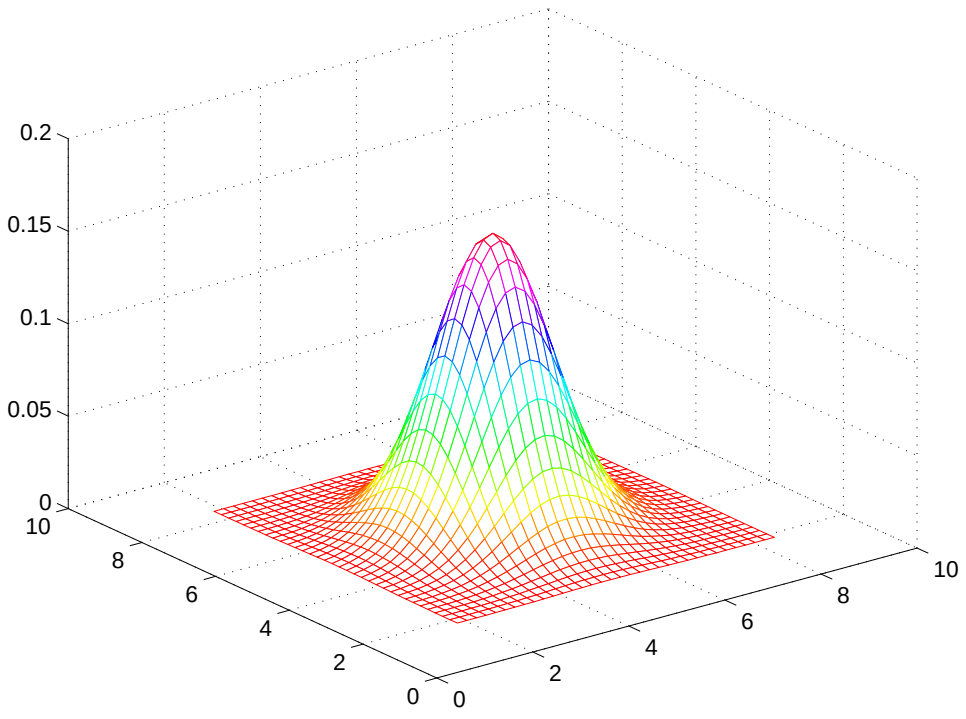
University of Cambridge
Engineering Part IIB

Paper 4F10: Statistical Pattern Processing

Handout 9: Relevance Vector Machines



Bill Byrne
bill.byrne@eng.cam.ac.uk
Michaelmas 2012

Outline

The previous lectures have examined the use of Gaussian processes and basis function interpolation for prediction and classification. This was related to the support vector machines as both make use of kernel-like operations to yield non-linear classifiers.

These (the mean for Gaussian processes/basis interpolation) could be expressed in the form

$$\mu_y = \sum_{i=1}^{n} \alpha_i k(\boldsymbol{x}, \boldsymbol{x}_i)$$

One difference between these approaches and the SVM is that:

- SVMs: Lagrange multipliers $\boldsymbol{\alpha}$ are sparse
- GPs: $\boldsymbol{\alpha}$ non-sparse
- basis interpolation: in the general case non-sparse

Though SVMs are sparse there are some limitations:

- SVMs are a discriminative function: sometimes useful to have a discriminative model as this gives the class posteriors;
- 'C' must be set: determines the error/margin play-off;

Would like to have some of the attributes of GPs/basis regression along with the sparseness of SVMs.

Basis Function Bayesian Regression

The underlying equations for basis function regression are

- input-target relation

$$y = \boldsymbol{\phi}(\boldsymbol{x})'\boldsymbol{w} + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_{\mathtt{n}}^2)$$

- weight distribution is given by

$$p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}; \mathbf{0}, \sigma_{\mathtt{w}}^2\mathbf{I})$$

- distribution of the training "targets"

$$p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w}) = \mathcal{N}(\boldsymbol{y}; \boldsymbol{\Phi}\boldsymbol{w}, \sigma_{\mathtt{n}}^2\mathbf{I})$$

where

$$\boldsymbol{\Phi} = \begin{bmatrix} \phi(||\boldsymbol{x}_1 - \boldsymbol{x}_1||) & \dots & \phi(||\boldsymbol{x}_1 - \boldsymbol{x}_n||) \\ \vdots & \ddots & \vdots \\ \phi(||\boldsymbol{x}_n - \boldsymbol{x}_1||) & \dots & \phi(||\boldsymbol{x}_n - \boldsymbol{x}_n||) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\phi}(\boldsymbol{x}_1)' \\ \vdots \\ \boldsymbol{\phi}(\boldsymbol{x}_n)' \end{bmatrix}$$

Prediction is then based on

$$p(y|\boldsymbol{x}, \boldsymbol{X}, \boldsymbol{y}) = \mathcal{N}(y; \boldsymbol{\phi}(\boldsymbol{x})'\boldsymbol{\mu}_w, \boldsymbol{\phi}(\boldsymbol{x})'\boldsymbol{\Sigma}_w\boldsymbol{\phi}(\boldsymbol{x}) + \sigma_{\mathtt{n}}^2)$$

where

$$p(\boldsymbol{w}|\boldsymbol{y}, \boldsymbol{X}) = \mathcal{N}(\boldsymbol{w}; \boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w)$$

and

$$\begin{aligned} \boldsymbol{\mu}_w &= \frac{1}{\sigma_{\mathtt{n}}^2}\left(\frac{1}{\sigma_{\mathtt{n}}^2}\boldsymbol{\Phi}'\boldsymbol{\Phi} + \frac{1}{\sigma_{\mathtt{w}}^2}\mathbf{I}\right)^{-1}\boldsymbol{\Phi}'\boldsymbol{y} = \frac{1}{\sigma_{\mathtt{n}}^2}\boldsymbol{\Sigma}_w\boldsymbol{\Phi}'\boldsymbol{y} \\ \boldsymbol{\Sigma}_w &= \left(\frac{1}{\sigma_{\mathtt{n}}^2}\boldsymbol{\Phi}'\boldsymbol{\Phi} + \frac{1}{\sigma_{\mathtt{w}}^2}\mathbf{I}\right)^{-1} \end{aligned}$$

All training observations contribute to the prediction - not a sparse representation

Gaussian Process Regression

The underlying equations for GPs are:

- input-prediction relation:

$$\begin{bmatrix} f(\boldsymbol{x}) \\ \boldsymbol{y} \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} k(\boldsymbol{x},\boldsymbol{x}) & \mathbf{k}(\boldsymbol{x},\boldsymbol{X})' \\ \mathbf{k}(\boldsymbol{x},\boldsymbol{X}) & \mathbf{K}(\boldsymbol{X},\boldsymbol{X}) + \sigma_{\mathtt{n}}^2\mathbf{I} \end{bmatrix}\right)$$

- prediction-target relation

$$y = f(\boldsymbol{x}) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_{\mathtt{n}}^2)$$

The prediction mean can then be expressed as

$$p(y|\boldsymbol{x},\boldsymbol{X},\boldsymbol{y}) = \mathcal{N}(y; \mu_{\mathtt{f}}, \sigma_{\mathtt{f}}^2 + \sigma_{\mathtt{n}}^2)$$

where

$$\begin{aligned} \mu_{\mathtt{f}} &= \mathbf{k}(\boldsymbol{x},\boldsymbol{X})'\left[\mathbf{K}(\boldsymbol{X},\boldsymbol{X}) + \sigma_{\mathtt{n}}^2\mathbf{I}\right]^{-1}\boldsymbol{y} \\ \sigma_{\mathtt{f}}^2 &= k(\boldsymbol{x},\boldsymbol{x}) - \mathbf{k}(\boldsymbol{x},\boldsymbol{X})'\left[\mathbf{K}(\boldsymbol{X},\boldsymbol{X}) + \sigma_{\mathtt{n}}^2\mathbf{I}\right]^{-1}\mathbf{k}(\boldsymbol{x},\boldsymbol{X}) \end{aligned}$$

where $\mathbf{K}(\boldsymbol{X},\boldsymbol{X})$ is determined by the covariance function

$$\mathbf{k}(\boldsymbol{x},\boldsymbol{X}) = \begin{bmatrix} k(\boldsymbol{x},\boldsymbol{x}_1) \\ \vdots \\ k(\boldsymbol{x},\boldsymbol{x}_n) \end{bmatrix}, \quad \mathbf{K}(\boldsymbol{X},\boldsymbol{X}) = \begin{bmatrix} k(\boldsymbol{x}_1,\boldsymbol{x}_1) & \dots & k(\boldsymbol{x}_n,\boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_1,\boldsymbol{x}_n) & \dots & k(\boldsymbol{x}_n,\boldsymbol{x}_n) \end{bmatrix}$$

Again all training observations contribute to the prediction - not a sparse representation

Relevance Vector Machine

The RVM is related to the basis regression form. However in order to create a more sparse representation the form of the weight prior is modified. Now

$$p(\boldsymbol{w}|\boldsymbol{\lambda}) = \prod_{i=1}^{n} \mathcal{N}(w_i; 0, \lambda_i^{-1}) = \mathcal{N}(\boldsymbol{w}; \mathbf{0}, \boldsymbol{\Lambda}^{-1})$$

Thus $\boldsymbol{\Lambda}$ is an inverse diagonal covariance matrix of the form

$$\boldsymbol{\Lambda} = \begin{bmatrix} \lambda_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \lambda_n \end{bmatrix}$$

Repeating the same process as before to get the weight posteriors yields the following values for the weight posteriors

$$p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{y}) = \mathcal{N}(\boldsymbol{w}; \boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w)$$

where

$$\begin{aligned} \boldsymbol{\mu}_w &= \frac{1}{\sigma_{\mathtt{n}}^2} \boldsymbol{\Sigma}_w \boldsymbol{\Phi}' \boldsymbol{y} \\ \boldsymbol{\Sigma}_w &= \left(\frac{1}{\sigma_{\mathtt{n}}^2} \boldsymbol{\Phi}' \boldsymbol{\Phi} + \boldsymbol{\Lambda} \right)^{-1} \end{aligned}$$

Thus for the RVM there are the following parameters to train

- noise variance: $\sigma_{\mathtt{n}}^2$ - the same as the Bayesian basis regression
- weight-specific variance: $\boldsymbol{\Lambda}$ - n parameters (matrix diagonal) rather than the single $\sigma_{\mathtt{w}}^2$

RVM Prediction

Prediction with the RVM is very similar to Bayesian basis prediction

$$p(y|\boldsymbol{x}, \boldsymbol{X}, \boldsymbol{y}) = \mathcal{N}(y; \boldsymbol{\phi}(\boldsymbol{x})'\boldsymbol{\mu}_w, \boldsymbol{\phi}(\boldsymbol{x})'\boldsymbol{\Sigma}_w\boldsymbol{\phi}(\boldsymbol{x}) + \sigma_{\mathrm{n}}^2)$$

Examining the just the mean of the prediction

$$\mu_y = \frac{1}{\sigma_{\mathrm{n}}^2}\boldsymbol{\phi}(\boldsymbol{x})' \left(\frac{1}{\sigma_{\mathrm{n}}^2}\boldsymbol{\Phi}'\boldsymbol{\Phi} + \boldsymbol{\Lambda}\right)^{-1} \boldsymbol{\Phi}'\boldsymbol{y}$$

it is interesting to see the impact of varying λ_i on the prediction of the RVM.

- $\lambda_i \to \infty$ this means that element i of $\boldsymbol{\phi}(\boldsymbol{x})$ will have no impact on the prediction.

Remembering that

$$\boldsymbol{\phi}(\boldsymbol{x}) = \left[\, \phi(||\boldsymbol{x} - \boldsymbol{x}_1||) \;\; \ldots \;\; \phi(||\boldsymbol{x} - \boldsymbol{x}_n||) \,\right]'$$

this means that training observation i will not alter the prediction - it's not relevant to the prediction process (see examples paper). This is related to the process of automatic relevance detection, and is the motivation for the name.

If a large number of the λ_i go to ∞ then there will be a sparse prediction.

In practice this is exactly what happens!

RVM Training

RVM training is based in the marginal likelihood

$$\begin{aligned} p(\boldsymbol{y}|\boldsymbol{X}) &= \int p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w}) p(\boldsymbol{w}|\boldsymbol{\Lambda}) d\boldsymbol{w} \\ &= \mathcal{N}(\boldsymbol{y}; \mathbf{0}, \boldsymbol{\Phi}\boldsymbol{\Lambda}\boldsymbol{\Phi}' + \sigma_{\mathrm{n}}^2 \mathbf{I}) \end{aligned}$$

The values of $\boldsymbol{\Lambda}$ and σ_{n}^2 are then found by maximising this marginal likelihood.

The parameters may be estimated in the same fashion as the hyper-parameters of the Gaussian process using gradient descent, noting that

$$\boldsymbol{\Phi}\boldsymbol{\Lambda}\boldsymbol{\Phi}' = \mathbf{K}(\boldsymbol{X}, \boldsymbol{X})$$

in the notation from the previous handout.

Alternatively expectation maximisation (EM) can be used. Here the value of the weight is treated as a continuous latent variable. This yields the auxiliary function

$$\begin{aligned} \mathcal{Q}(\boldsymbol{\theta}^{(k)}, \boldsymbol{\theta}^{(k+1)}) &= \int p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\theta}^{(k)}) \log \left(p(\boldsymbol{w}, \boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}^{(k+1)}) \right) d\boldsymbol{w} \\ &= \int p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\theta}^{(k)}) \log \left(p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w}, \boldsymbol{\theta}^{(k+1)}) \right) d\boldsymbol{w} \\ &\quad + \int p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\theta}^{(k)}) \log \left(p(\boldsymbol{w}|\boldsymbol{\theta}^{(k+1)}) \right) d\boldsymbol{w} \end{aligned}$$

where

$$\boldsymbol{\theta} = \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_n \\ \sigma_{\mathrm{n}}^2 \end{bmatrix}$$

RVM Training (cont)

From the previous slides

$$\begin{aligned} p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w}, \boldsymbol{\theta}^{(k+1)}) &= \mathcal{N}(\boldsymbol{y}; \boldsymbol{\Phi}\boldsymbol{w}, \sigma_{\mathtt{n}}^{(k+1)2}) \\ p(\boldsymbol{w}|\boldsymbol{\theta}^{(k+1)}) &= \mathcal{N}(\boldsymbol{w}; \mathbf{0}, \boldsymbol{\Lambda}^{(k+1)-1}) \\ p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\theta}^{(k)}) &= \mathcal{N}(\boldsymbol{w}, \boldsymbol{\mu}_w^{(k)}, \boldsymbol{\Sigma}_w^{(k)}) \end{aligned}$$

Dealing with the two elements separately yields

- weights inverse variance

$$\lambda_i^{(k+1)} = \frac{1}{\sigma_{wi}^{(k)2} + \mu_{wi}^{(k)2}}$$

- noise variance

$$\sigma_{\mathtt{n}}^{(k+1)2} = \frac{1}{n}\left((\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\mu}_w^{(k)})'(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\mu}_w^{(k)}) + \text{tr}\left(\boldsymbol{\Phi}'\boldsymbol{\Phi}\boldsymbol{\Sigma}_w^{(k)}\right)\right)$$

This is an iterative process, at each iteration the marginal likelihood is guaranteed not to decrease.

For reference: it is possible to introduce priors on the hyper-parameters. Suitable priors are Gamma distributions

$$\begin{aligned} p(\boldsymbol{\Lambda}) &= \prod_{i=1}^{n} \text{Gamma}(\lambda_i|a, b) \\ p(\sigma_{\mathtt{n}}^{-2}) &= \text{Gamma}(\sigma_{\mathtt{n}}^{-2}|c, d) \end{aligned}$$

where

$$\begin{aligned} \text{Gamma}(\lambda|a, b) &= \Gamma(a)^{-1} b^a \lambda^{a-1} \exp(-b\lambda) \\ \Gamma(a) &= \int_0^\infty t^{a-1} \exp(-t) dt \end{aligned}$$

RVM EM Training (details)

The estimation of the weights inverse variance

$$\boldsymbol{\Lambda}^{(k+1)} = \arg\max_{\Lambda} \left\{ \int p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\theta}^{(k)}) \log\left(p(\boldsymbol{w}|\boldsymbol{\Lambda})\right) d\boldsymbol{w} \right\}$$

Noting that the mean for $p(\boldsymbol{w}|\boldsymbol{\Lambda}^{(k+1)})$ is constrained to be zero and $\boldsymbol{\Lambda}$ is diagonal yields the update formulae.

The estimation of the noise variance is given by

$$\sigma_{\mathtt{n}}^{(k+1)2} = \arg\max_{\sigma} \left\{ \int \mathcal{N}(\boldsymbol{w}, \boldsymbol{\mu}_w^{(k)}, \boldsymbol{\Sigma}_w^{(k)}) \log\left(p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w}, \sigma^2)\right) d\boldsymbol{w} \right\}$$

The function of σ^2 to find may be written as

$$\log\left(p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w}, \sigma^2)\right) = -\frac{n}{2}\log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{w})'(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{w})$$

Considering part of the second term on the RHS (expectation over $\boldsymbol{w}$)

$$\mathcal{E}\left\{(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{w})'(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{w})\right\} = (\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\mu}_w^{(k)})'(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\mu}_w^{(k)}) + \mathrm{tr}\left(\boldsymbol{\Phi}'\boldsymbol{\Phi}\boldsymbol{\Sigma}_w^{(k)}\right)$$

Noting this uses the equality

$$\begin{aligned}
\mathcal{E}\left\{\boldsymbol{w}'\boldsymbol{\Phi}'\boldsymbol{\Phi}\boldsymbol{w}\right\} &= \mathrm{tr}\left(\boldsymbol{\Phi}'\boldsymbol{\Phi}\mathcal{E}\left\{\boldsymbol{w}\boldsymbol{w}'\right\}\right) \\
&= \mathrm{tr}\left(\boldsymbol{\Phi}'\boldsymbol{\Phi}\left(\boldsymbol{\Sigma}_w^{(k)} + \boldsymbol{\mu}_w^{(k)}\boldsymbol{w}_w^{(k)\prime}\right)\right) \\
&= \mathrm{tr}\left(\boldsymbol{\Phi}'\boldsymbol{\Phi}\boldsymbol{\Sigma}_w^{(k)}\right) + \boldsymbol{\mu}_w^{(k)\prime}\boldsymbol{\Phi}'\boldsymbol{\Phi}\boldsymbol{\mu}_w^{(k)}
\end{aligned}$$

Note: the update expression can be further simplified to yield

$$\sigma_{\mathtt{n}}^{(k+1)2} = \frac{1}{n}\left((\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\mu}_w^{(k)})'(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\mu}_w^{(k)}) + \sigma_{\mathtt{n}}^{(k)2}\sum_{i=1}^{n}(1 - \lambda_i^{(k)}\sigma_{wi}^{(k)2})\right)$$

RVM Classification

RVM classification uses exactly the same form as classification with Gaussian processes. Again the regression "prediction" distribution can be mapped using either:

- logistic regression: here

$$P(y = +1|\boldsymbol{x}, \boldsymbol{w}) = \frac{1}{1 + \exp(-\boldsymbol{w}'\boldsymbol{\phi}(\boldsymbol{x}))}$$

- probit regression: again considering linear regression

$$P(y = +1|\boldsymbol{x}, \boldsymbol{w}) = \int_{-\infty}^{\boldsymbol{w}'\boldsymbol{\phi}(\boldsymbol{x})} \mathcal{N}(z; 0, 1)dz$$

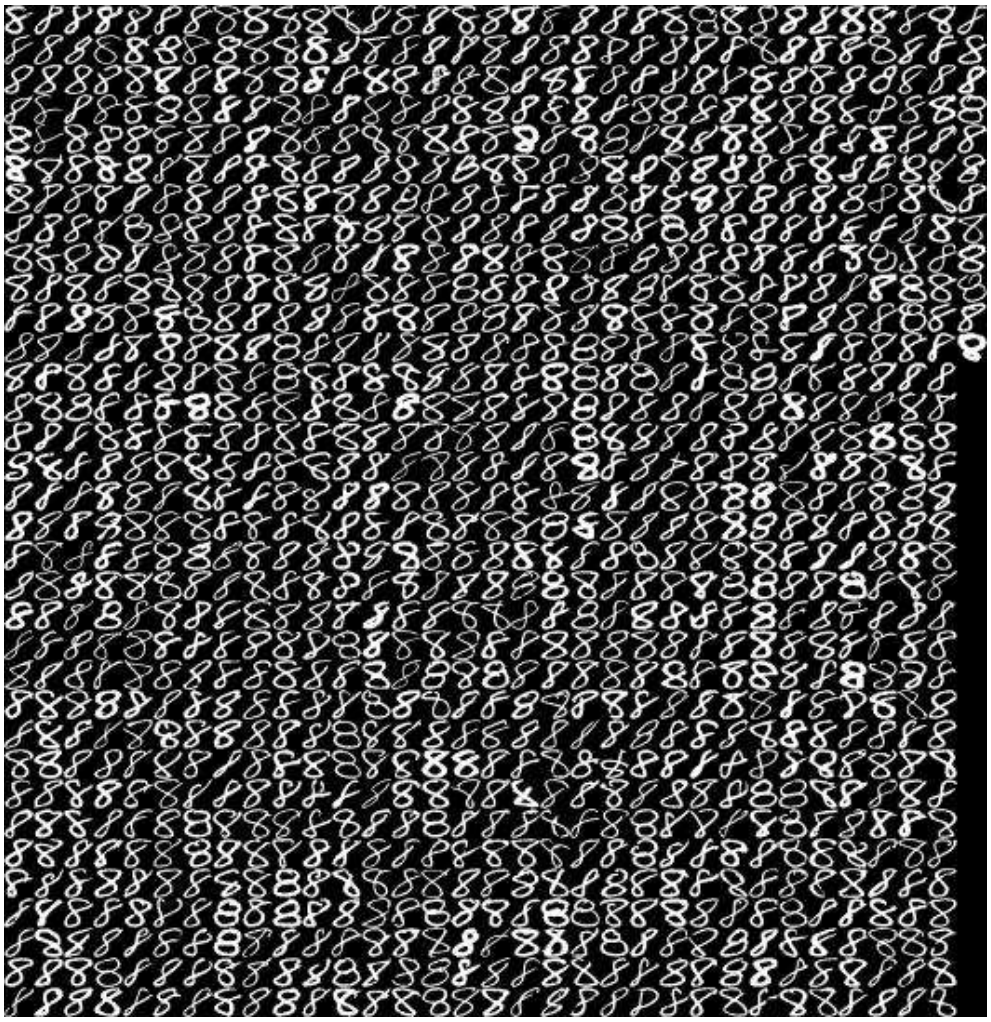
Classification is then based on

$$P(y = +1|\boldsymbol{x}, \boldsymbol{X}, \boldsymbol{y}) = \int P(y = +1|\boldsymbol{x}, \boldsymbol{w})p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{y})d\boldsymbol{w}$$

Laplace's approximation is again often used to handle the non-Gaussian nature that results from the mapping.

RVM Example Application - USPS digits

Recognition of handwritten digits - standard task - all the eights shown below



Train a set of 1-v-rest classifiers using either an SVM or RVM with Gaussian kernels.

Classifier	Training		Number SVs/RVs	Errors %
	n	d		
SVM	7291	256	2540	4.4
RVM			316	5.1

RVM Summary

Various aspects of RVMs examined:

- relationship to basis function regression
- a sparse representation using the weight prior
- training using EM
- classification in same fashion as Gaussian processes