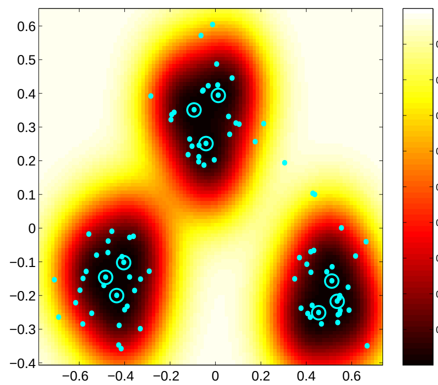
Bayesian Inference: Principles and Practice

4. Sparse Bayesian Models: Analysis, Optimisation and Applications

Mike Tipping



Microsoft Research *Cambridge, UK*

Lecture 4: Overview

- Further analysis of the sparse Bayesian marginal likelihood function

- Based on this, an improved optimisation algorithm

- Extensions and applications of (sparse) Bayesian models

The Marginal Likelihood Function

- We integrated out weights $\mathbf{w}$ to obtain *marginal likelihood*:

$$
\begin{aligned}
p(\mathbf{t}|\boldsymbol{\alpha},\sigma^2) &= \int p(\mathbf{t}|\mathbf{w},\sigma^2)\, p(\mathbf{w}|\boldsymbol{\alpha})\, d\mathbf{w}, \\
&= (2\pi)^{-N/2}|\mathbf{C}|^{-1/2} \exp\left\{-\frac{1}{2}\mathbf{t}^{\mathsf{T}}\mathbf{C}^{-1}\mathbf{t}\right\}
\end{aligned}
$$

  with $\mathbf{C} = \sigma^2\mathbf{I} + \sum_m \alpha_m^{-1}\phi_m\phi_m^{\mathsf{T}}$

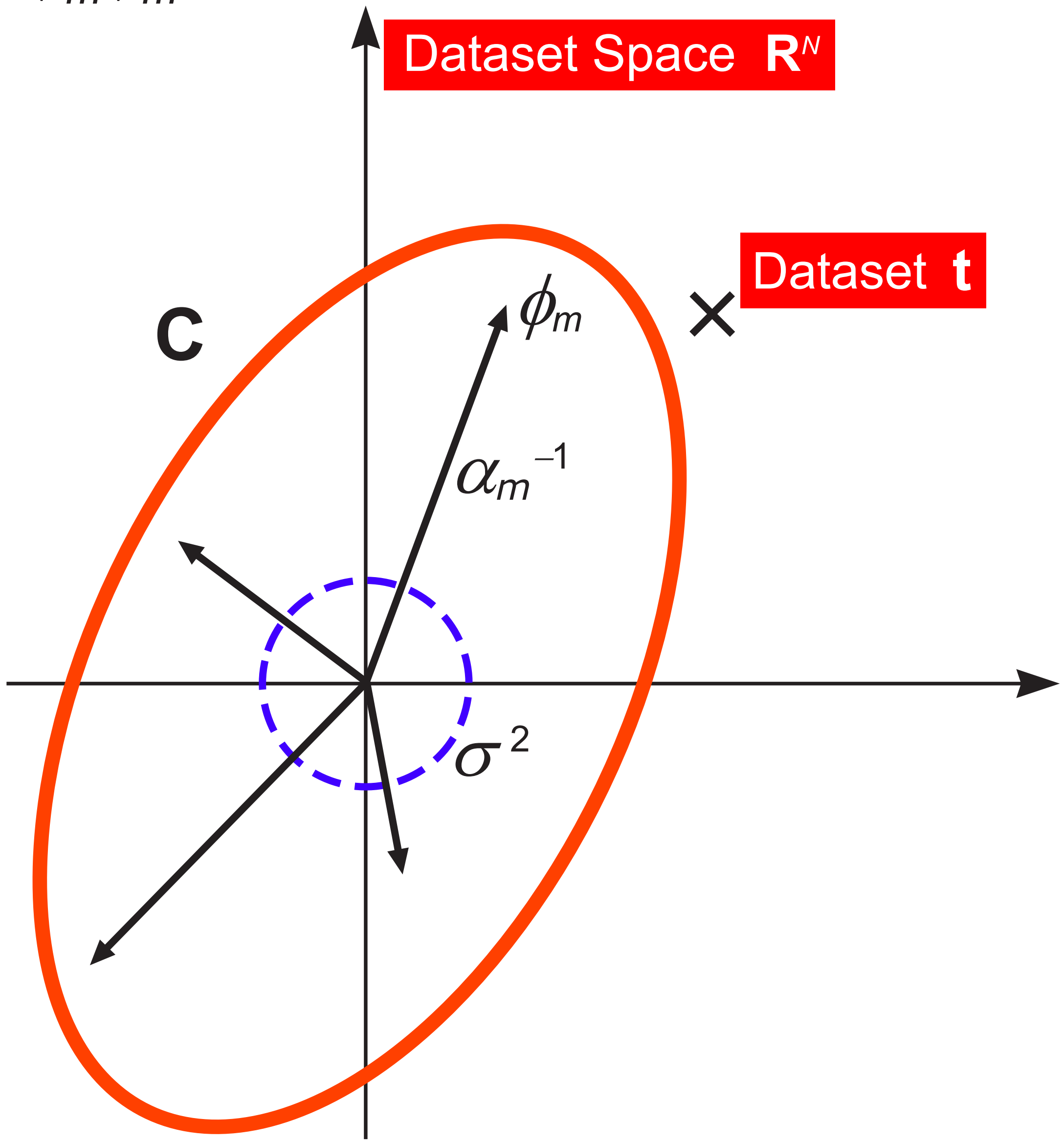
- Further integration over $\boldsymbol{\alpha}$ intractable

- We maximise $p(\boldsymbol{\alpha},\sigma^2|\mathbf{t})$ to find $\boldsymbol{\alpha}_{\mathrm{MP}}$ and σ^2_{MP}

- For uniform hyperpriors, equivalent to maximising $p(\mathbf{t}|\boldsymbol{\alpha},\sigma^2)$

That Picture Again. . .

$$\mathbf{C} = \sigma^2\mathbf{I} + \sum_m \alpha_m^{-1}\phi_m\phi_m^{\mathsf{T}}$$



Dependence on a Single Hyperparameter (1)

- Our objective is to maximise:

$$\log p(\mathbf{t}|\alpha,\sigma^2) = -\frac{1}{2}\left[\log|\mathbf{C}| + \mathbf{t}^{\mathsf{T}}\mathbf{C}^{-1}\mathbf{t}\right] + \text{constant terms}$$

- Decompose:

$$\begin{aligned}\mathbf{C} &= \sigma^2\mathbf{I} + \sum_{m\neq i}\alpha_m^{-1}\phi_m\phi_m^{\mathsf{T}} + \alpha_i^{-1}\phi_i\phi_i^{\mathsf{T}}\\ &= \mathbf{C}_{-i} + \alpha_i^{-1}\phi_i\phi_i^{\mathsf{T}}\end{aligned}$$

- Now we exploit some established matrix identities:

$$|\mathbf{C}| = |\mathbf{C}_{-i}|\,|1 + \alpha_i^{-1}\phi_i^{\mathsf{T}}\mathbf{C}_{-i}^{-1}\phi_i|$$

$$\mathbf{C}^{-1} = \mathbf{C}_{-i}^{-1} - \frac{\mathbf{C}_{-i}^{-1}\phi_i\phi_i^{\mathsf{T}}\mathbf{C}_{-i}^{-1}}{\alpha_i + \phi_i^{\mathsf{T}}\mathbf{C}_{-i}^{-1}\phi_i}$$

Dependence on a Single Hyperparameter (2)

- $\log p(\mathbf{t}|\boldsymbol{\alpha}, \sigma^2)$ can then be written in the form:

$$\log p(\mathbf{t}|\boldsymbol{\alpha}_{-i}, \sigma^2) + \frac{1}{2}\left[\log \alpha_i - \log(\alpha_i + s_i) + \frac{q_i^2}{\alpha_i + s_i}\right]$$

  where $\log p(\mathbf{t}|\boldsymbol{\alpha}_{-i}, \sigma^2)$ is independent of α_i

- For convenience, "quality" and "sparsity" terms have been defined:

$$q_i = \phi_i^{\mathrm{T}} \mathbf{C}_{-i}^{-1} \mathbf{t}$$

$$s_i = \phi_i^{\mathrm{T}} \mathbf{C}_{-i}^{-1} \phi_i$$

- Note these terms are independent of α_i (but depend on all other $\boldsymbol{\alpha}_{-i}$)

Maxima of the Marginal Likelihood

- Dependence of marginal likelihood on single hyperparameter α_i is captured by:

$$\ell(\alpha_i) = \log \alpha_i - \log(\alpha_i + s_i) + \frac{q_i^2}{\alpha_i + s_i}$$

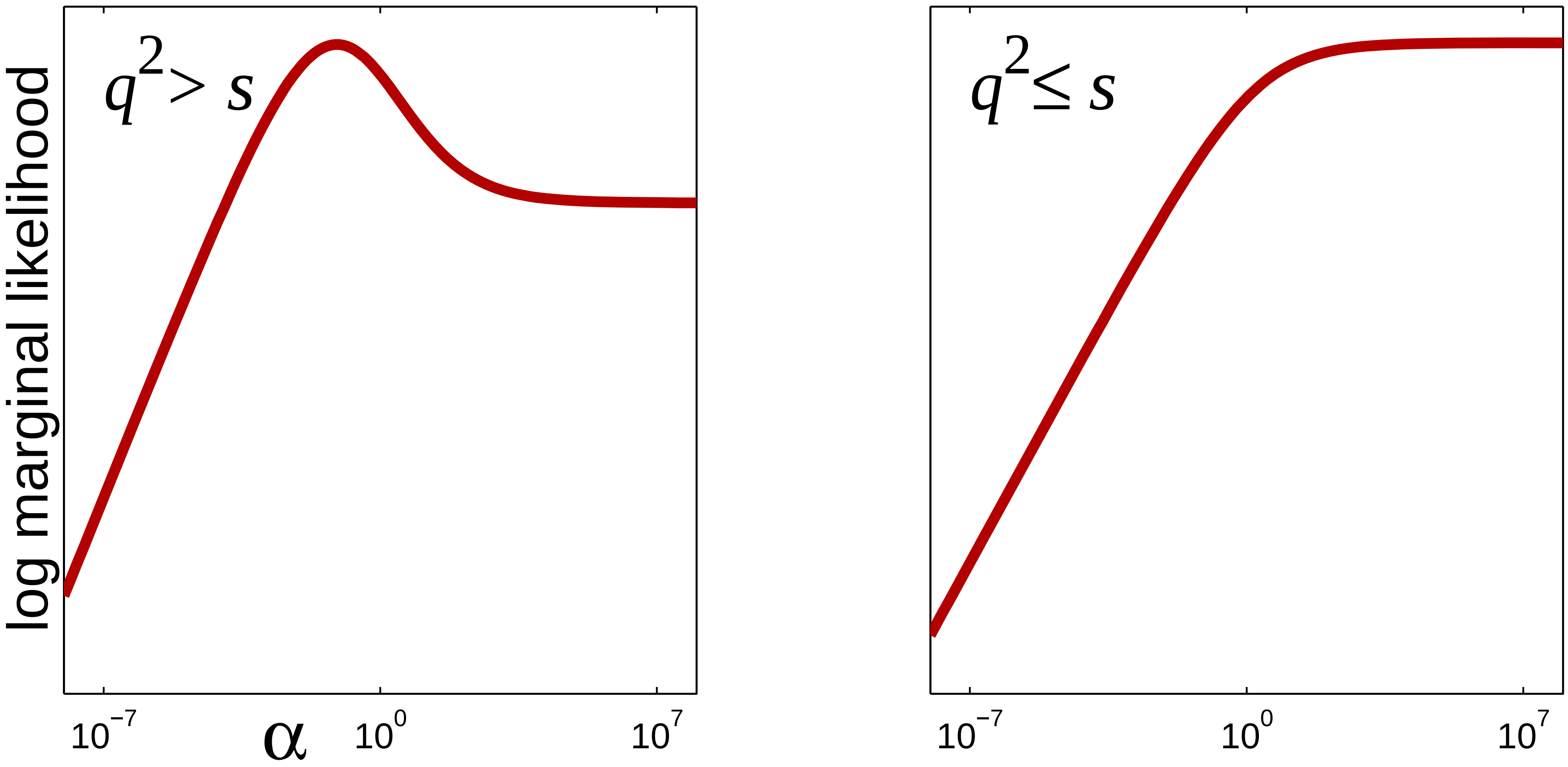
- Setting $\partial\ell(\alpha_i)/\partial\alpha_i = 0$ gives analytic solutions:
  - If $q_i^2 > s_i$:

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

  - If $q_i^2 \leq s_i$:

$$\alpha_i^{\text{opt}} = \infty$$

Maxima Visualised



Optimisation Operations

- For any given basis function $\phi_i(\mathbf{x})$ and associated hyperparameter α_i we can compute the quantities s_i and q_i^2 (true even if $\alpha_i = \infty$)

- Depending on the criterion $q_i^2 > s_i$ and the value of α_i we can then perform the following updates, all of which will increase $p(\mathbf{t}|\boldsymbol{\alpha}, \sigma^2)$:

	"In model": $\alpha_i < \infty$	**"Out of model"**: $\alpha_i = \infty$
$q_i^2 > s_i$	*re-estimation* of α_i	*addition* of $\phi_i(\mathbf{x})$
$q_i^2 \leq s_i$	*deletion* of $\phi_i(\mathbf{x})$	—

Optimisation Algorithm Sketch

❶ Initialise σ^2 sensibly and all $\alpha_m = \infty$ (*i.e.* the 'empty' model)

❷ Select a function $\phi_i(\mathbf{x})$ from the set of all M

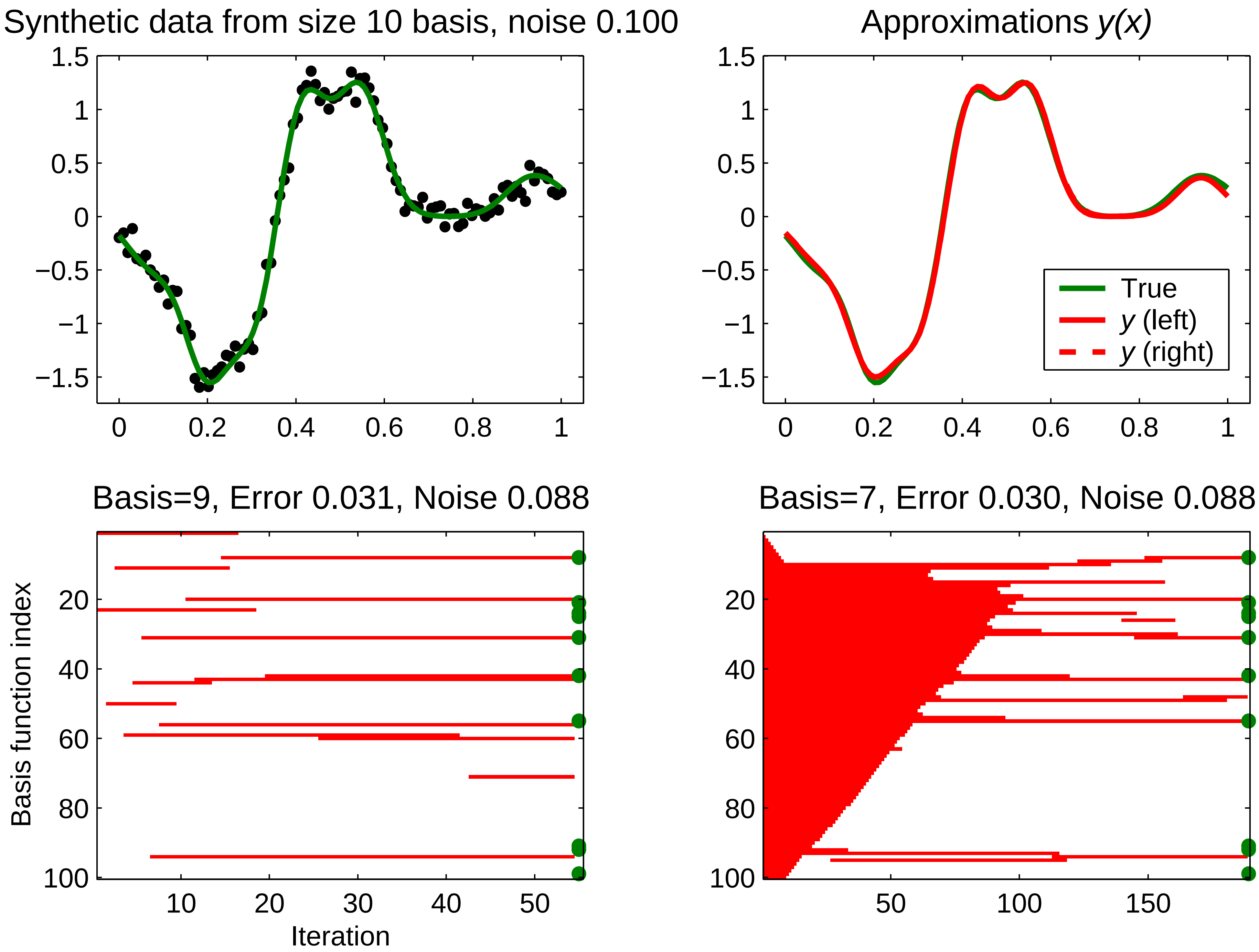
❸ Compute "relevance" $\mathcal{R}_i \triangleq q_i^2 - s_i$

- If $\mathcal{R}_i > 0$ and $\alpha_i < \infty$: **re-estimate** α_i
- If $\mathcal{R}_i > 0$ and $\alpha_i = \infty$: **add** ϕ_i to the model with updated α_i
- If $\mathcal{R}_i \leq 0$ and $\alpha_i < \infty$: **delete** ϕ_i from the model and set $\alpha_i = \infty$

❹ If estimating the noise level, update σ^2

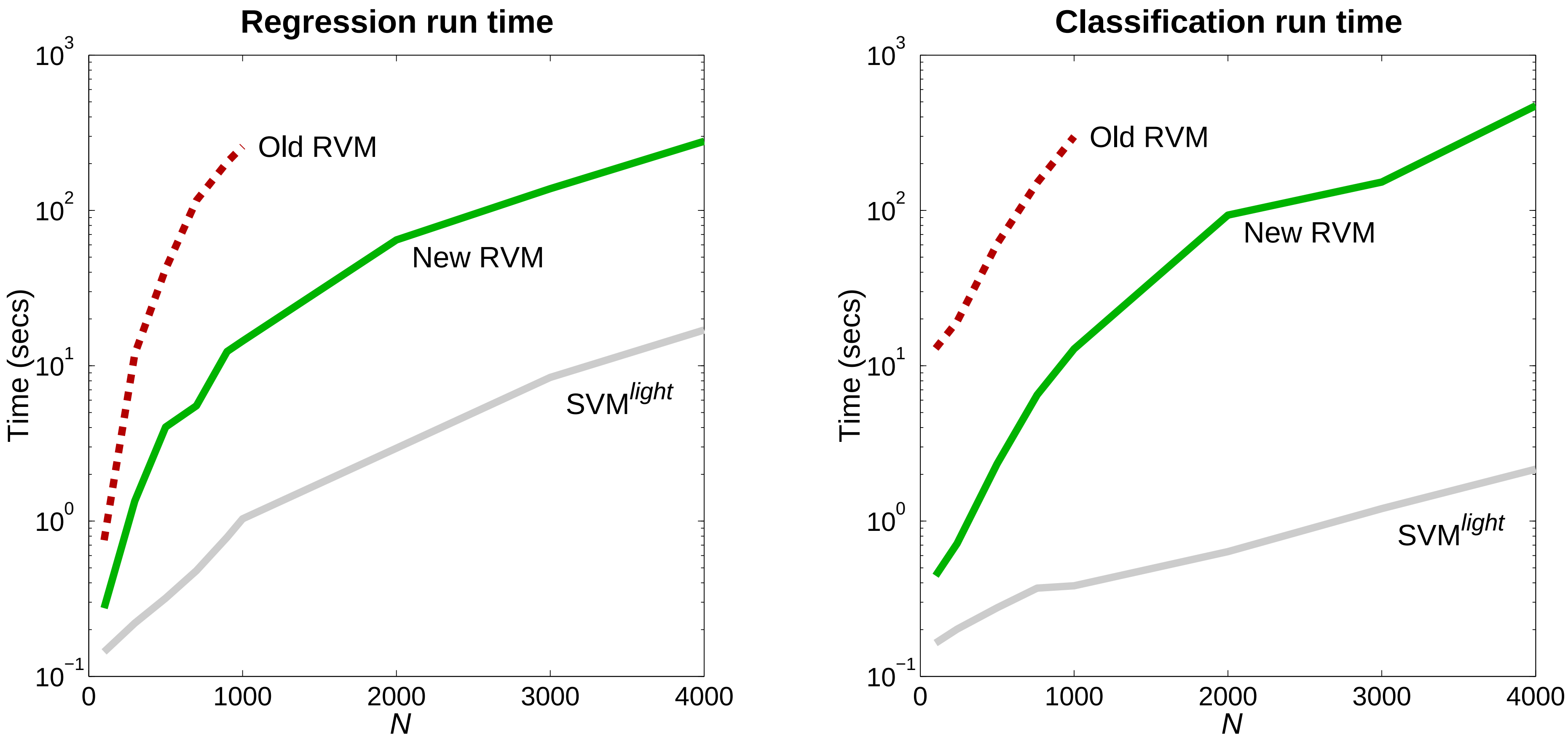
❺ Recalculate all q_m and s_m

❻ If converged terminate, otherwise goto ❷



Synthetic data from size 10 basis, noise 0.100
Approximations y(x)
True
y (left)
y (right)
Basis=9, Error 0.031, Noise 0.088
Basis=7, Error 0.030, Noise 0.088
Basis function index
Iteration

Performance Illustration: run time



Performance Illustration: example timing

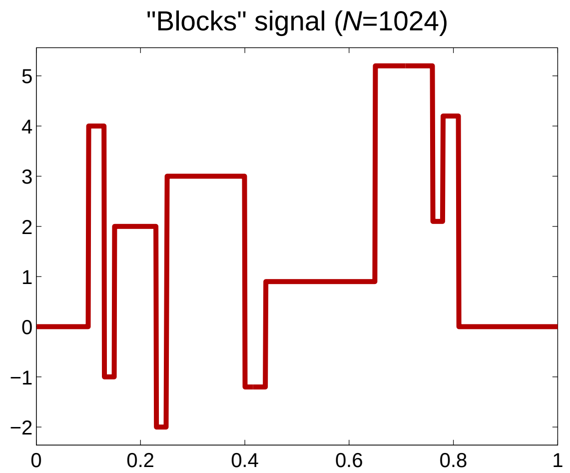
- Comparing at $N = 1000$ we have:

	Regression	Classification
Old RVM	4 mins 17 secs	4 mins 58 secs
New RVM	14.42 secs	12.84 secs
SVM^{light}	1.03 secs	0.38 secs

Greediness?

- Agglomerative algorithms (*e.g.* “matching pursuit”) are often *greedy* — *i.e.* “early” additions can be significantly sub-optimal

- Demonstration: a popular signal processing test data set



- Approximate with a basis comprising:
  - “heaviside” step functions (easy)
  - “heaviside” *and* Gaussians (hard?)

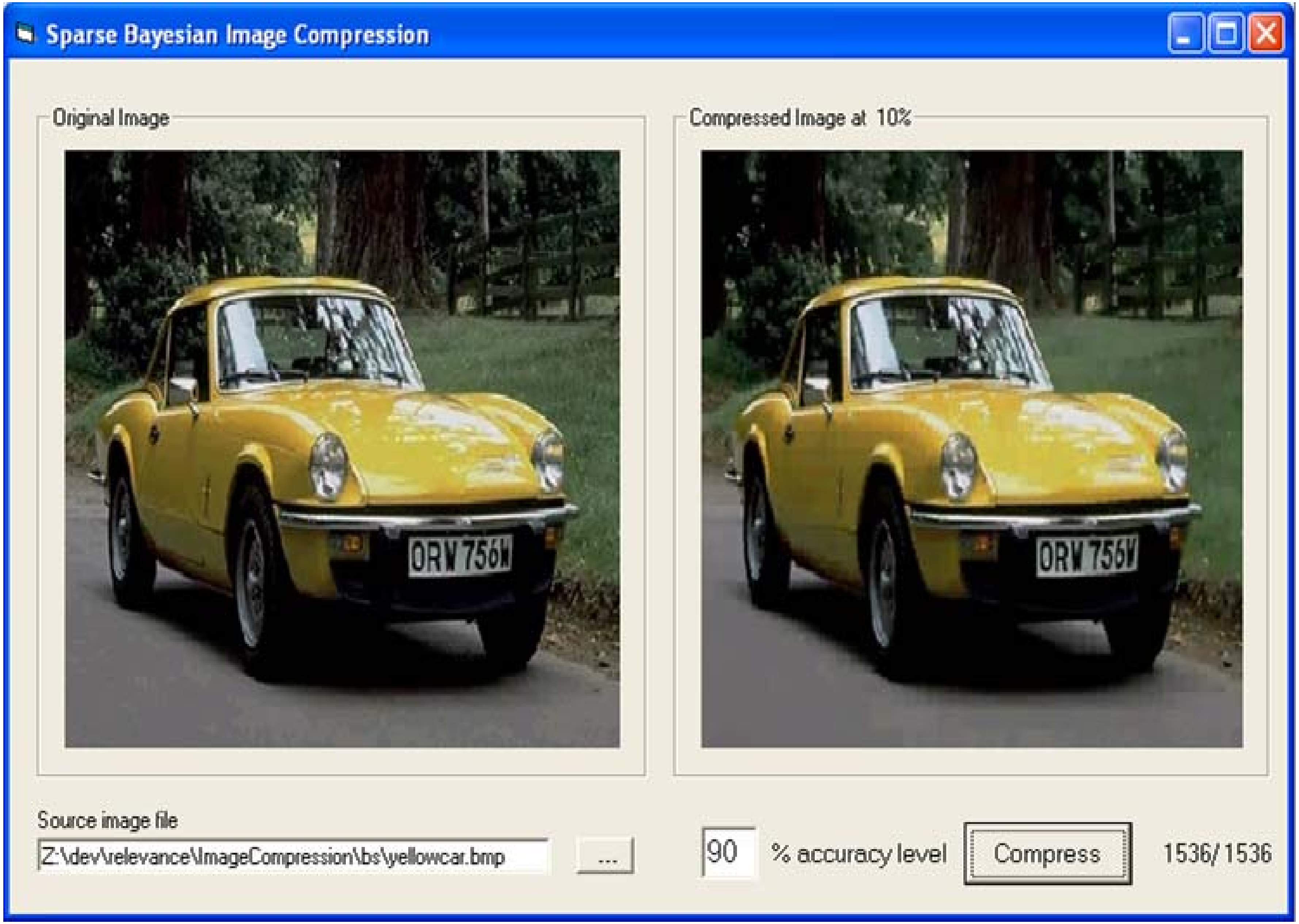
“Blocks” Data Results Summary

	Heaviside		*Heaviside + Gauss*	
	Bayes	**ORMP**	**Bayes**	**ORMP**
M	1024	1024	5120	5120
$\widehat{M}$	**12**	**12**	**12**	**82**
Iterations	21	11	224	82
Additions	11	11	107	82
Deletions	0	–	96	–
Re-estimates	10	–	21	–
Time	1.34s	1.19s	43.3s	24.6s

Applications: approximation (1)

- Assume the target is noise-free and is to be approximated more ‘cheaply’, *e.g.* an image which is to be compressed
- Choose some appropriate basis set (*e.g.* Gabor wavelets)
- Fix σ^2 as desired
- Run the sparse Bayes regression algorithm
- Interpretation of σ^2 has changed — it now models the approximation error, not the noise process

Applications: image compression



Applications: approximation (2)

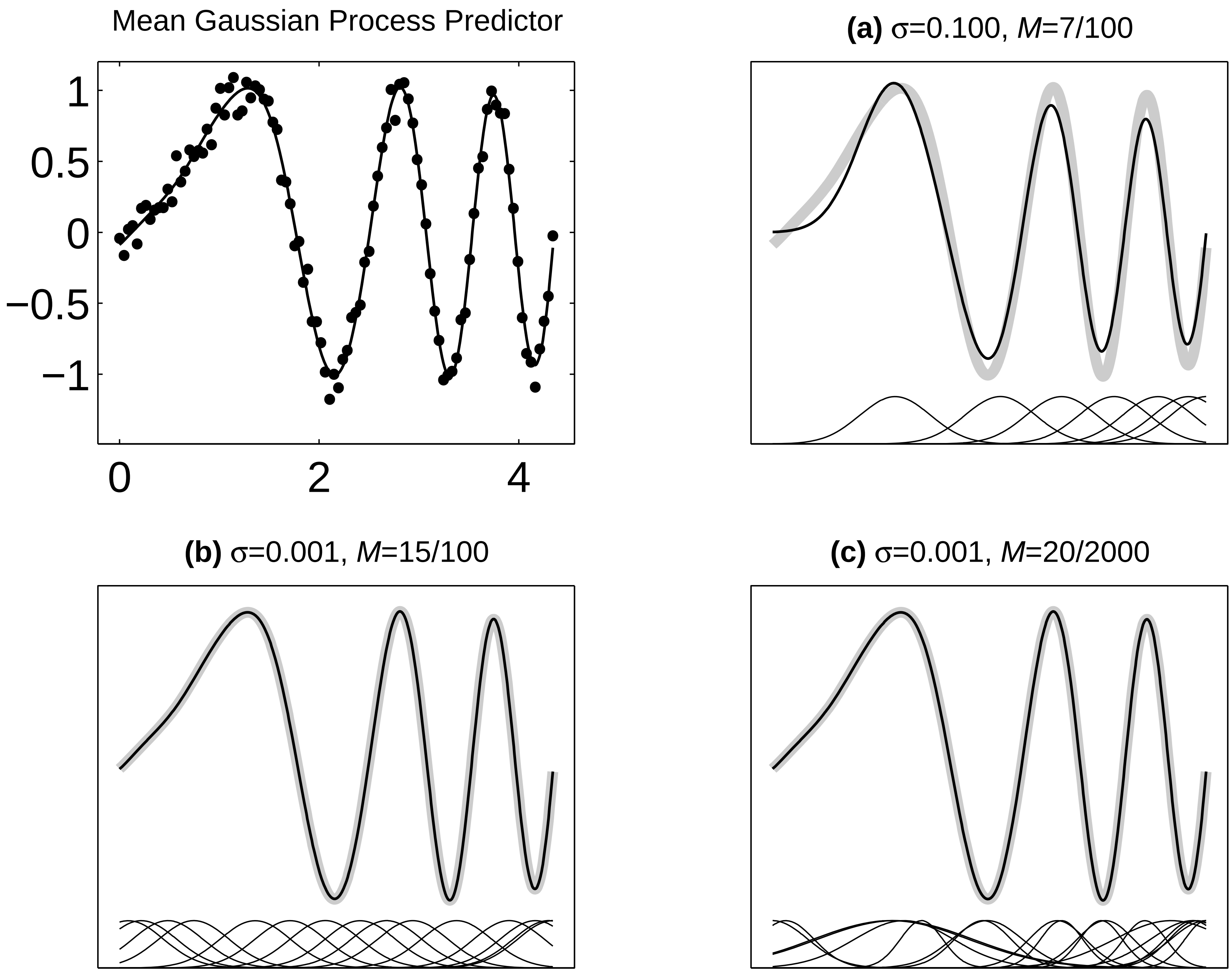
- Can approximate *functions* $f(\mathbf{x})$:

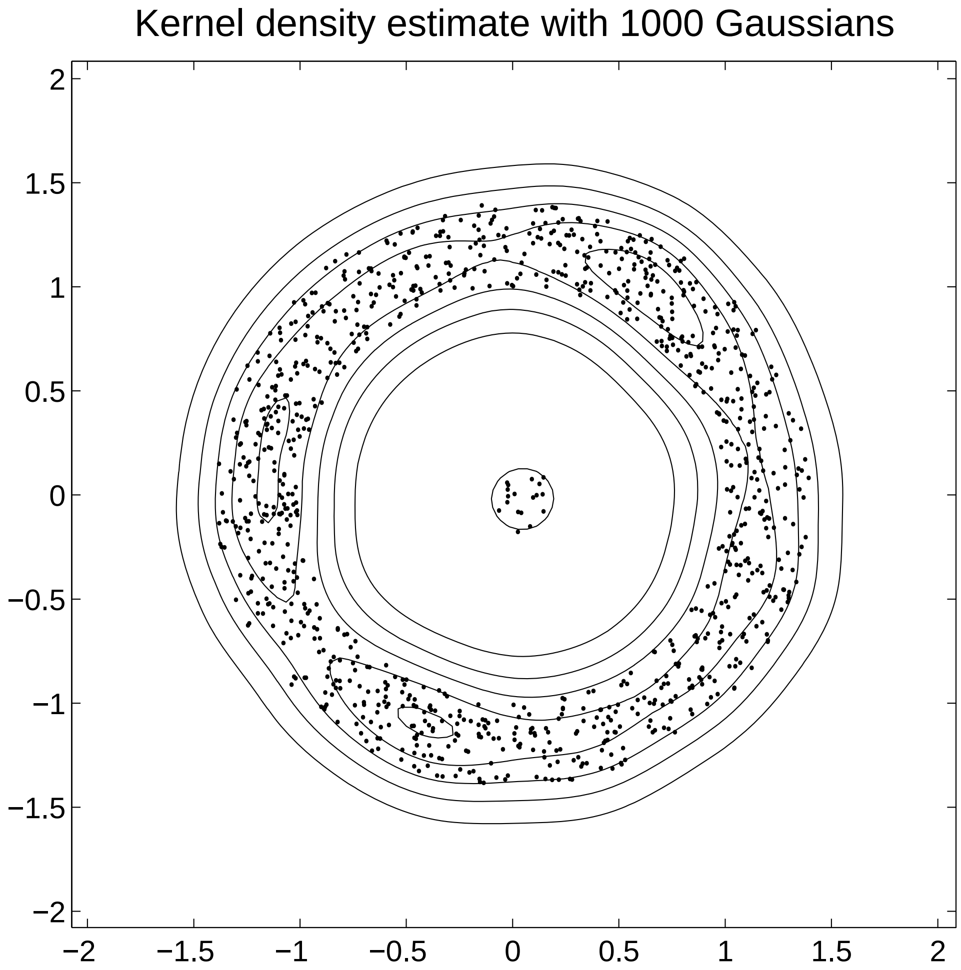
$$\text{Likelihood} \propto \exp\left\{-\frac{1}{2\sigma^2}\int \|y(\mathbf{x};\mathbf{w}) - f(\mathbf{x})\|^2\, d\mathbf{x}\right\}$$

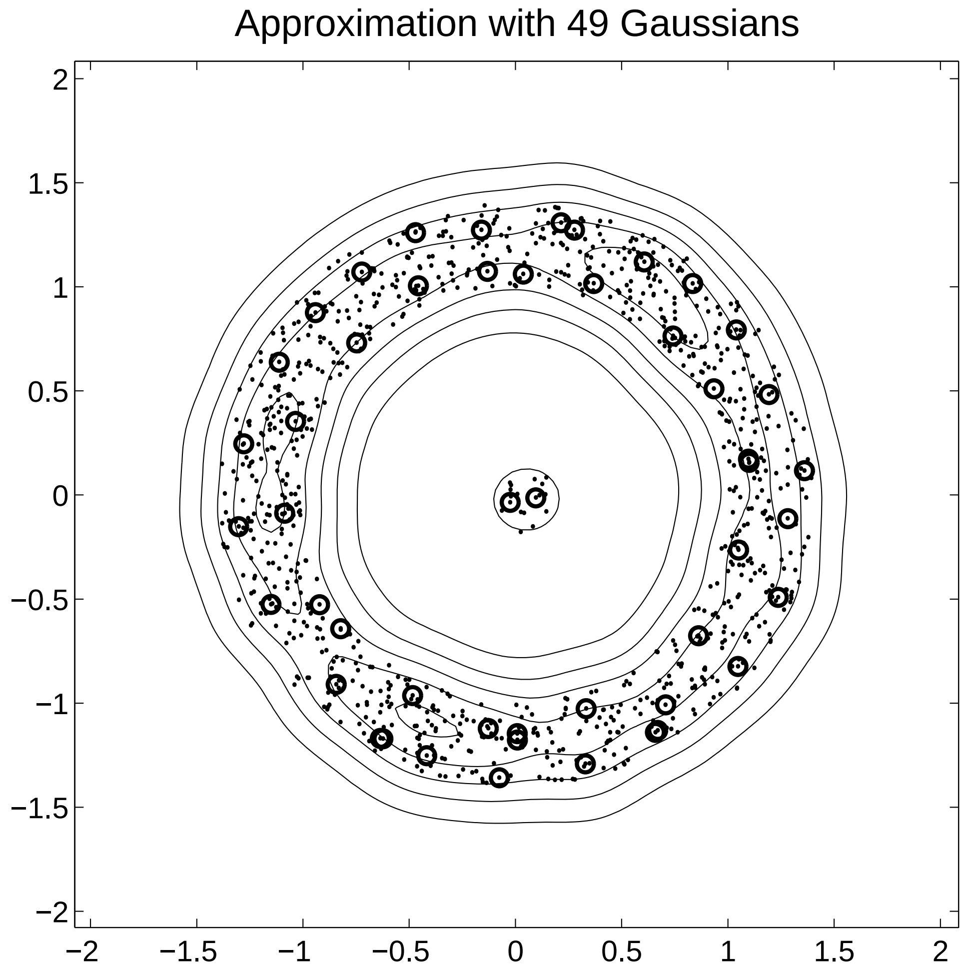
- Condition: we need to compute all $\int \phi_i(\mathbf{x}) f(\mathbf{x})\, d\mathbf{x}$ and $\int \phi_i(\mathbf{x})\phi_j(\mathbf{x})\, d\mathbf{x}$

- Practical example: $f(\mathbf{x}) = \sum_j v_j \psi_j(\mathbf{x})$ with ψ_j Gaussian

- Potential target functions: Gaussian process, SVM, kernel density estimator *etc*

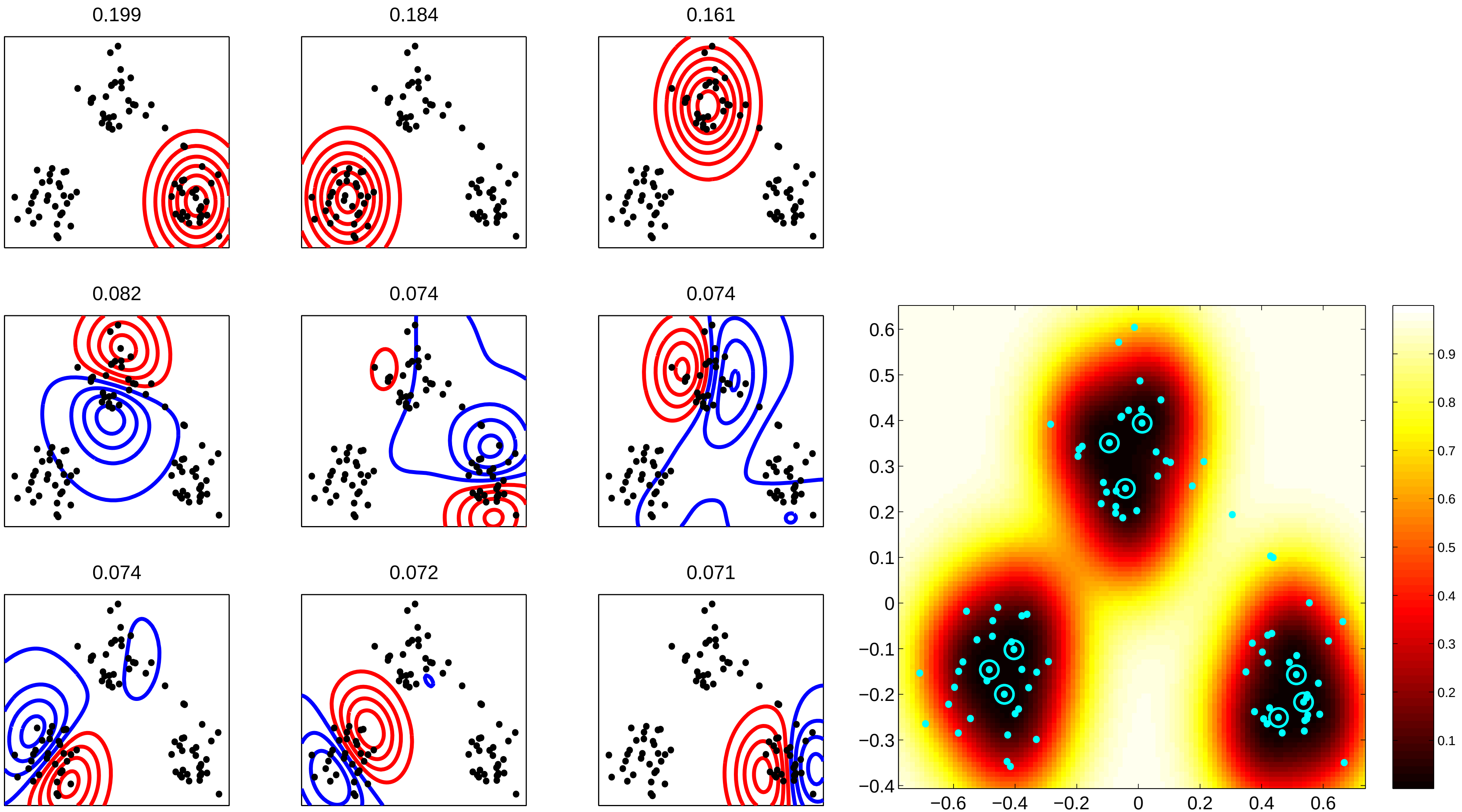

Mean Gaussian Process Predictor
1
0.5
0
−0.5
−1
0
2
4
(a) σ=0.100, M=7/100
(b) σ=0.001, M=15/100
(c) σ=0.001, M=20/2000


Kernel density estimate with 1000 Gaussians
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2 −1.5 −1 −0.5 0 0.5 1 1.5 2


Approximation with 49 Gaussians
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2 −1.5 −1 −0.5 0 0.5 1 1.5 2

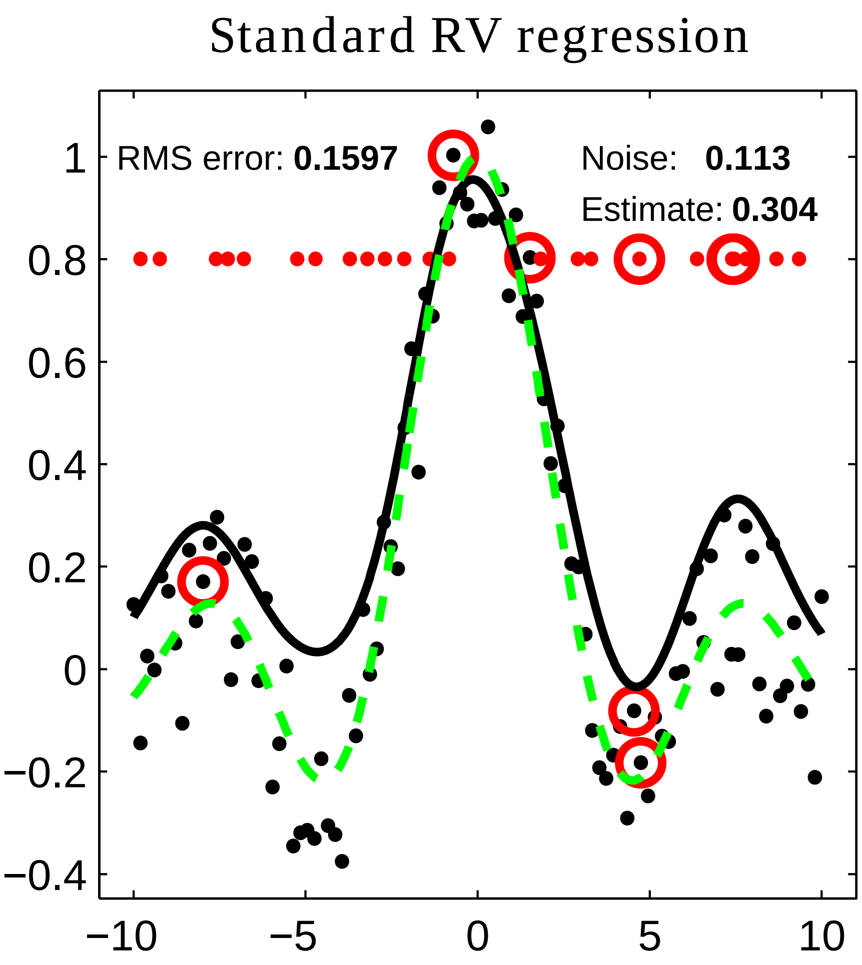
Applications: sparse kernel PCA

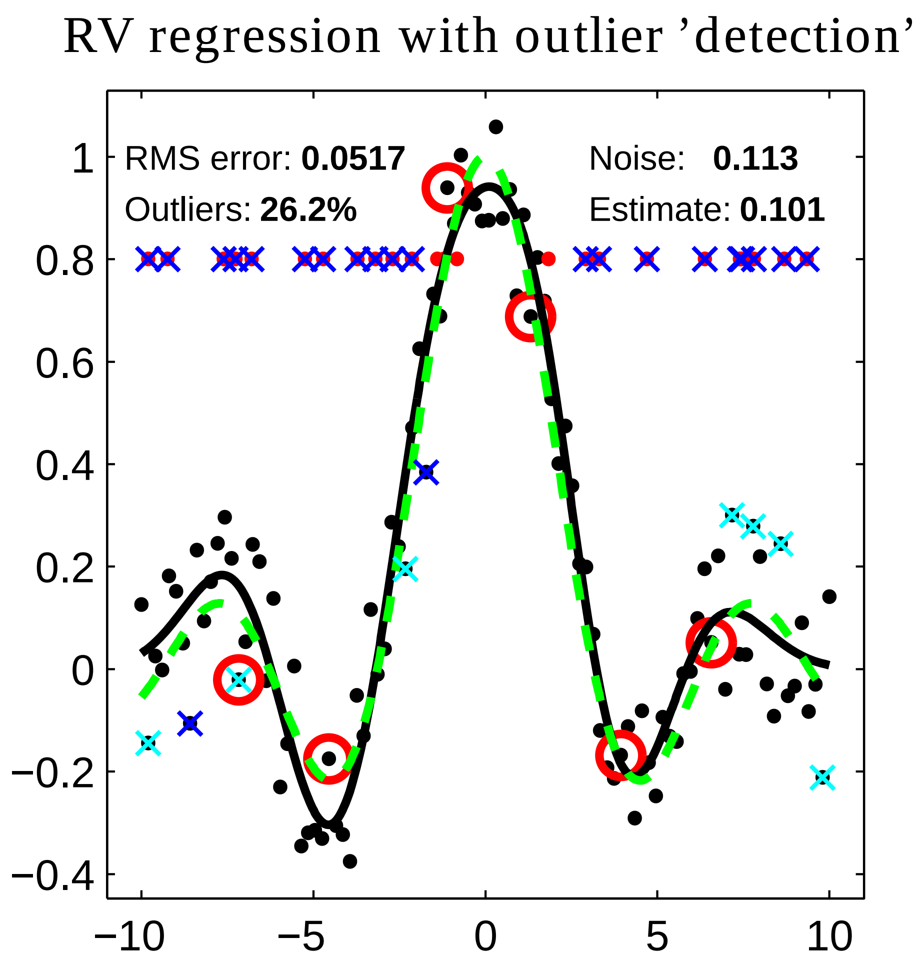
- Work directly with $\mathbf{C} = \sum_{n=1}^{N} \alpha_n^{-1} \phi_n \phi_n^{\mathsf{T}} + \sigma^2 \mathbf{I}$



Applications: robust regression

- Exploit *variational* formalism to incorporate outlier distribution. *i.e.* ‘mixture’ likelihood: $p(\mathbf{t}|\mathbf{w}) = \theta.p_{\text{data}}(\mathbf{t}|\mathbf{w}) + (1-\theta).p_{\text{outlier}}(\mathbf{t})$

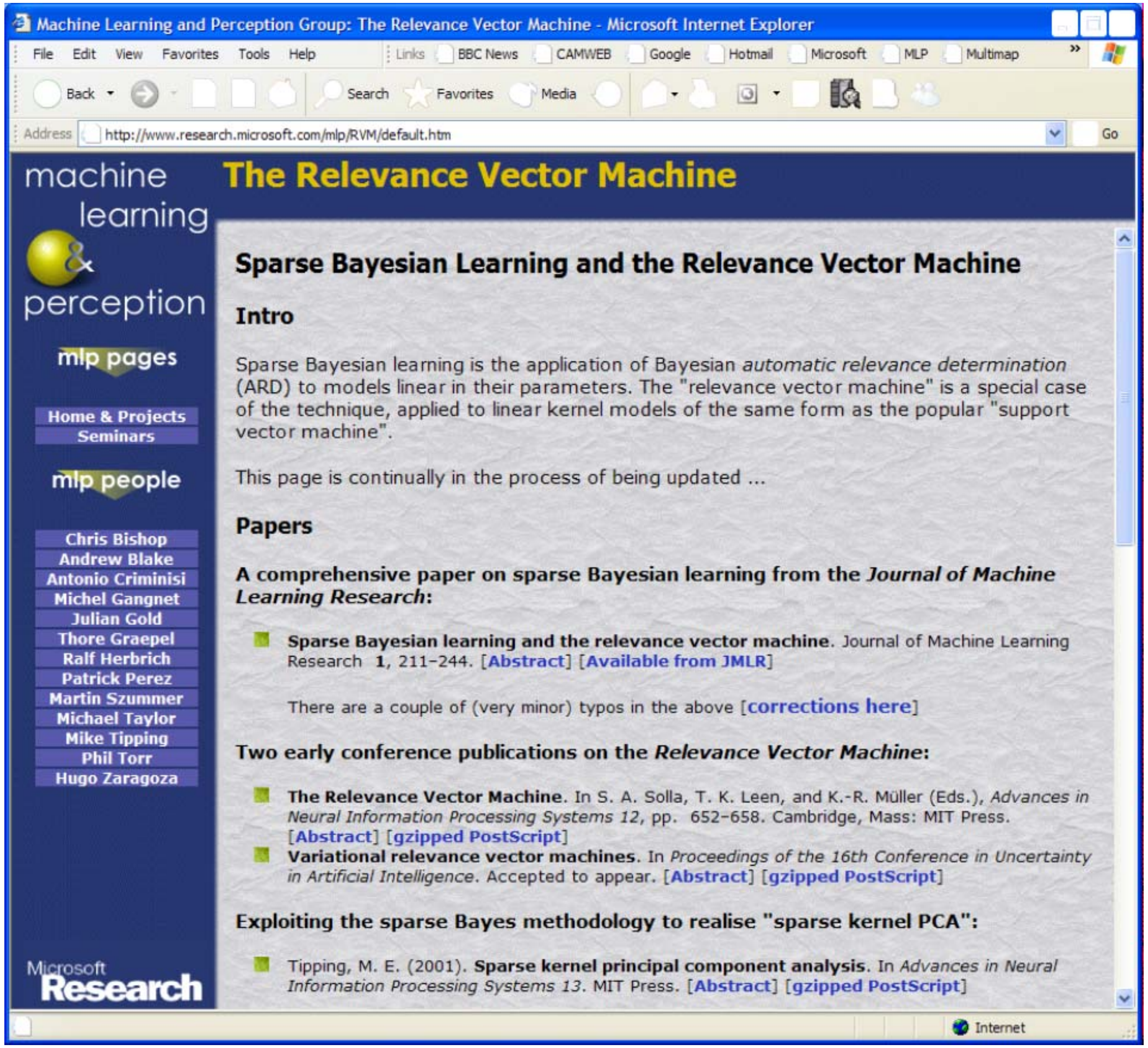




Applications: Image Super-Resolution

Exploits marginalisation over the unknown high-resolution image to optimise registration parameters

More Information



`http://www.research.microsoft.com/mlp/RVM/`