Spares Kernel Machines

A self-study metarials for PRML [1]

Jisung Lim[1]

[1] B.S. Candidate of Industrial Engineering
Yonsei University, South Korea.

8th February, 2017


Christopher M. Bishop. **Pattern Recognition and Machine Learning**. Springer, 2006.

Summary

Kernel Method (Review)

Dual Representation

1. Many linear models can be reformulated in terms of a **dual representation.**
2. Kernel function $k(\boldsymbol{x}, \boldsymbol{x}')$ arises naturally. (at the same time hide basis term).

$$\begin{aligned} y(\boldsymbol{x}_{\text{new}}) &= \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_{\text{new}}) = \boldsymbol{\phi}(\boldsymbol{x}_{\text{new}})^T (\lambda \mathbf{I} + \boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T \mathbf{t} \\ &= \boldsymbol{a}^T \boldsymbol{\Phi} \boldsymbol{\phi}(\boldsymbol{x}_{\text{new}}) = \boldsymbol{k}(\boldsymbol{x}_{\text{new}})^T (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{t} \end{aligned} \tag{1}$$

Constructing Kernel

1. From basis function (or feature space)
   (ex) $k(\boldsymbol{x}_n, \boldsymbol{x}_m) = \boldsymbol{\phi}(\boldsymbol{x}_n)\boldsymbol{\phi}(\boldsymbol{x}_m)/\alpha$
2. Directly (Should be followed by validation step)
3. Build them out of simpler one

Gaussian Process

1. Not $\boldsymbol{w}$, now $\boldsymbol{y}_n$.
2. Kernel function arise in covariance matrix $\mathbf{K}$
3. Pros: Can consider infinite number of basis function.
4. Cons: Computationally expensive, it requires computations $O(N^3)$ (instead of $O(M^3)$), for each new input.

Sparse Kernel Machines

We will look at kernel-based algorithms that have ***sparse solutions***.

Sparse Kernel Machine

- Predictions for new inputs depend only on the kernel function evaluated at a **"subset of the training data points."** (Less expensive)
- We shall look at SVM (Supported Vector Machines) and RVM (Relevance Vector Machines).

Support Vector Machine

- Sparse Kernel Machine with determinstc approach. (Non-probabilistic)
- Determine model parameters solving convex optimization problem. (Local opt. = Global opt.)
- Popular for solving problems in classification, regression and novelty detection.

Relevance Vector Machine

- Based on a Bayesian formulation
- Provides posterior probabilistic outputs
- Typically much sparser solution than the SVM.

Convex optimization and KKT conditions (1)

Primal Problem

$$\begin{gathered} \boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} f(\boldsymbol{x}) \\ \text{where} \\ \boldsymbol{h}(\boldsymbol{x}) = \mathbf{0} \\ \boldsymbol{g}(\boldsymbol{x}) \leq \mathbf{0} \\ \boldsymbol{x} \geq \mathbf{0} \end{gathered} \tag{2}$$

Dual problem

$$\begin{gathered} (\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = \arg\max_{\boldsymbol{\lambda},\boldsymbol{\mu}} \tilde{L}(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \arg\max_{\boldsymbol{\lambda},\boldsymbol{\mu}} \inf_{\boldsymbol{x}} L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \\ \text{where} \\ \boldsymbol{\lambda} \geq \mathbf{0} \end{gathered} \tag{3}$$

Convex optimization and KKT conditions (2)

KKT multipliers

$$L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\boldsymbol{x}) + \boldsymbol{\lambda}^T \boldsymbol{g}(\boldsymbol{x}) + \boldsymbol{\mu}^T \boldsymbol{h}(\boldsymbol{x}) \tag{4}$$

KKT conditions

$$\begin{aligned} \text{Stationarity:} \quad & \nabla_{\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{\mu}} L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \mathbf{0} \\ \text{primal feasible:} \quad & \boldsymbol{g}(\boldsymbol{x}) \leq \mathbf{0} \\ & \boldsymbol{h}(\boldsymbol{x}) = \mathbf{0} \\ \text{dual feasible:} \quad & \boldsymbol{\lambda} \geq \mathbf{0} \\ \text{complementary slackness:} \quad & \boldsymbol{\lambda}^T \boldsymbol{g}(\boldsymbol{x}) = \mathbf{0} \end{aligned} \tag{5}$$

Model Setting

- Let's begin with two-class classification problem using linear models of the form

$$y(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}) + b$$
$$\text{where}$$
$$\boldsymbol{\phi}(\boldsymbol{x}): \text{ finite, fixed feature space} \tag{6}$$
$$b: \text{ bias parameter}$$

- The training data set comprises N observations which is given by

$$\{(\boldsymbol{x}_1, t_1), \ldots, (\boldsymbol{x}_N, t_N)\} \quad \text{where} \quad t_n \in \{-1, 1\} \tag{7}$$

- We assume, for the moment, that theg training data set is ***linearly separable*** in feature space.
- In linearly separable setting, we can deterimine $y(\boldsymbol{x})$ exactly satisfying

$$y(\boldsymbol{x}) \begin{cases} > 0 & \text{if} \quad t_n = 1 \\ < 0 & \text{if} \quad t_n = -1 \end{cases}, \tag{8}$$

so that $|y(\boldsymbol{x})| = t_n y(\boldsymbol{x})$.

Maximum Margin Classifiers (1)

The support vector machine approaches this problem through the concept of the ***margin***, which is defined to be the smallest distance between the decision boundary and any of the samples.

- ***Perpendicular distance*** between a point $\boldsymbol{x}$ and a hyperplane $y(\boldsymbol{x}) = 0$ is given by

$$\frac{|y(\boldsymbol{x})|}{||\boldsymbol{w}||} = \frac{t_n(\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}) + b)}{||\boldsymbol{w}||}, \tag{9}$$

  since $y(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}) + b$ and $|y(\boldsymbol{x})| = t_n y(\boldsymbol{x})$.

- ***Margin*** defined to be the smallest perpendicular distance is given by

$$\min_n \left\{ \frac{t_n(\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}) + b)}{||\boldsymbol{w}||} \right\} \tag{10}$$

- We wish to optimize the parameters $\boldsymbol{w}$ and b in order to ***maximize the margin***.

$$\arg\max_{\boldsymbol{w},b} \left\{ \frac{1}{||\boldsymbol{w}||} \min_n \left[t_n(\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}) + b) \right] \right\} \tag{11}$$

  Solving this problem directly would be vary challenging, so we will use some technique in next slide.

Maximum Margin Classifiers (2)

- Rescale $\boldsymbol{w}$ and b by same ratio ($t_n y(\boldsymbol{x})/||\boldsymbol{w}||$ unchanged) to fix the value of the numerator of the margin

$$t_n(\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}) + b) = 1, \tag{12}$$

  so that all data points will satisfies ***the constraints***

$$t_n(\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}) + b) \geq 1 \quad \forall n. \tag{13}$$

- Hence we get more simplified optimization problem, given by

$$\arg\max_{\boldsymbol{w},b} \frac{1}{||\boldsymbol{w}||} = \arg\min_{\boldsymbol{w},b} \frac{1}{2}||\boldsymbol{w}||^2 \tag{14}$$

- Since this problem is convex optimization, we will use KKT conditions and solve dual problem. First, the KKT multiplier is given by

$$L(\boldsymbol{w}, b, \boldsymbol{a}) = \frac{1}{2}||\boldsymbol{w}||^2 - \boldsymbol{a}^T(\mathbf{y} - \mathbf{1}_N) \tag{15}$$

Maximum Margin Classifiers (3); a

- Following equations are satisfied by stationarity of KKT conditions

$$\boldsymbol{w} = \sum_{n=1}^{N} a_n t_n \boldsymbol{\phi}(\boldsymbol{x}_n) \quad \text{and} \quad 0 = \sum_{n=1}^{N} a_n t_n, \tag{16}$$

  so that we get dual problem, given by

$$\arg\max_{\boldsymbol{a}} \left[\tilde{L}(\boldsymbol{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m k(\boldsymbol{x}_n, \boldsymbol{x}_m) \right] \tag{17}$$

$$\text{where}$$

$$\boldsymbol{a} \geq \mathbf{0} \quad \text{and} \quad \boldsymbol{a}^T \mathbf{t} = 0$$

- For a fixed set of basis functions whose number $M < N$, dual problem appears much more expensive. However, it allows the model to be ***reformulated using kernels***, and so the maximum margin classifier can be applied efficiently to feature spaces whose dimensionality exceeds the number of data points, including ***infinite feature space***.

Maximum Margin Classifiers (4); support vectors

- In order to classify new data points using the trained model, we evaluate a prediction function based on kernel which is given by

$$y(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}) + b = \sum_{i=1}^{N} a_n t_n k(\boldsymbol{x}, \boldsymbol{x}_n) + b \tag{18}$$

  since $\boldsymbol{w} = \sum_{n=1}^{N} a_n t_n \boldsymbol{\phi}(\boldsymbol{x}_n)$.
- For the equation (18), we can find $\boldsymbol{a}$ solving the dual problem. Before we evalute b, we will use KKT condtions to make the model sparser. The KKT conditions is given by

$$\begin{aligned} a_n &\geq 0 \quad \forall n \\ t_n y(\boldsymbol{x}_n) - 1 &\geq 0 \quad \forall n \\ a_n(t_n y(\boldsymbol{x}_n) - 1) &= 0 \quad \forall n, \end{aligned} \tag{19}$$

  i.e., for every sample point, $a_n = 0$ or $t_n y(\boldsymbol{x}_n) - 1 = 0$.

Maximum Margin Classifiers (4); support vectors

- In equation (18), the point $(\boldsymbol{x}_n, t_n)$ plays no role in making prediction if $a_n = 0$. We call the remaining data points ***support vectors***, which satisfies $t_n y(\boldsymbol{x}_n) = 1$, so that lie on the maximum margin hyperplanes in feature space.
- The origin of sparsity in the SVM is that the maximum margin hyperplain is can be defined only by the location of the support vectors.

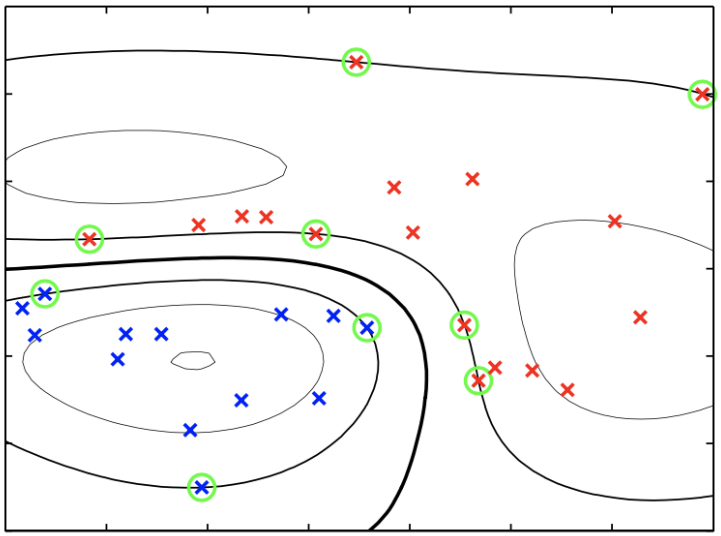

Figure 1: Contours of constant $y(x)$ obtained from a support vector machine having a Gaussian kernel function.

Maximum Margin Classifiers (5); b

We've found $\boldsymbol{a}$. Now, we can then determine the value of the threshold parameter b.

- From $t_n y(\boldsymbol{x}_n) = 1$

$$t_n \left(\sum_{m \in \mathcal{S}} a_m t_m k(\boldsymbol{x}_n, \boldsymbol{x}_m) + b \right) = 1 \tag{20}$$

  Where $\mathcal{S}$ denote the set of indices of the support vectors.
- Multiply through by t_n, make use of $t_n^2 = 1$, and then average over every support vector for numerical stability.

$$b = \frac{1}{N_{\mathcal{S}}} \sum_{n \in \mathcal{S}} \left(t_n - \sum_{m \in \mathcal{S}} a_m t_m k(\boldsymbol{x}_n, \boldsymbol{x}_m) \right) \tag{21}$$

  where $N_{\mathcal{S}}$ is the cardinality of $\mathcal{S}$.
- We can express the maximum margin classifier in terms of the minimization of an error function, with a simple quadratic regularizer

$$\sum_{n=1}^{N} E_\infty (y(\boldsymbol{x}_n) t_n - 1) + \lambda ||\boldsymbol{w}||^2 \tag{22}$$

where

$$E_\infty(z) = 0 \quad \text{for} \quad z \geq 0 \quad \text{or} \quad \infty \quad \text{otherwise}$$

Overlapping class distributions

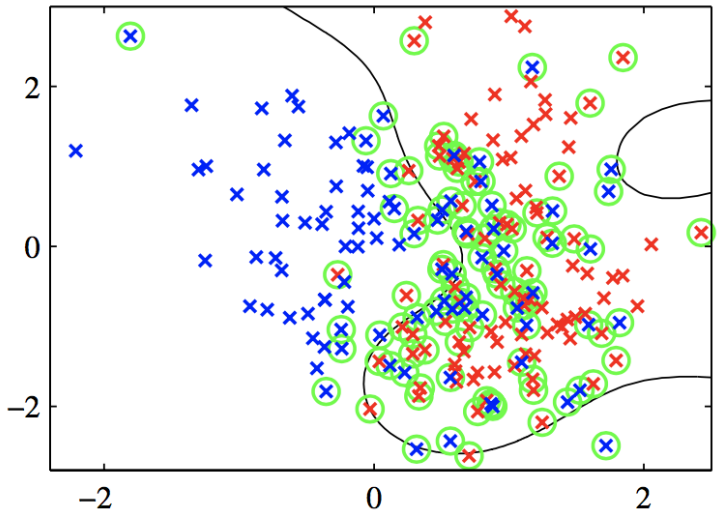

Figure 2: Illustration of the ν-SVM applied to a nonseparable data set in two dimensions.

SVM for regression

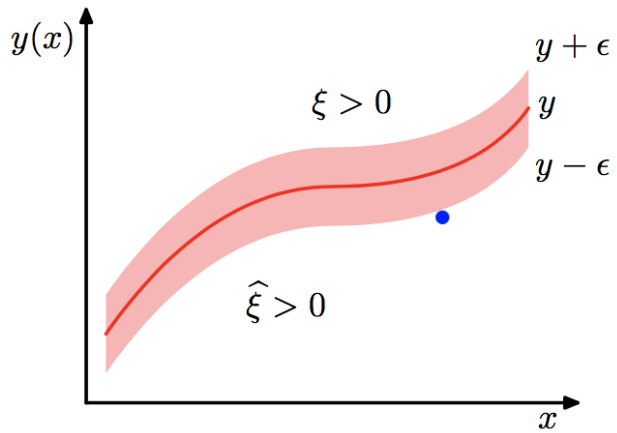

Figure 3: Illustration of SVM regression.

The limitations of SVM

Even though SVM have been used in a variety of classification and regression applications, they suffer from a number of limitations, which are given by

- ***Decision machine***; Decisions rather than posterior probabilities.
- originally formulated for two classes, problemetic to the extension to $K > 2$ classes.
- There is a ***complexity parameter*** C, ν, or even ϵ (in the case of regression), that must be found using a hold-out method such as cross-validation.
- Predictions are expressed as linear combinations of kernel functions that are ***centred on*** training data points and that are required to be ***positive definite***.

The ***relevance vector machine*** or RVM is

- A Bayesian sparse kernel technique for regression and classification.
- It shares many of the characteristics of the SVM.
- Whilst, it avoids SVM's principal limitations.
- Additionally, it typically leads to much sparser models whilst maintaining comparable generalization error.

Model Setting

The model is defined by the conditional distribution for a real-valued target variable t, given an input vector x, which takes the form

$$p(t|\boldsymbol{x}, \boldsymbol{w}, \beta) = \mathcal{N}(t|y(\boldsymbol{x}), \beta^{-1})$$
$$\text{where}$$
$$y(\boldsymbol{x}) = \sum_{i=1}^{M} w_i \phi_i(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}) \tag{23}$$

Particularly, The RVM is the specific instance of this model where the basis functions are given by kernels, with one kernel associated with each of the data points from the training set, which taks the form

$$y(\boldsymbol{x}) = \sum_{i=1}^{N} w_n k(\boldsymbol{x}, \boldsymbol{x}_n) + b \tag{24}$$

- b is a bias parameter, hence the number of parameters in this case is $M = N + 1$.
- $y(\boldsymbol{x})$ is similar with SVM's the predictive model, except directly using w_i, not a_i.
- Subsequent analysis is valid no matter what basis functions are chosen. For generality, we shall work with the form given in (23).
- No restriction: positive-definite kernel for solving optimization problem, choice of basis function restricted by the number or locations of training data points.

Inference stage

Consider the form of input as data matrix $\mathbf{X}$ and correspoinding target values $\mathbf{t}$, then the likelihood function is given by

$$p(\mathbf{t}|\mathbf{X}, \boldsymbol{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x_n}), \beta^{-1}). \tag{25}$$

Next, we introduce a prior distribution over the parameter vector $\boldsymbol{w}$. The key difference in the RVM is that we introduce a separate hyperparameter α_i for each of the weight parameter w_i, which takes the form

$$p(\boldsymbol{w}|\boldsymbol{\alpha}) = \prod_{i=1}^{N} \mathcal{N}(w_i|0, {\alpha_i}^{-1}) \tag{26}$$

Therefore, the posterior can be evaluated as follows

$$\begin{gathered} p(\boldsymbol{w}|\mathbf{t}, \mathbf{X}, \boldsymbol{\alpha}, \beta) = \mathcal{N}(\boldsymbol{w}|\mathbf{m}, \boldsymbol{\Sigma}) \\ \text{where} \\ \mathbf{m} = \beta\boldsymbol{\Sigma}\boldsymbol{\Phi}^T\mathbf{t} \\ \boldsymbol{\Sigma} = (\mathbf{A} + \beta\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1} \end{gathered} \tag{27}$$

where $\boldsymbol{\Phi} = \phi_i(\boldsymbol{x}_n)$ and $\mathbf{A} = \text{diag}(\alpha_i)$.

Evidence approximation

The value of $\boldsymbol{\alpha}$ and β are determined by ***evidence approximation***,

$$p(t|\mathbf{t}) \simeq p(t|\mathbf{t}, \hat{\boldsymbol{\alpha}}, \hat{\beta}) = \int p(t|\boldsymbol{w}, \hat{\beta}) p(\boldsymbol{w}|\mathbf{t}, \hat{\boldsymbol{\alpha}}, \hat{\beta}) \mathrm{d}\boldsymbol{w} \tag{28}$$

and then, evaluating the marginal likelihood function, we get

$$\ln p(t|\mathbf{t}) \simeq p(t|\mathbf{t}, \hat{\boldsymbol{\alpha}}, \hat{\beta}) = \ln \mathcal{N}(\mathbf{t}|\mathbf{0}, \mathbf{C}) = -\frac{1}{2}\left\{N \ln(2\pi) + \ln|\mathbf{C}| + \mathbf{t}^T \mathbf{C}^{-1} \mathbf{t}\right\} \tag{29}$$

where $\mathbf{C} = \beta^{-1}\mathbf{I} + \boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^T$. Maximizing the equation with respect to $\boldsymbol{\alpha}$ and β, we obtain the re-estimation equations

$$\alpha_i^{(\text{new})} = \frac{\gamma_i}{m_i^2} \tag{30}$$

$$\beta^{(\text{new})\,-1} = \frac{||\mathbf{t} - \boldsymbol{\Phi}\mathbf{m}||^2}{N - \sum_i \gamma_i} \tag{31}$$

where $\gamma_i = 1 - \alpha_i \Sigma_{ii}$. Therefore, we shall follow this process of

1. Choosing initial value for $\boldsymbol{\alpha}$ and β.
2. Evaluating the mean $\mathbf{m}$ and covariance $\boldsymbol{\Sigma}$ of the psoterior.
3. Re-estimating the hyperparameters $\boldsymbol{\alpha}_{\text{new}}$ and β_{new}.
4. Repeating re-estimation until a suitable convergence criterion is statified.

Predictive distribution

Having found values $\boldsymbol{\alpha}^*$ and β^* satisfying specific convergence criteria, we can evaluate the predictive distribution over t for a new input $\boldsymbol{x}$ as follows

$$\begin{aligned} p(t_{\text{new}}|\boldsymbol{x}_{\text{new}}, \mathbf{X}, \mathbf{t}, \boldsymbol{\alpha}^*, \beta^*) &= \int p(t|\boldsymbol{x}, \boldsymbol{w}, \beta^*) p(\boldsymbol{w}|\mathbf{x}, \mathbf{t}, \boldsymbol{\alpha}^*, \beta^*) \mathrm{d}\boldsymbol{w} \\ &= \mathcal{N}(t|\mathbf{m}^T \boldsymbol{\phi}(\boldsymbol{x}), \sigma^2(\boldsymbol{x})) \end{aligned} \tag{32}$$

where

$$\sigma^2(\boldsymbol{x}) = \frac{1}{(\beta^*)} + \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{\Sigma} \boldsymbol{\phi}(\boldsymbol{x}) \tag{33}$$

In the case of an RVM with the basis functions centred on data points, the model will therefore become increasingly certain of its predictions when extrapolating outside the domain of the data, which of course is undesirable.

Sparsity and Relevance Vectors

The Origin of Sparsity

When we maximize the evendeince function with respect to the hpyerparameters, a significant proportion of α_i go to infinity, and the corresponding weight parameters w_i have posterior distributions that are concentrated at zero, $\mathcal{N}(w_i|0,0)$. The basis functions associated with these parameters therefore play no role in the predictions made by the model and so are effectively pruned out, reslting in a spare model.

Relevance Vector

The hyperparameters $\{\alpha_i\}$ driven to large values makes its associated weight parameters w_i have posterior distributions $\mathcal{N}(w_i|0,0)$. Since the parameter w_i is determined to be zero stritly, corresponding basis function $\phi_i(\boldsymbol{x})$ has no role in making prediction for new inputs. In this casel, the input $\boldsymbol{x}_n$ corresponding to the remaining nonzero weights are called ***relevance vectors***.

Pros and Cons

We see that the number of relevance vectors in the RVM is significantly smaller than the number of support vectors used by the SVM, and remarkably, this greater sparsity is achieved with little or no reduction in generalization error compared with the corresponding SVM. The principal disadvantage of the RVM compared to the SVM is that training involves optimizing a nonconvex function, and training times can be longer than for a comparable SVM.

Informal insight into the origin of sparsity

To gett informal insight into the origin of sparsity, we are getting started with simpler model specified as follows

- 2 observations t_1 and t_2
- a signle basis $\phi(\boldsymbol{x})$ with hyperparameter α
- isotropic noise having precision β
- The marginal likelihood is given by $p(\mathbf{t}|\alpha, \beta) = \mathcal{N}(\mathbf{0}, \mathbf{C})$

Then the covariance matrix of marginal likelihood function $\mathbf{C}$ is given by

$$\mathbf{C} = \frac{1}{\beta}\mathbf{I} + \frac{1}{\alpha}\boldsymbol{\varphi}\boldsymbol{\varphi}^T \tag{34}$$

If there is poor alignment betwen the direction of $\boldsymbol{\varphi}$ and that of the training data vector $\mathbf{t}$, then the crresponding hyperparmeter α will be driven to ∞, and the basis vector will be pruned out from the model.

General case of M basis vectors (1)

For the more general case of M basis vectors $\boldsymbol{\varphi}_m$ a similar intuition holds, namely that if a particular basis vector is poorly aligned with the data vector $\boldsymbol{t}$, then it is likely to be pruned from the model. We now investigate this mechanism for sparsity from a more mathematical perspective.

At first, the re-estimation of the parameter α_i is given by

$$\alpha_i^{(\text{new})} = \frac{\gamma_i}{m_i^2} \quad \text{where} \quad \gamma_i = 1 - \alpha_i \Sigma_{ii} \quad \text{and} \quad \boldsymbol{\Sigma} = \left(\mathbf{A} + \beta \boldsymbol{\Phi}^T \boldsymbol{\Phi}\right)^{-1} \tag{35}$$

These results therefore represent implicit solutions, and iteration would be required even to determine a single α_i with all other fixed α_j for $j \neq i$.

General case of M basis vectors (2)

First, we pull out the contribution from α_i in the matrix $\mathbf{C}$ to give

$$\mathbf{C} = \beta^{-1}\mathbf{I} + \sum_{j \neq i} \alpha_j^{-1} \boldsymbol{\varphi}_j \boldsymbol{\varphi}_j^T + \alpha_i^{-1} \boldsymbol{\varphi}_i \boldsymbol{\varphi}_i^T = C_{-i} + \alpha_i^{-1} \boldsymbol{\varphi}_i \boldsymbol{\varphi}_i^T \tag{36}$$

Using the matrix identity, the determinant and inverse of $\mathbf{C}$ can then be written

$$\begin{aligned} |\mathbf{C}| &= |\mathbf{C}_{-i}|(1 + \alpha_i^{-1} \boldsymbol{\varphi}_i^T \mathbf{C}_{-i}^{-1} \boldsymbol{\varphi}_i) \\ \mathbf{C}^{-1} &= \mathbf{C}_{-i}^{-1} - \frac{\mathbf{C}_{-i}^{-1} \boldsymbol{\varphi}_i \boldsymbol{\varphi}_i^T \mathbf{C}_{-i}^{-1}}{\alpha_i \boldsymbol{\varphi}_i^T \mathbf{C}_{-i}^{-1} \boldsymbol{\varphi}_i} \end{aligned} \tag{37}$$

Using these results, we can separate log marginal likelihood function

$$\begin{gathered} L(\boldsymbol{\alpha}) = L(\boldsymbol{\alpha}_{-i}) + \lambda(\alpha_i) \\ \text{where} \\ \lambda(\alpha_i) = \frac{1}{2}\left[\ln \alpha_i - \ln(\alpha_i + s_i) + \frac{q_i^2}{\alpha_i + s_i}\right] \end{gathered} \tag{38}$$

where the two qunatities: *sparsity* $s_i = \boldsymbol{\varphi}_i^T \mathbf{C}_{-i}^{-1} \boldsymbol{\varphi}_i$ and *quality* $q_i = \boldsymbol{\varphi}_i^T \mathbf{C}_{-i}^{-1} \mathbf{t}$

Sparsity and Quality

- The sparsity s_i measures the extent to which basis function $\boldsymbol{\varphi}_i$ overlaps with the other basis vectors in the mode
- The quality q_i represents a measure of the alignment of the basis vector $\boldsymbol{\varphi}_i$ with the error between the training set values $\mathbf{t}$ and the vector $\mathbf{y}_{-i}$ of predictions that would result from the model with the vector $\boldsymbol{\varphi}_i$ excluded.

The stationary points of the marginal likelihood with respect to α_i occur when the derivative

$$\frac{\mathrm{d}}{\mathrm{d}\alpha_i}\lambda(\alpha_i) = \frac{\alpha_i^{-1}s_i^2 - (q_i^2 - s_i)}{2(\alpha_i + s_i)^2} \tag{39}$$

is equal to zero. There are two possilble forms for the solution

$$\begin{aligned} q_i^2 < s_i; \quad & \alpha_i \to \infty \\ q_i^2 > s_i; \quad & \alpha_i = \frac{s_i^2}{q_i^2 - s_i} \end{aligned} \tag{40}$$

We see that the relative size of the quality and sparsity terms determines whether a particular basis vector will be pruned from the model or not.
This approach

- has yielded closed-form solution for α_i
- probides insight into the origin of sparsity in the RVM
- leads to a practical algorithm for optimizing the hyperparameters that has significant spped advantages.

RVM for classification

To start with, we consider two-class problems with a binary target variable $t \in \{0, 1\}$. The model is given by

$$y(\boldsymbol{x}, \boldsymbol{w}) = \sigma(\boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x})) \tag{41}$$

where $\sigma(\cdot)$ is logistic sigmoid function. We introduce Gaussian prior over the weight vector $\boldsymbol{w}$, in which there is a separate precision hyperparameter associated with each weight parameter.
Here we use the Laplace approximation, which was applied to the closely related problem of Bayesian logistic regression. The description of the process is given by

1. Begin with choosing initial hyperparameter $\boldsymbol{\alpha}$
2. Given $\boldsymbol{\alpha}$, build Gaussian approximation to the posterior distribution and thereby an approximation to the marginal likelihood
3. Optimize the marginal likelihood to evaluate re-estimation value of $\boldsymbol{\alpha}$
4. repeat until convergence

RVM for classification

The results are as follows

re-estimation formula

$$\alpha_i^{\text{new}} = \frac{\gamma_i}{(w_i^*)^2} \tag{42}$$

Log marginal likelihood

$$\ln p(\mathbf{t}|\boldsymbol{\alpha}) = -\frac{1}{2}\left\{N\ln(2\pi) + \ln|\mathbf{C}| + \hat{\mathbf{t}}^T\mathbf{C}^{-1}\hat{\mathbf{t}}\right\} \tag{43}$$

where

$$\mathbf{C} = \mathbf{B} + \mathbf{\Phi}\mathbf{A}\mathbf{\Phi}^T$$

This takes the same form in the regression case, and so we can apply the same analysis of sparsity and obtain the same fast learning algorithm in which we fully optimize a single hyperparameter α_i at each step.

RVM for classification

We see that the relevance vectors tend not to lie in the region of the decision boundary, in contrast to the support vector machine. This is consistent with our earlier discussion of sparsity in the RVM, because a basis function $\phi_i(\boldsymbol{x})$ centred on a data point near the boundary will have a vector $\boldsymbol{\varphi}_i$ that is poorly aligned with the training data vector $\mathbf{t}$.

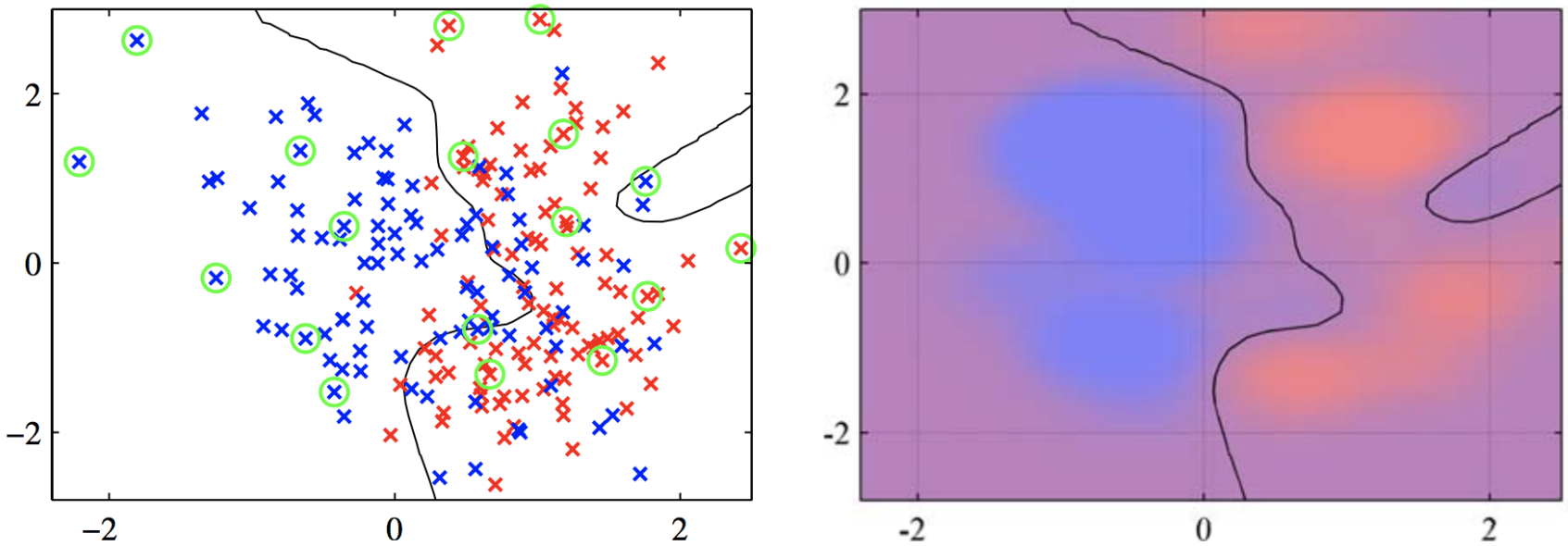

Figure 4: Illustration of RVM classification.

For $K > 2$ classes

K linear models of the form

$$a_k = \boldsymbol{w}_k^T \boldsymbol{x} \tag{44}$$

which are conbined using a softmax function to give outputs

$$y_k(\boldsymbol{x}) = \frac{\exp(a_k)}{\sum_j \exp(a_j)} \tag{45}$$

The log likelihood function is then given by

$$\ln p(\mathbf{T}|\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K) = \prod_{n=1}^{N} \prod_{k=1}^{K} y_{nk}^{t_n k} \tag{46}$$

where $t_n k$ is 1-of-K coding for each data point n.