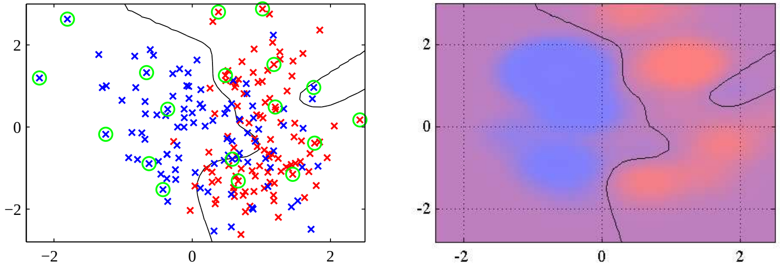
- There are efficient (non-recursive) ways to evaluate the parameters.
- The relative complexity is still significant.

Outline

Relevance vectors for classification

- For classification we can apply the same framework
- Consider the two class problem with binary targets $t \in \{0, 1\}$ then the form is
$$y(\mathbf{x}) = \sigma(\mathbf{w}^t \phi(\mathbf{x}))$$
- where $\sigma(.)$ is the logistic sigmoid function
- Closed form integration is no longer an option
- We can use the Laplace approach to estimate the mode and which in turn allow estimation of weights (α) and in term re-estimate the mode and then new values for α until convergence.

Synthetic example



Outline

Summary

- An approach to storage of “key” data for recognition/regression
- Definition of optimization to recognize data points
- The learning is fairly involved (complex)
- Basically a quadratic optimization problem
- Evaluation across all training data
- Keep the essential data
  1. Training can be costly
  2. Execution can be fast - optimized
- Multi-class cases can pose a bit of a challenge
- SVM is a fixed metric and RVM is probabilistic.