Sparse Kernel Machines

Dit-Yan Yeung

Department of Computer Science and Engineering
Hong Kong University of Science and Technology

COMP621P: Beyond Introductory Machine Learning (Spring 2008)
1 Introduction

2 Relevance Vector Machines
   - RVM for Regression
   - RVM for Classification
A limitation of many kernel-based learning algorithms is that the kernel function $k(x_n, x_m)$ must be evaluated for all possible pairs $x_n$ and $x_m$ of training points, leading to high computational requirements especially for large data sets.

It would be desirable to have kernel-based algorithms that have sparse solutions, so that predictions for new inputs depend only on the kernel function evaluated at a small subset of the training data points.
Support Vector Machine and Beyond

- Support vector machine (SVM) is a maximum margin classifier.
- SVM is a decision machine that does not provide posterior probabilities (which may be useful for making decisions).

Figure: Synthetic data from two classes in two dimensions showing contours of constant $y(x)$ obtained from an SVM having a Gaussian kernel function. Also shown are the decision boundary, margin boundaries, and support vectors.
Another kernel machine, called **relevance vector machine (RVM)**, formulated under the **Bayesian framework**, has some advantages over SVM:

- Allowing for making **probabilistic predictions**
- Allowing for using **arbitrary basis functions**, including non-Mercer kernels
- Typically having a higher degree of **sparsity**
- Automatic hyperparameter estimation.
RVM for Regression

- Linear regression model:
  - Conditional distribution for a real-valued target variable $t$, given an input vector $x$:
    \[
    p(t|x, w, \beta) = \mathcal{N}(t|y(x), \beta^{-1}),
    \]
    where $\beta = \sigma^{-2}$ is the noise precision (inverse noise variance). Note that this corresponds to an additive zero-mean Gaussian noise model with variance $\beta^{-1}$.
  - The mean is given by a linear model of the form
    \[
    y(x) = \sum_{i=1}^{M} w_i \phi_i(x) = w^T \phi(x),
    \]
    with fixed nonlinear basis functions $\phi_i(x)$, which typically includes a constant term so that the corresponding weight parameter represents a bias.
RVM for Regression

- RVM is a specific instance of this linear model in which the basis functions are given by kernels, with one kernel associated with each of the training data points:

\[ y(x) = \sum_{n=1}^{N} w_n k(x, x_n) + b, \]  

(3)

where \( b \) is a bias parameter. The model has \( N + 1 \) parameters.

- This model is similar in form to the predictive model for SVM:

\[ y(x) = \sum_{n=1}^{N} (a_n - \hat{a}_n) k(x, x_n) + b, \]

where \( a_n \) and \( \hat{a}_n \) are Lagrange multipliers, effectively defining one basis function for each training data point.
The model (3) is valid for arbitrary choices of basis function.

In contrast to SVM, there is no restriction to Mercer kernels, nor are the basis functions tied in either number or location to the training data points.

The key feature of this approach is that as well as offering good generalization performance, the inferred predictors are exceedingly sparse in that they contain relatively few nonzero parameters \( w_i \).

The majority of parameters are automatically set to zero during the learning process, giving a procedure that is effective at discerning those basis functions which are relevant for making good predictions.
Likelihood Function

- Let the $N$ observations of the input vector $\mathbf{x}$ be represented collectively by a data matrix

$$\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^T$$

and the corresponding target values be represented by

$$\mathbf{t} = (t_1, \ldots, t_N)^T.$$ 

- Likelihood function:

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \prod_{n=1}^{N} p(t_n|\mathbf{x}_n, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|y(\mathbf{x}_n), \beta^{-1}), \quad (4)$$

with $y(\mathbf{x}_n) = \mathbf{w}^T \phi(\mathbf{x}_n)$. 
Prior Distribution

- Since the number of model parameters ($w$ and $\beta$) can be as large as the number of training data points (when the model (3) is used), estimating $w$ and $\beta$ by performing maximum likelihood estimation (MLE) on (4) is expected to lead to severe over-fitting.

- While SVM imposes some additional constraint on the parameters through margin maximization, RVM, which is based on the Bayesian framework, constrains the parameters by defining an explicit prior distribution over them.

- To encode a preference for smoother (less complex) functions, the choice of a zero-mean Gaussian prior distribution is made.
Prior Distribution

- Prior distribution over parameter vector $\mathbf{w}$:

$$p(\mathbf{w}|\alpha) = \prod_{i=1}^{M} \mathcal{N}(w_i|0, \alpha_i^{-1}),$$ \hspace{1cm} (5)

where we introduce a separate precision hyperparameter $\alpha_i$ for each of the weight parameters $w_i$ instead of a single shared hyperparameter, and $\alpha = (\alpha_1, \ldots, \alpha_M)^T$.

- When we maximize some objective function (to be described) w.r.t. $\alpha$, a significant proportion of the hyperparameters go to $\infty$ and hence the corresponding weight parameters have posterior distributions concentrated at zero. The basis functions associated with these parameters therefore play no role in the predictions made by the model and so are effectively pruned out, resulting in a sparse model.
Relevance Vector Machines
RVM for Regression

Posterior Distribution

- Since both the likelihood function and the prior distribution are Gaussian, the posterior distribution is also Gaussian (conjugate prior).
- Posterior distribution over parameter vector $w$ (by Bayes’ theorem):

$$p(w|t, X, \alpha, \beta) = \frac{p(t|X, w, \beta)p(w|\alpha)}{\int p(t|X, w, \beta)p(w|\alpha)dw} = \mathcal{N}(w|m, \Sigma). \quad (6)$$

- By inspecting the corresponding terms of the exponents, we can find the mean and covariance as

$$m = \beta \Sigma \Phi^T t \quad (7)$$
$$\Sigma = \left( A + \beta \Phi^T \Phi \right)^{-1}, \quad (8)$$

where $\Phi$ is an $N \times M$ matrix with elements $\Phi_{ni} = \phi_i(x_n)$ and $A = \text{diag}(\alpha_i)$.

- In the specific case of the model (3), we have $\Phi = K$ where $K$ is the symmetric $(N + 1) \times (N + 1)$ kernel matrix with elements $k(x_n, x_m)$. 

D.Y. Yeung (CSE, HKUST)

Sparse Kernel Machines
COMP621P (Spring 2008) 12 / 40
Estimation of Hyperparameters

- The values of hyperparameters $\alpha$ and $\beta$ can be determined by evidence approximation (a.k.a. empirical Bayes or type-2 maximum likelihood).

- Marginal likelihood function (a.k.a. integrated likelihood or evidence):

$$p(t|X, \alpha, \beta) = \int p(t|X, w, \beta)p(w|\alpha)dw,$$

obtained by integrating out the weight parameters.

- Evidence approximation refers to the method of maximizing the evidence w.r.t. $\alpha$ and $\beta$.

- Since this represents the convolution of two Gaussians, it is readily evaluated to give the log marginal likelihood in the form

$$\ln p(t|X, \alpha, \beta) = \ln \mathcal{N}(t|0, C)$$

$$= -\frac{1}{2} \left\{ N \ln(2\pi) + \ln |C| + t^T C^{-1}t \right\}. \quad (10)$$
Evidence Approximation

- Using (4) and (5), we write (9) as

\[
p(t|X, \alpha, \beta) = \left( \frac{\beta}{2\pi} \right)^{N/2} \left( \frac{1}{2\pi} \right)^{M/2} \prod_{i=1}^{M} \alpha_i^{1/2} \int \exp\{-E(w)\} \, dw,
\]

where

\[
E(w) = \frac{\beta}{2} \| t - \Phi w \|^2 + \frac{1}{2} w^T A w.
\]

- Completing the square over \( w \), we get

\[
E(w) = \frac{1}{2} (w - m)^T \Sigma^{-1} (w - m) + E(t),
\]

where \( m \) and \( \Sigma \) are given by (7) and (8), respectively, and

\[
E(t) = \frac{1}{2} \left( \beta t^T t - m^T \Sigma^{-1} m \right).
\]
Evidence Approximation

- Using (12), we can evaluate the integral in (11) to obtain

\[
\int \exp\{-E(w)\} dw = \exp\{-E(t)\}(2\pi)^{M/2}|\Sigma|^{1/2}.
\]  

(14)

- Considering this as a function of \( t \), we see from (11) that we only need to deal with the factor \( \exp\{-E(t)\} \).

- In the following derivation we make use of the Woodbury identity for matrix inverses:

\[
(A + BD^{-1}C)^{-1} = A^{-1} - A^{-1}B(D + CA^{-1}B)^{-1}CA^{-1}.
\]

(15)
Evidence Approximation

- Using (7), (8) and (15), we can rewrite (13) as follows:

\[
E(t) = \frac{1}{2} \left( \beta t^T t - m^T \Sigma^{-1} m \right) \\
= \frac{1}{2} \left( \beta t^T t - \beta t^T \Phi \Sigma \Sigma^{-1} \Sigma \Phi^T t \beta \right) \\
= \frac{1}{2} t^T \left( \beta I - \beta \Phi \Sigma \Phi^T \beta \right) t \\
= \frac{1}{2} t^T \left( \beta I - \beta \Phi (A + \beta \Phi^T \Phi)^{-1} \Phi^T \beta \right) t \\
= \frac{1}{2} t^T \left( \beta^{-1} I + \Phi A^{-1} \Phi^T \right)^{-1} t.
\]

- By defining

\[ C = \beta^{-1} I + \Phi A^{-1} \Phi^T, \]

we can get the last term on the right-hand side of (10); the two preceding terms are given implicitly, as they form the normalization constant for \( p(t|X, \alpha, \beta). \)
Evidence Approximation

- Our goal is now to maximize (10) w.r.t. $\alpha$ and $\beta$.

- One approach: set the derivatives of $\ln p(t|X, \alpha, \beta)$ w.r.t. $\alpha$ and $\beta$ to zero to obtain re-estimation equations for an iterative algorithm.

- Using (11) and (14), we can write (10) as:

$$
\ln p(t|X, \alpha, \beta) = \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi) + \frac{1}{2} \sum_{i=1}^{M} \ln \alpha_i - E(t) + \frac{1}{2} \ln |\Sigma|.
$$
Evidence Approximation

- By making use of (8) and (13) and the following results on matrix derivatives:

\[
\frac{\partial}{\partial x} (A^{-1}) = -A^{-1} \frac{\partial A}{\partial x} A^{-1}
\]

\[
\frac{\partial}{\partial x} \ln |A| = \text{Tr} \left( A^{-1} \frac{\partial A}{\partial x} \right),
\]

we can take the derivatives of \( \ln p(t|X, \alpha, \beta) \) w.r.t. \( \alpha_i \), yielding

\[
\frac{\partial}{\partial \alpha_i} \ln p(t|X, \alpha, \beta) = \frac{1}{2\alpha_i} - \frac{1}{2} m_i^2 - \frac{1}{2} \Sigma_{ii}, \quad (16)
\]

where \( m_i \) is the \( i \)th component of \( m \) and \( \Sigma_{ii} \) is the \( i \)th diagonal element of \( \Sigma \).
Setting (16) to zero and re-arranging, we obtain:

$$\alpha_i = \frac{1 - \alpha_i \Sigma_{ii}}{m_i^2} = \frac{\gamma_i}{m_i^2},$$

where $\gamma_i (\equiv 1 - \alpha_i \Sigma_{ii})$ is a measure of the degree to which the corresponding parameter $w_i$ is “determined” by the data.
Evidence Approximation

- Similarly, for $\beta$ we see that

$$\frac{\partial}{\partial \beta} \ln p(t|X,\alpha,\beta) = \frac{1}{2} \left( \frac{N}{\beta} - \|t - \Phi m\|^2 - \text{Tr}(\Sigma \Phi^T \Phi) \right). \quad (17)$$

- Using (8), we obtain

$$\text{Tr}(\Sigma \Phi^T \Phi) = \text{Tr} \left( \Sigma \Phi^T \Phi + \beta^{-1} \Sigma A - \beta^{-1} \Sigma A \right)$$

$$= \text{Tr} \left( \beta^{-1} \Sigma (\beta \Phi^T \Phi + A) - \beta^{-1} \Sigma A \right)$$

$$= \text{Tr} \left( \beta^{-1} (A + \beta \Phi^T \Phi)^{-1} (\beta \Phi^T \Phi + A) - \beta^{-1} \Sigma A \right)$$

$$= \text{Tr} \left( \beta^{-1} (I - \Sigma A) \right)$$

$$= \beta^{-1} \sum_{i=1}^{M} \gamma_i.$$
Setting (17) to zero and re-arranging, we obtain
\[
\| t - \Phi m \|^2 = \beta^{-1} N - \text{Tr}(\Sigma \Phi^T \Phi)
\]
\[
= \beta^{-1} \left( N - \sum_{i=1}^{M} \gamma_i \right).
\]
So we have
\[
\beta^{-1} = \frac{\| t - \Phi m \|^2}{N - \sum_{i=1}^{M} \gamma_i}.
\]
Evidence Approximation

- Summary of re-estimation equations:

\[
\alpha_{\text{new}}^i = \frac{\gamma_i}{m_i^2} \quad (18)
\]

\[
(\beta^{\text{new}})^{-1} = \frac{\|t - \Phi m\|^2}{N - \sum_{i=1}^{M} \gamma_i} \quad (19)
\]

- Iterative algorithm:
  - Choose initial values for \(\alpha\) and \(\beta\).
  - (A full Bayesian approach, which is not taken here, would go one level up to specify the hyperprior distributions \(p(\alpha)\) and \(p(\beta)\) over \(\alpha\) and \(\beta\).)
  - Use (7) and (8) to evaluate the mean and covariance of the posterior.
  - Use (18) and (19) to re-estimate \(\alpha\) and \(\beta\).
  - Repeat the procedure until a suitable convergence criterion is satisfied.

- Another approach: use the expectation maximization (EM) algorithm, but it is somewhat slower than the previous approach.
Predictive Distribution

- Having found the “optimal” hyperparameter values \( \alpha^* \) and \( \beta^* \), we make use of (1) and (6) to compute the predictive distribution over \( t \) for a new input \( x \):

\[
p(t|x, X, t, \alpha^*, \beta^*) = \int p(t|x, w, \beta^*)p(w|t, X, \alpha^*, \beta^*)dw
= \mathcal{N}(t|\mu(x), \sigma^2(x)).
\]

- By completing the square over \( w \) and then completing the square over \( t \), we can show that

\[
\mu(x) = m^T \phi(x)
\]
\[
\sigma^2(x) = (\beta^*)^{-1} + \phi(x)^T \Sigma \phi(x),
\]

where \( m \) and \( \Sigma \) are given by (7) and (8) in which \( \alpha \) and \( \beta \) are set to their optimized values \( \alpha^* \) and \( \beta^* \), and the predictive mean is given by (2) with \( w \) set to the posterior mean \( m \).
Sparsity

- Sparsity is achieved because in practice the posterior distributions of many of the weights in $w$ are sharply (infinitely) peaked around zero.

- The training data points associated with the remaining nonzero weights are called relevance vectors, in deference to the principle of automatic relevance determination (ARD) which inspired the work on RVM.

- The relevance vectors of an RVM are somewhat analogous to the support vectors of an SVM.

- For a wide range of regression and classification tasks, RVM is found to give models that are typically an order of magnitude more compact than the corresponding SVM, yet with little or no reduction in generalization error.
**Illustrative Example: RVM vs. SVM**

Figure: (a) RVM for regression. The mean of the predictive distribution for the RVM is shown by the red line and the one SD predictive distribution is shown by the shaded region. Also, the data points are shown in green and the 3 relevance vectors are indicated by blue circles. (b) $\nu$-SVM for regression. The predicted regression curve is shown by the red line and the $\epsilon$-insensitive tube corresponds to the shaded region. Also, the data points are shown in green and the 7 support vectors are indicated by blue circles.
Disadvantages of RVM

- Unlike SVM, training an RVM involves optimizing a nonconvex function.
- Training an RVM generally takes longer than training a comparable SVM (but there exists an alternative training procedure that can improve the training speed significantly).
In contrast to the regression model, we can no longer integrate analytically over $w$ to obtain the marginal likelihood function and the predictive distribution.

One approach is to use the **Laplace approximation** to approximate the integral:

- Initialize the hyperparameter vector $\alpha$.
- Find a Gaussian approximation to the posterior distribution over $w$.
- Find an approximation to the marginal likelihood.
- Maximize the approximate marginal likelihood to give a re-estimated value for $\alpha$.
- Repeat the process until convergence.
Generalized Logistic Regression

- We consider two-class classification tasks with a binary target variable $t \in \{0, 1\}$.
- The model represents the posterior probability of class $C_1$ as a linear combination of basis functions transformed by a logistic sigmoid function $\sigma(\cdot)$:
  $$ p(C_1|\mathbf{x}, \mathbf{w}) = y(\mathbf{x}, \mathbf{w}) = \sigma(\mathbf{w}^T \phi(\mathbf{x})), $$

where

$$ \sigma(a) = \frac{1}{1 + \exp(-a)} $$

and $p(C_0|\mathbf{x}, \mathbf{w}) = 1 - p(C_1|\mathbf{x}, \mathbf{w})$. This is the same as the logistic regression model for classification.

- We can write $p(t|\mathbf{x}, \mathbf{w})$ as:
  $$ p(t|\mathbf{x}, \mathbf{w}) = p(C_1|\mathbf{x}, \mathbf{w})^t p(C_0|\mathbf{x}, \mathbf{w})^{1-t}. $$

For notational simplicity, we will drop $\mathbf{x}$ from $p(t|\mathbf{x}, \mathbf{w})$ in the sequel.
Likelihood Function

- Data set: \( \{(\phi_n, t_n)\}_{n=1}^{N} \), where \( \phi_n = \phi(x_n) \) and \( t_n \in \{0, 1\} \).
- We let \( t = (t_1, \ldots, t_N)^T \).
- Likelihood function:

\[
p(t|w) = \prod_{n=1}^{N} p(t_n|w) = \prod_{n=1}^{N} p(C_1|x_n, w)^{t_n} p(C_0|x_n, w)^{1-t_n} = \prod_{n=1}^{N} y_n^{t_n} (1 - y_n)^{1-t_n},
\]

where \( y_n = p(C_1|x_n, w) \).
- Log likelihood function:

\[
\ln p(t|w) = \sum_{n=1}^{N} \{t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\}.
\]
Prior Distribution

- Gaussian prior (in the form of ARD prior) over $w$:

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

where each weight parameter $w_i$ has a separate precision hyperparameter $\alpha_i$ (like RVM for regression), which is different from logistic regression.
Posterior Distribution

- Posterior distribution over $\mathbf{w}$ for a fixed value of $\alpha$:

$$p(\mathbf{w}|\mathbf{t}, \alpha) = \frac{p(\mathbf{t}|\mathbf{w})p(\mathbf{w}|\alpha)}{p(\mathbf{t}|\alpha)}. $$

- Taking log on both sides, we get the log posterior distribution:

$$\ln p(\mathbf{w}|\mathbf{t}, \alpha) = \ln p(\mathbf{t}|\mathbf{w}) + \ln p(\mathbf{w}|\alpha) - \ln p(\mathbf{t}|\alpha)$$

$$= \sum_{n=1}^{N} \{ t_n \ln y_n + (1 - t_n) \ln(1 - y_n) \} - \frac{1}{2} \mathbf{w}^T \mathbf{A} \mathbf{w} + \text{const}, \quad (20)$$

where $\mathbf{A} = \text{diag}(\alpha_i)$.

- We maximize (20) to obtain the mode using iterative reweighted least squares (IRLS).
Suppose we want to minimize/maximize a function $E(w)$ that does not give a closed-form solution.

IRLS takes an iterative procedure based on this update formula:

$$w^{(\text{new})} = w^{(\text{old})} - H^{-1}\nabla E(w),$$

where $H = \nabla \nabla E(w)$ is the Hessian matrix whose elements comprise the second derivatives of $E(w)$ w.r.t. the components of $w$.

This scheme is based on the Newton-Raphson method.
Iterative Reweighted Least Squares

- Using the following property for the derivative of $\sigma(\cdot)$:

$$
\frac{d\sigma}{da} = \sigma(1 - \sigma),
$$

we can find the gradient vector and Hessian matrix of the log posterior distribution as:

$$
\nabla \ln p(w|t, \alpha) = \Phi^T(t - y) - Aw \tag{21}
$$

$$
\nabla\nabla \ln p(w|t, \alpha) = -(\Phi^T B \Phi + A) \tag{22}
$$

where $\Phi = (\phi(x_1), \ldots, \phi(x_N))^T$ is the design matrix, $y = (y_1, \ldots, y_N)^T$, and $B$ is an $N \times N$ diagonal matrix with elements $b_n = y_n(1 - y_n)$. 
Laplace Approximation

- The **mode** $\mathbf{w}^*$ of the resulting approximation to the posterior distribution, which corresponds to the **mean** of the Gaussian approximation, is obtained by setting (21) to zero:

\[
\mathbf{w}^* = \mathbf{A}^{-1} \mathbf{\Phi}^T (\mathbf{t} - \mathbf{y}).
\]

- The **inverse covariance matrix** $\mathbf{\Sigma}^{-1}$ of the Gaussian approximation to the posterior distribution is equal to the **negative Hessian** at $\mathbf{w}^*$:

\[
\mathbf{\Sigma} = (\mathbf{\Phi}^T \mathbf{B} \mathbf{\Phi} + \mathbf{A})^{-1}.
\]
Marginal Likelihood

- The Laplace approximation is then used to approximate the marginal likelihood $p(t|\alpha)$.
- As discussed above, we approximate $p(w|t, \alpha)$ by the Gaussian approximation at the mode of $p(w|t, \alpha)$, i.e.
  \[ p(w|t, \alpha) \approx N(w|w^*, \Sigma). \]
- From this Gaussian approximation, we obtain
  \[
  p(t, w|\alpha) = p(w|t, \alpha)p(t|\alpha) \\
  \approx N(w|w^*, \Sigma)p(t|\alpha) \\
  = \frac{1}{(2\pi)^{M/2}|\Sigma|^{1/2}} \exp \left\{ -\frac{1}{2} (w - w^*)\Sigma^{-1}(w - w^*) \right\} p(t|\alpha)
  \]
- Setting $w$ to $w^*$ in $p(t, w|\alpha)$, we get the marginal likelihood:
  \[
  p(t|\alpha) \approx p(t, w^*|\alpha) (2\pi)^{M/2}|\Sigma|^{1/2} = p(t|w^*)p(w^*|\alpha)(2\pi)^{M/2}|\Sigma|^{1/2}. \tag{23}
  \]
Evidence Approximation

- We now substitute for $p(t|w^*)$ and $p(w^*|\alpha)$ in (23) and set the derivative of $\ln p(t|\alpha)$ w.r.t. $\alpha_i$ to zero to obtain

$$-rac{1}{2}(w_i^*)^2 + \frac{1}{2\alpha_i} - \frac{1}{2}\Sigma_{ii} = 0.$$ 

- Defining $\gamma_i = 1 - \alpha_i\Sigma_{ii}$ and re-arranging gives

$$\alpha_i^{\text{new}} = \frac{\gamma_i}{(w_i^*)^2},$$

which is identical to the re-estimation formula (18) for the regression SVM.
Evidence Approximation

- If we define
  \[ \hat{t} = \Phi w^* + B^{-1}(t - y), \]
  we can write the approximate log marginal likelihood in the form
  \[ \ln p(t|\alpha) = -\frac{1}{2} \left\{ N \ln(2\pi) + \ln |C| + (\hat{t})^T C^{-1} \hat{t} \right\}, \]
  where
  \[ C = B + \Phi A \Phi^T. \]

- This takes the same form as (10) in the regression case, and so we can apply similar extensions to speed up the learning algorithm.
Multi-Class Classification

- For \( K > 2 \) classes, we define \( K \) linear models of the form
  \[
  a_k = \mathbf{w}_k^T \phi(\mathbf{x}).
  \]

- The \( K \) linear models are combined using a softmax function to give outputs:
  \[
  y_k(\mathbf{x}) = \frac{\exp(a_k)}{\sum_{j=1}^{K} \exp(a_j)}.
  \]

- Likelihood function:
  \[
  p(\mathbf{T} | \mathbf{w}_1, \ldots, \mathbf{w}_K) = \prod_{n=1}^{N} \prod_{k=1}^{K} y_{nk}^{t_{nk}},
  \]
  where the target values \( t_{nk} \) have a 1-of-\( K \) coding for each data point \( n \), and \( \mathbf{T} \) is a matrix with elements \( t_{nk} \).
RVM vs. SVM for Classification

- Unlike support vectors in SVM, relevance vectors tend not to lie in the region of the decision boundary.
- RVM tends to give a much sparser model than SVM.
- Unlike SVM, the RVM output can be used directly to make probabilistic predictions.
- RVM gives a more principled approach to multi-class classification than the pairwise method used in SVM, but its disadvantage is the relatively long training time compared with SVM (since the Hessian matrix has size $MK \times MK$).
Illustrative Example

Figure: $\nu$-SVM applied to a nonseparable 2-dimensional data set. The support vectors are indicated by circles.

Figure: (a) RVM applied to the same data set, with the relevance vectors indicated by circles. (b) Posterior probability given by the RVM output in which the proportion of red (blue) ink indicates the probability of that point belonging to the red (blue) class.