Joint Gaussian Process–Density Mixtures

Ole Winther

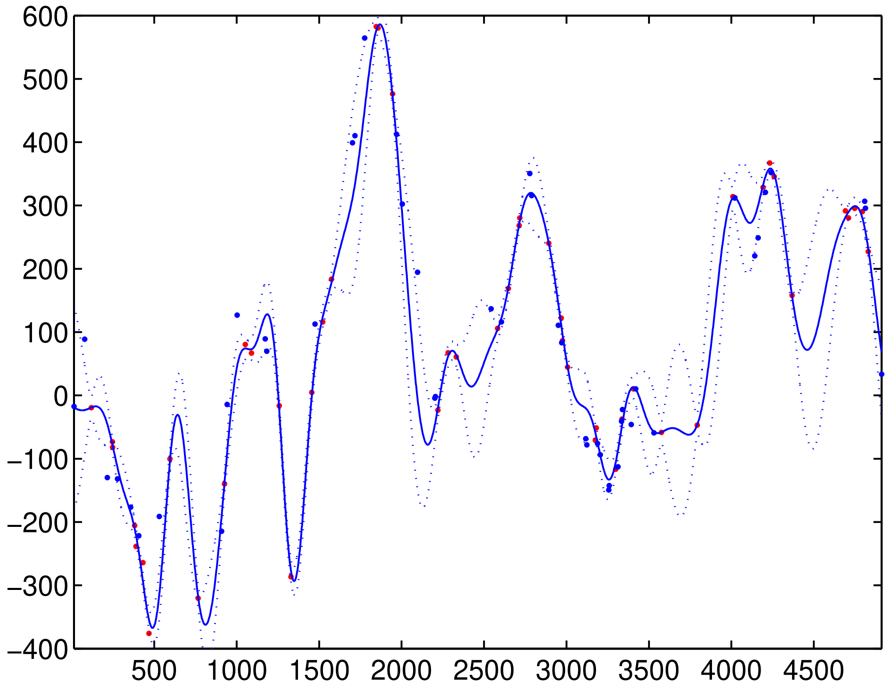
Informatics and Mathematical Modelling
Technical University of Denmark
DK-2800 Lyngby, Denmark
owi@imm.dtu.dk

Inference Group, Cambridge, UK

Gaussian Processes Revisited

Non-parametric
Probabilistic
Bayesian Kernel Method

Supervised learning:
infer $y(x)$ from training set
$\mathcal{D}_N = \{x_n, y_n\}_{n=1,\ldots,N}$.



Linear model approach to supervised regression

$$y(\mathbf{x}) = f_{\mathbf{w}}(\mathbf{x}) + \epsilon = \sum_{h=1}^{H} w_h \phi_h(\mathbf{x}) + \epsilon$$

From Linear Models to Gaussian Processes

Bayesian - prior over $\mathbf{w}$ (we expect simple functions - small $\mathbf{w}$)

$$P(\mathbf{w}) = P(\mathbf{w}|\sigma_w^2) = \mathcal{N}(\mathbf{w}; \mathbf{0}, \mathbf{d}(\sigma_{w,h}^2)) \ .$$

Prior over function

$$\begin{aligned}
\langle f_{\mathbf{w}}(\mathbf{x}_n) \rangle &= \sum_h \langle w_h \rangle \phi_h(\mathbf{x}_n) = 0 \\
\langle f_{\mathbf{w}}(\mathbf{x}_n) f_{\mathbf{w}}(\mathbf{x}_{n'}) \rangle &= \sum_{h,h'} \langle w_h w_{h'} \rangle \phi_h(\mathbf{x}_n) \phi_{h'}(\mathbf{x}_{n'}) = \sum_h \langle w_h^2 \rangle \phi_h(\mathbf{x}_n) \phi_h(\mathbf{x}_{n'}) \\
&= \sum_h \sigma_{w,h}^2 \phi_h(\mathbf{x}_n) \phi_h(\mathbf{x}_{n'})
\end{aligned}$$

We can thus write the prior of the collection of outputs as

$$P(\mathbf{f}) = \mathrm{Normal}(\mathbf{f}; \mathbf{0}, \sigma_w^2 \mathbf{R}\, \mathbf{d}(\sigma_{w,h}^2)\, \mathbf{R}^T)$$

Kernel trick: $\mathbf{R}\, \mathbf{d}(\sigma_{w,h}^2)\, \mathbf{R}^T \to \mathbf{K}$, e.g.

$$K_{ij} = C(\mathbf{x}_i, \mathbf{x}_j) = \exp(-||\mathbf{x}_i - \mathbf{x}_j||^2 / 2r^2)$$

Inference with Gaussian Processes – Regression

Regression with additive noise $\mathcal{N}(\epsilon; 0, \sigma_\epsilon^2)$

$$y(\mathbf{x}) = f(\mathbf{x}) + \epsilon(\mathbf{x})$$

Likelihood

$$P(\mathbf{y}|\mathbf{f}, \mathbf{X}) = P(\mathbf{y}|\mathbf{f}, \mathbf{X}, \sigma_\epsilon^2) = \mathcal{N}(\mathbf{y}; \mathbf{f}, \sigma_\epsilon^2 \mathbf{I}) \ .$$

Predictive distribution –
for $y = y_{N+1}$ at input $\mathbf{x} = \mathbf{x}_{N+1}$ given $\mathbf{t}_N = \{y_n\}_{n=1}^N$

$$P(\mathbf{y}(\mathbf{x})|\mathbf{x}, \mathbf{X}_N, \mathbf{y}_N) = \frac{P(y, \mathbf{y}_N|\mathbf{x}, \mathbf{X}_N)}{P(\mathbf{y}_N|\mathbf{X}_N)} \ .$$

Marginal Likelihood – Evidence

Marginalize over GP:

$$P(\mathbf{y}_N|\mathbf{X}_N) = \int d\mathbf{f}_N P(\mathbf{y}_N|\mathbf{f}_N, \mathbf{X}_N) P(\mathbf{f}_N) = \mathcal{N}(\mathbf{y}_N; \mathbf{m}_N, \mathbf{C}_N)$$

Mean and covariance:

$$\begin{aligned} m_n &= \langle y(\mathbf{x}_n) \rangle = \langle\, [f(\mathbf{x}_n) + \epsilon(\mathbf{x}_n)]\, \rangle = 0 \\ C_{nn'} &= \langle y(\mathbf{x}_n) y(\mathbf{x}_{n'}) \rangle = \langle [f(\mathbf{x}_n) + \epsilon(\mathbf{x}_n)][y(\mathbf{x}_{n'}) + \epsilon(\mathbf{x}_{n'})] \rangle \\ &= \sum_h \sigma^2_{w,h} \phi_h(\mathbf{x}_n) \phi_h(\mathbf{x}_{n'}) + \sigma^2_\epsilon \delta_{nn'} = \sum_h \sigma^2_{w,h} R_{nh} R_{n'h} + \sigma^2_\epsilon \delta_{nn'} \ , \end{aligned}$$

$$\mathbf{C} = \mathbf{K} + \sigma^2_\epsilon \mathbf{I} \ .$$

Predictive Distribution

Bayes theorem: Predictive distribution from ratio of evidences

$$P(\mathbf{y}(\mathbf{x})|\mathbf{x},\mathbf{X}_N,\mathbf{y}_N) = \frac{P(\mathbf{y}_{N+1}|\mathbf{X}_{N+1})}{P(\mathbf{y}_N|\mathbf{X}_N)} = \text{Normal}(y(\mathbf{x}); m(\mathbf{x}), v(\mathbf{x})) \ ,$$

$$\begin{aligned} m(\mathbf{x}) &= \mathbf{k}^T\mathbf{C}_N^{-1}\mathbf{y}_N \\ v(\mathbf{x}) &= \kappa - \mathbf{k}^T\mathbf{C}_N^{-1}\mathbf{k} \\ \mathbf{C}_{N+1} &= \left[\begin{array}{ccc|c} & & & k_1 \\ & \mathbf{C}_N & & \vdots \\ & & & k_N \\ k_1 & \cdots & k_N & \kappa \end{array}\right] . \end{aligned}$$

Prediction at $\mathbf{x}$: $m(\mathbf{x}) \pm \sqrt{v(\mathbf{x})}$

Computational Complexity $\mathcal{O}(N^3)$**.**

Very accurate approximations for classification (Opper&Winther, 2000, Minka 2001).

Inference with A Gaussian Process II – Empirical Bayes

Maximize the marginal Likelihood (i.e. the evidence)

$$\log P(\mathbf{y}_N|\mathbf{X}_N) = -\frac{N}{2}\log 2\pi - \frac{1}{2}\log\det \mathbf{C}_N - \frac{1}{2}\mathbf{t}_N^T\mathbf{C}_N^{-1}\mathbf{t}_N$$

wrt hyperparameters, e.g.

$$\sigma_\epsilon^2$$

$$\begin{aligned} C(\mathbf{x},\mathbf{x}') &= \sum_h \sigma_{w,h}^2 \phi_h(\mathbf{x})\phi_h(\mathbf{x}') \;, \quad h = 1,\ldots \\ C(\mathbf{x},\mathbf{x}') &= c \;\exp\left(-\sum_l^d \frac{||x_l - x_l'||^2}{2r_l^2}\right) \;, \quad l = 1,\ldots,d \end{aligned}$$

Note only one set of hyperparameters governs whole space.

Semi-Supervised Learning

Model the input $\mathbf{x}$ and output y jointly

$$p(y, \mathbf{x}) = p(y|\mathbf{x})p(\mathbf{x})$$

Dataset:

- $N_{y,\mathbf{x}}$ labelled examples: $\mathcal{D}_{y,\mathbf{x}} = \{y_n, \mathbf{x}_n\}_{n=1,\dots,N_{y,\mathbf{x}}}$.
- $N_{\mathbf{x}}$ unlabelled examples: $\mathcal{D}_{\mathbf{x}} = \{\mathbf{x}_n\}_{n=1,\dots,N_{\mathbf{x}}}$.

Model

$$p(y, \mathbf{x}|\theta) = p(y|\mathbf{x}, \theta)p(\mathbf{x}|\theta)$$

Modelling $p(y, \mathbf{x})$ to learn about $p(y|\mathbf{x})$ is not always a good idea!

Using Unlabelled Data in Modelling?

Maximum Likelihood (ML)

$$\begin{aligned}\mathcal{L}(\theta) &= \mathcal{L}_{y|\mathbf{x}}(\theta) + \mathcal{L}_{\mathbf{x}}(\theta) \\ \mathcal{L}_{y|\mathbf{x}}(\theta) &= \sum_{n=1}^{N_{y,\mathbf{x}}} \log p(y_n|\mathbf{x}_n, \theta) \\ \mathcal{L}_{\mathbf{x}}(\theta) &= \sum_{n=1}^{N_{y,\mathbf{x}}+N_{\mathbf{x}}} \log p(\mathbf{x}_n|\theta)\ .\end{aligned}$$

Independent parameters $\theta = (\theta_{y|\mathbf{x}}, \theta_{\mathbf{x}})$: $\mathcal{L}(\theta) = \mathcal{L}_{y|\mathbf{x}}(\theta_{y|\mathbf{x}}) + \mathcal{L}_{\mathbf{x}}(\theta_{\mathbf{x}})$

$$p(y, \mathbf{x}|\theta) = p(y|\mathbf{x}, \theta_{y|\mathbf{x}})p(\mathbf{x}|\theta_{\mathbf{x}})$$

ML solution $\theta^{\mathsf{ML}} = \mathrm{argmin}_\theta\, \mathcal{L}(\theta)$ **factorizes**.

Input-output prediction $p(y|\mathbf{x}, \theta^{\mathsf{ML}}_{y|\mathbf{x}})$ rely on labelled data only.

Model needs **dependent** parameters to learn $p(y|\mathbf{x}, \theta)$ from unlabelled data.

Bayesian Semi-Supervised Learning

Predictive distribution

$$p(y|\mathbf{x}, \mathcal{D}) = \int d\theta \, p(y|\mathbf{x}, \theta) p(\theta|\mathcal{D})$$

for $\mathcal{D} = (\mathcal{D}_{y|\mathbf{x}}, \mathcal{D}_{\mathbf{x}})$.

$$p(\theta|\mathcal{D}) \propto \prod_{n=1}^{N_{y,\mathbf{x}}} p(y_n|\mathbf{x}_n, \theta) \prod_{n=1}^{N_{y,\mathbf{x}}+N_{\mathbf{x}}} p(\mathbf{x}_n|\theta) \, p(\theta)$$

Independent parameters both in prior and Likelihood:

$$p(\theta|\mathcal{D}) = p(\theta_{y|\mathbf{x}}|\mathcal{D}_{y|\mathbf{x}}) p(\theta_{\mathbf{x}}|\mathcal{D}_x)$$

$$p(y|\mathbf{x}, \mathcal{D}) = \int d\theta_{y|\mathbf{x}} d\theta_{\mathbf{x}} \, p(y|\mathbf{x}, \theta_{y|\mathbf{x}}) \, p(\theta_{y|\mathbf{x}}|\mathcal{D}_{y,\mathbf{x}}) p(\theta_{\mathbf{x}}|\mathcal{D}_{\mathbf{x}})$$

Same conclusion as above!

Mixture Modelling

Entangling the parameters $\theta = \left\{ p(k), \theta_{y|x}^{(k)}, \theta_x^{(k)} \right\}_{k=1,\ldots,K}$

$$p(x, y|\theta) = \sum_k p\left(y|x, \theta_{y|x}^{(k)}\right) \, p\left(x|\theta_x^{(k)}\right) \, p(k)$$

ML learning straightforward with EM:

$$\begin{aligned}
p(k|x, y, \theta) &= \frac{p\left(y|x, \theta_{y|x}^{(k)}\right) p\left(x|\theta_x^{(k)}\right) p(k)}{\sum_{k'} p\left(y|x, \theta_{y|x}^{(k')}\right) p\left(x|\theta_x^{(k')}\right) p(k')} \\
p(k) &= \frac{1}{N} \sum_n p(k|x_n, y_n, \theta) \\
\ldots &= \ldots
\end{aligned}$$

Examples without labels use $p(k|x_n, \theta)$ instead.

Variational Bayes better! Use GP to model $p(y|x, \theta)$.

Motivation and Open Questions

- More flexible modelling –
  let hyperparameters be mix. comp. dependent.

- Useful if noise-level and smoothness change with $\mathbf{x}$.

- Semi-supervised learning.

- Computational complexity Gaussian processes $\mathcal{O}(N^3)$.

- Using K mix. comp. can we get it down to $\mathcal{O}(K(N/K)^3)$?

Joint Gaussian Process-Density Mixture

GP model for each mixture component

$$
\begin{aligned}
p(\mathbf{f}|\{K_k\}) = \prod_k p(\mathbf{f}_k|K_k) &= \prod_k \mathcal{N}(\mathbf{f}_k; 0, K_k) \\
\mathbf{y}(\mathbf{x}) &= \mathbf{f}_k(\mathbf{x}) + \epsilon_k \;, \qquad k = 1, \ldots, K
\end{aligned}
$$

The probability of everything:

Probability(observations $\mathbf{X}, \mathbf{y}$, GP and other parameters):

$$
\begin{aligned}
p(\mathbf{X}, \mathbf{y}, \mathbf{f}, \theta) &= \prod_n \left\{ \sum_{k=1}^{K} p\left(y_n | f_k(\mathbf{x}_n), \theta_{y|x}^{(k)}\right) p\left(\mathbf{x}_n | \theta_x^{(k)}\right) p(k) \right\} \\
&\quad \prod_k p(\mathbf{f}_k|K_k)\, p(\theta)
\end{aligned}
$$

Variational inference possible for Gaussian Processes and some other parameters ϵ_ϵ^2 and $\theta_{\mathbf{x}}$.

Related Work

- Mixture of Experts, (Jacobs, Jordan, Nowlan & Hinton); Variational Bayes Hierarchical Mixtures of Experts, (Steve Waterhouse), Infinite Mixtures of Gaussian Process Experts, (Rasmussen & Ghahramani, NIPS 2000)

$$p(\mathbf{y}|\mathbf{X},\theta) = \sum_{\mathbf{Z}} p(\mathbf{y}|\mathbf{X},\mathbf{Z},\theta_{y|x})p(\mathbf{Z}|\mathbf{X},\theta_x) \ ,$$

  $\mathbf{Z} = (\mathbf{z}_1, \ldots, \mathbf{z}_N)$ assignments and $p(\mathbf{Z}|\mathbf{X},\theta_x)$ gating network.

- Efficient Kernel Methods using Fast Improved Gauss Transform, (Yang, Duraiswami and Davis, NIPS 2004).

- Bayesian Committee Machine (Tresp & Schwaighofer).

[Thanks to David MacKay and Sam Roweis].

Suboptimal Two-Stage Approximation

1. Cluster (e.g. with Variational Bayes MoG):

$$p(k|\mathbf{x}, \mathcal{D}_{\mathbf{x}}) = \frac{p\,(x|k, \mathcal{D}_{\mathbf{x}})\,p\,(k|\mathcal{D}_{\mathbf{x}})}{\sum_{k'} p\,(x|k', \mathcal{D}_{\mathbf{x}})\,p\,(k'|\mathcal{D}_{\mathbf{x}})}$$

2. Assign examples to mixtures and train GPs individually

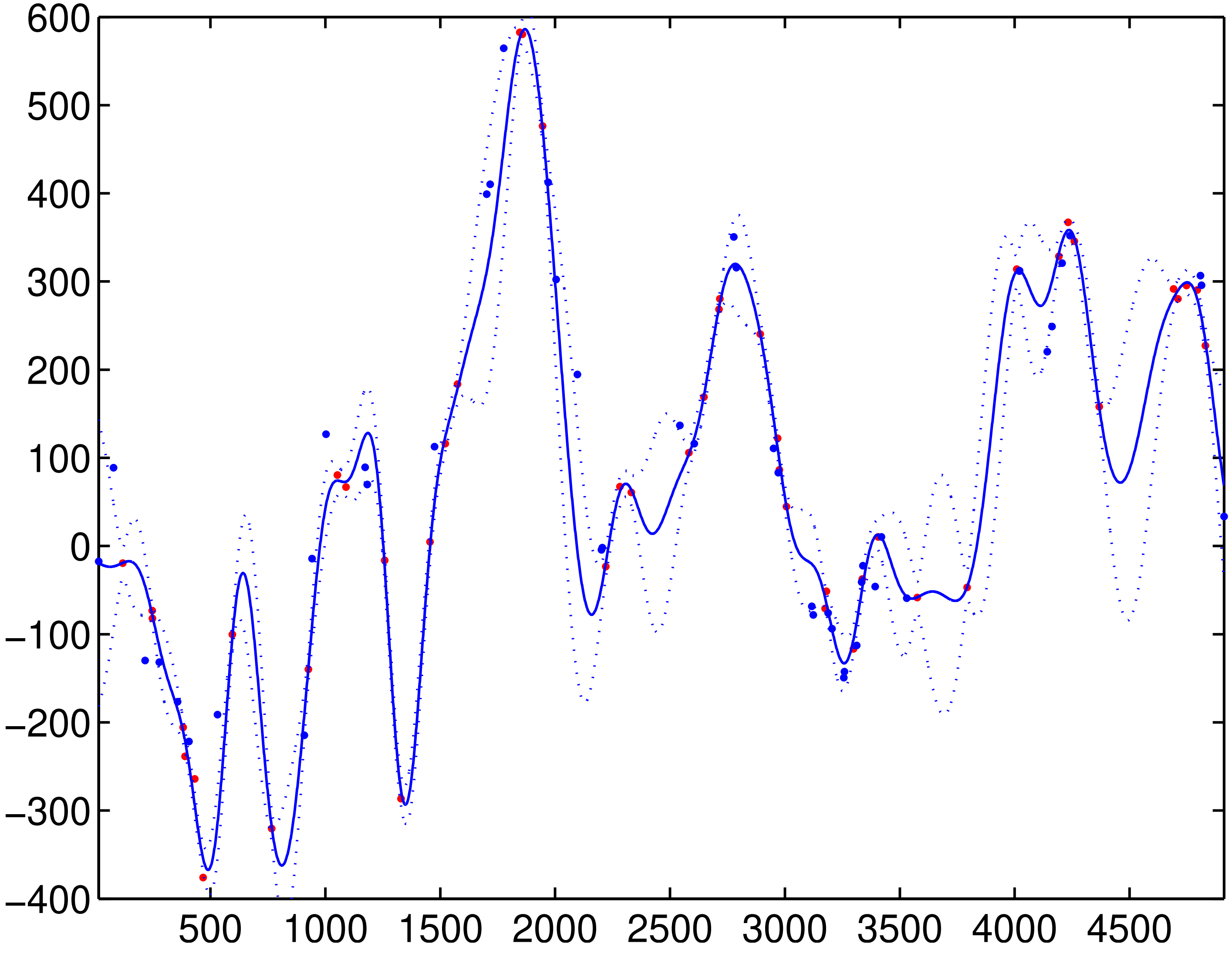
$$\mathcal{D}_{y|\mathbf{x}}(k) = \{(y_n, \mathbf{x}_n)\,|n = 1, \ldots, N_{y,\mathbf{x}}, p(k|\mathbf{x}_n, \mathcal{D}_{\mathbf{x}}) \geq \delta\} \ .$$

Overlapping training sets, e.g. $\delta = 0.05$.

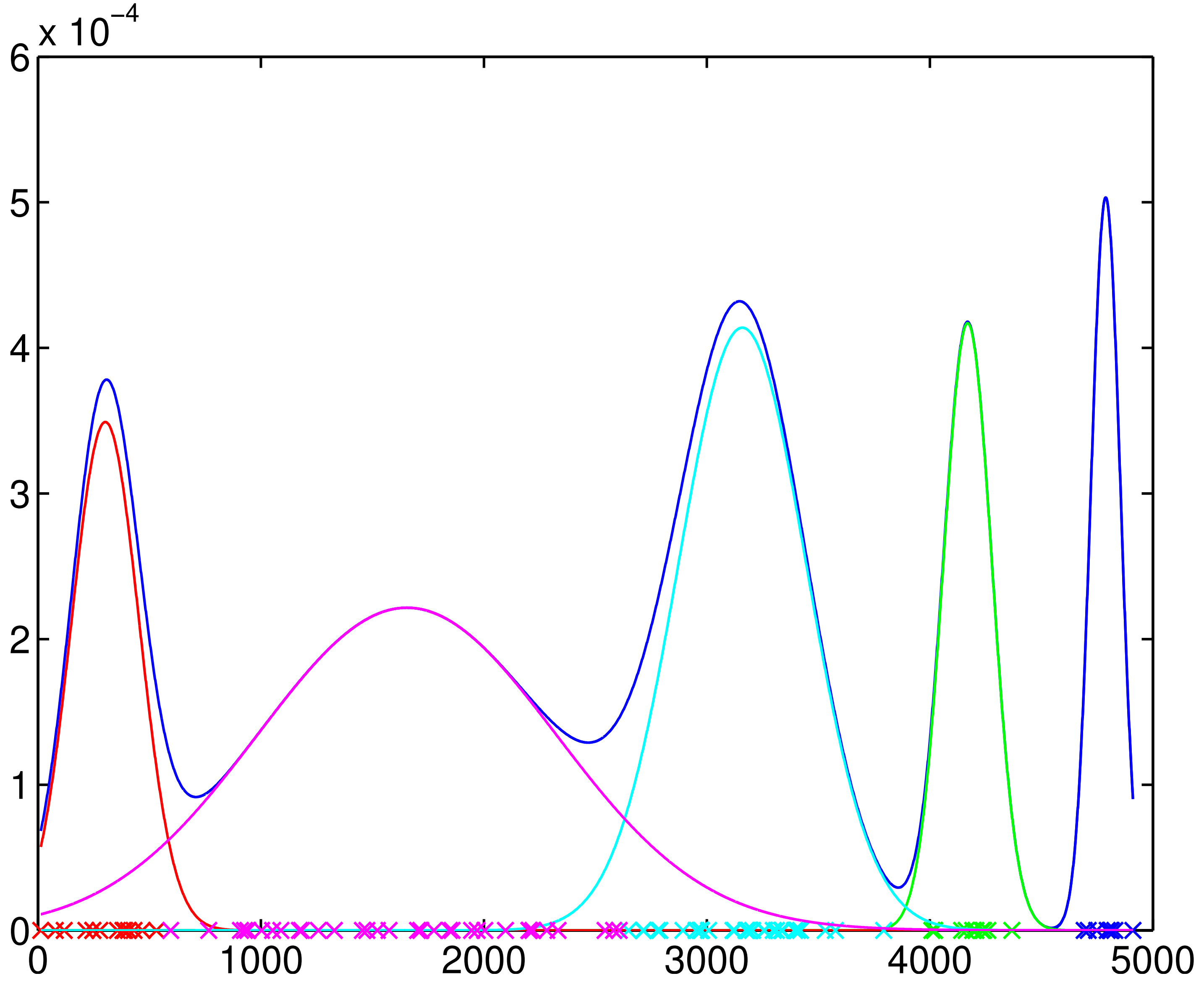
Prediction – weighted average:

$$\begin{aligned}
m(\mathbf{x}) &= \sum_k m_k(\mathbf{x}) p(k|\mathbf{x}, \mathcal{D}_{\mathbf{x}}) \\
v(\mathbf{x}) &= \sum_k \left\langle (\, y_k(\mathbf{x}) - m(\mathbf{x})\,)^2 \right\rangle p(k|\mathbf{x}, \mathcal{D}_{\mathbf{x}}) \\
&= \sum_k \left(v_k(\mathbf{x}) + (\, m_k(\mathbf{x}) - m(\mathbf{x})\,)^2 \right) p(k|\mathbf{x}, \mathcal{D}_{\mathbf{x}})
\end{aligned}$$

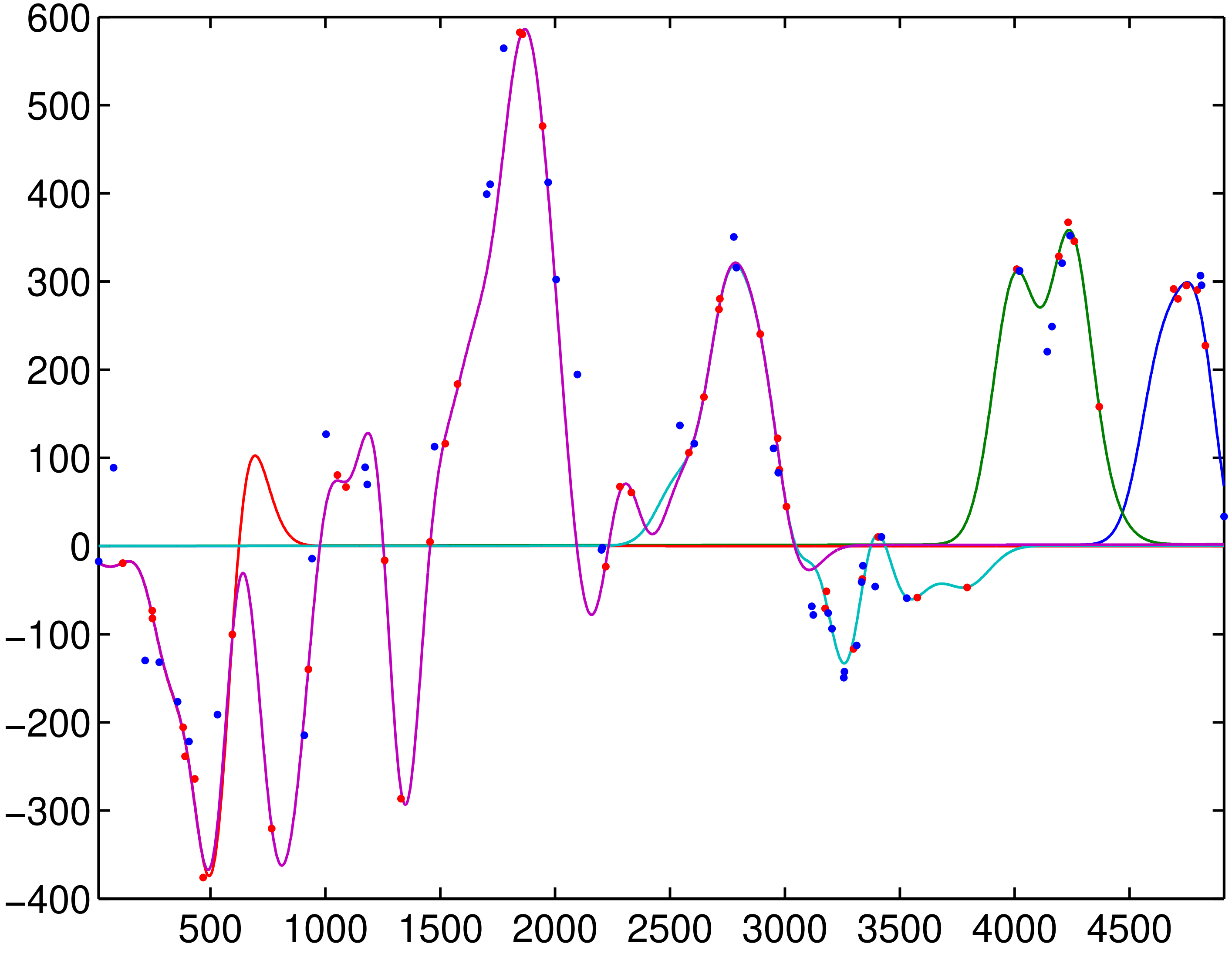
Empirical Bayes Gaussian covariance + linear covariance function



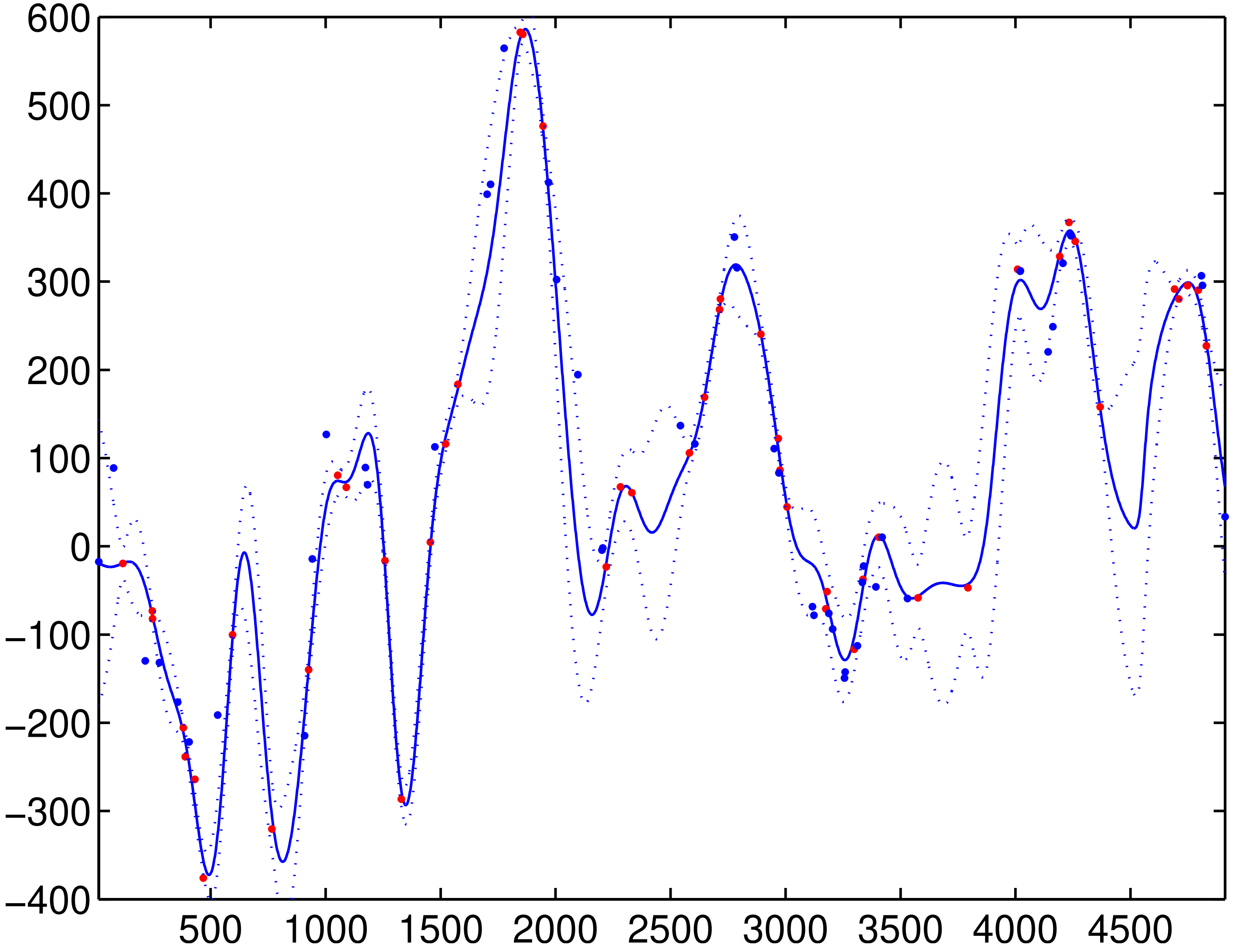
Clustering both labelled and unlabelled examples with Variational Bayes mixture of Gaussians.



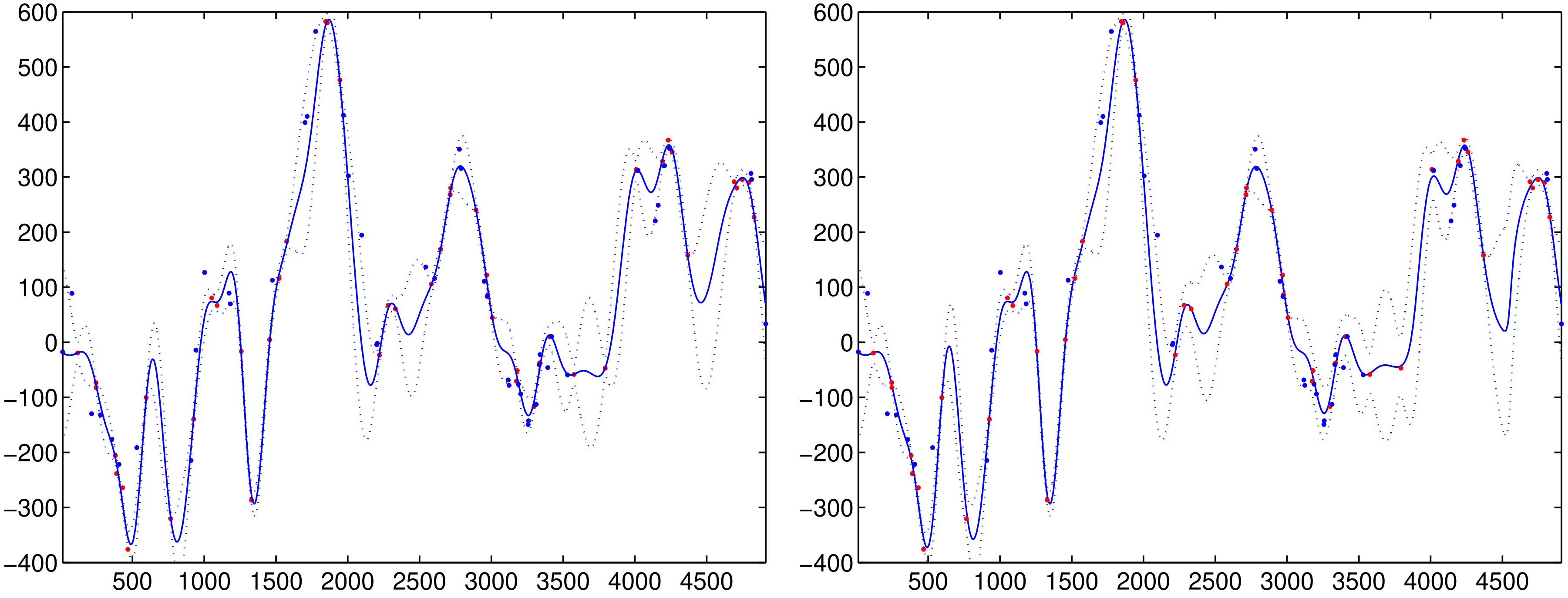
The prediction of each individual member Another approximation did not use predictive distribution, but rather ML.



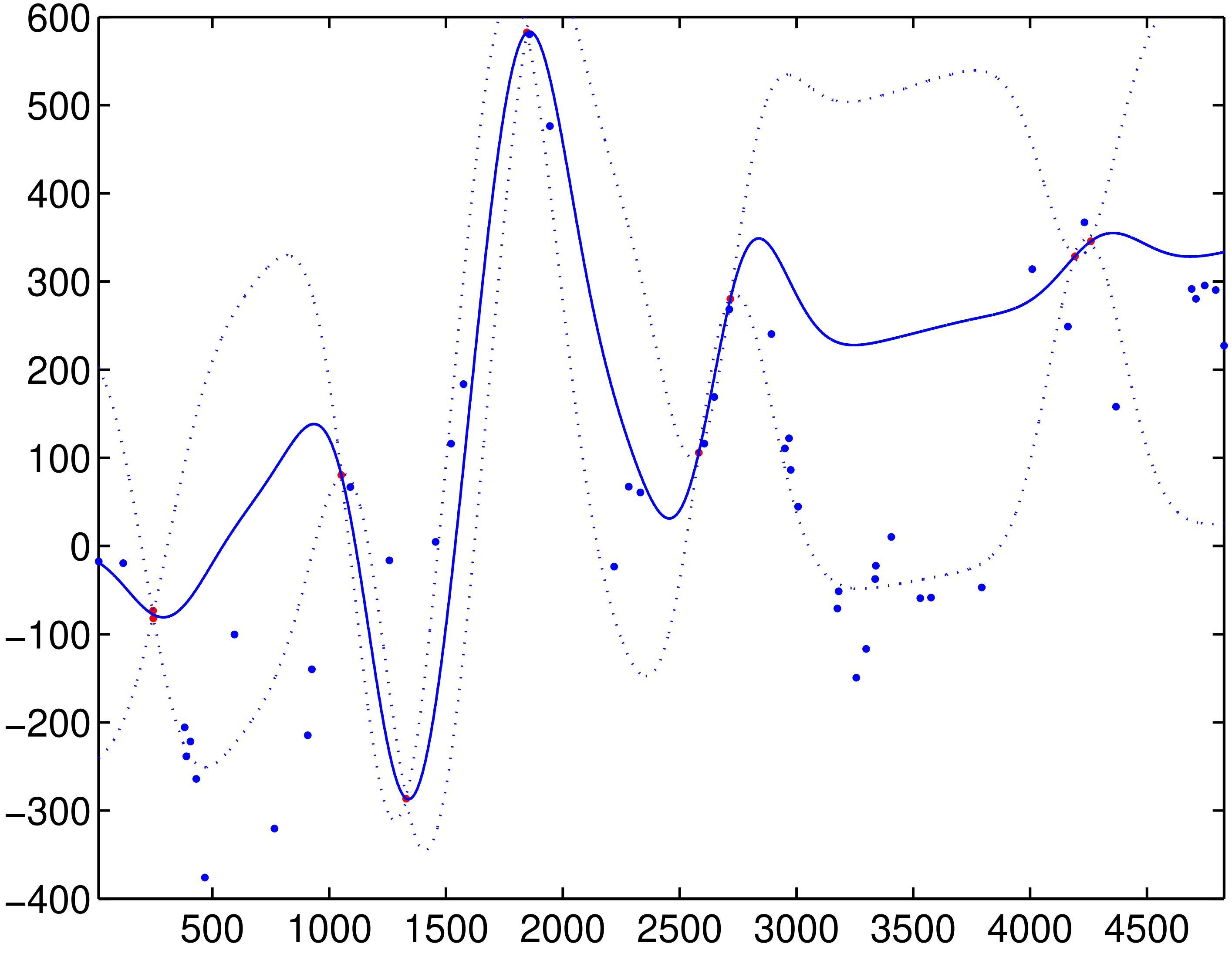
The averaged prediction



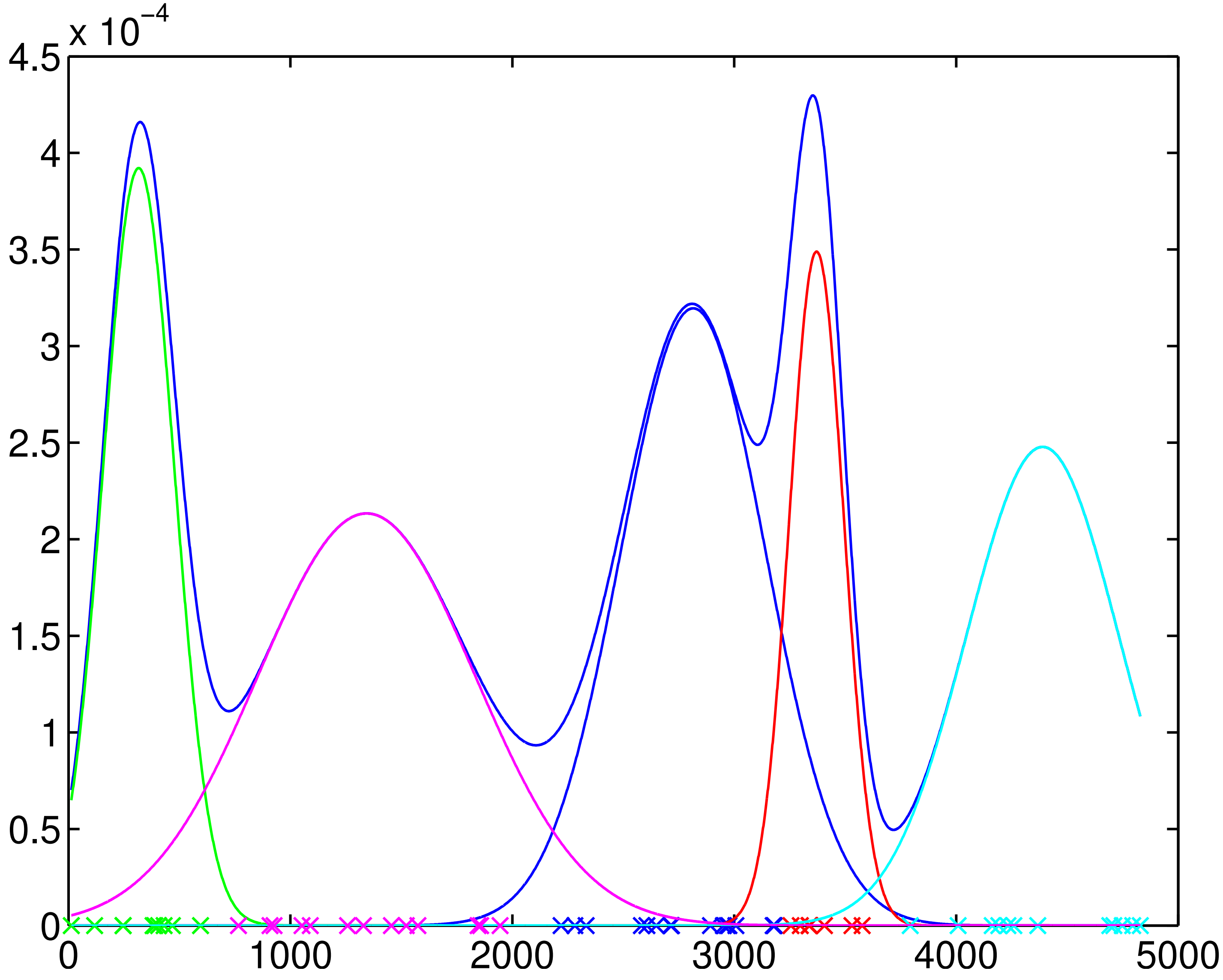
Comparing Classic and Mixture



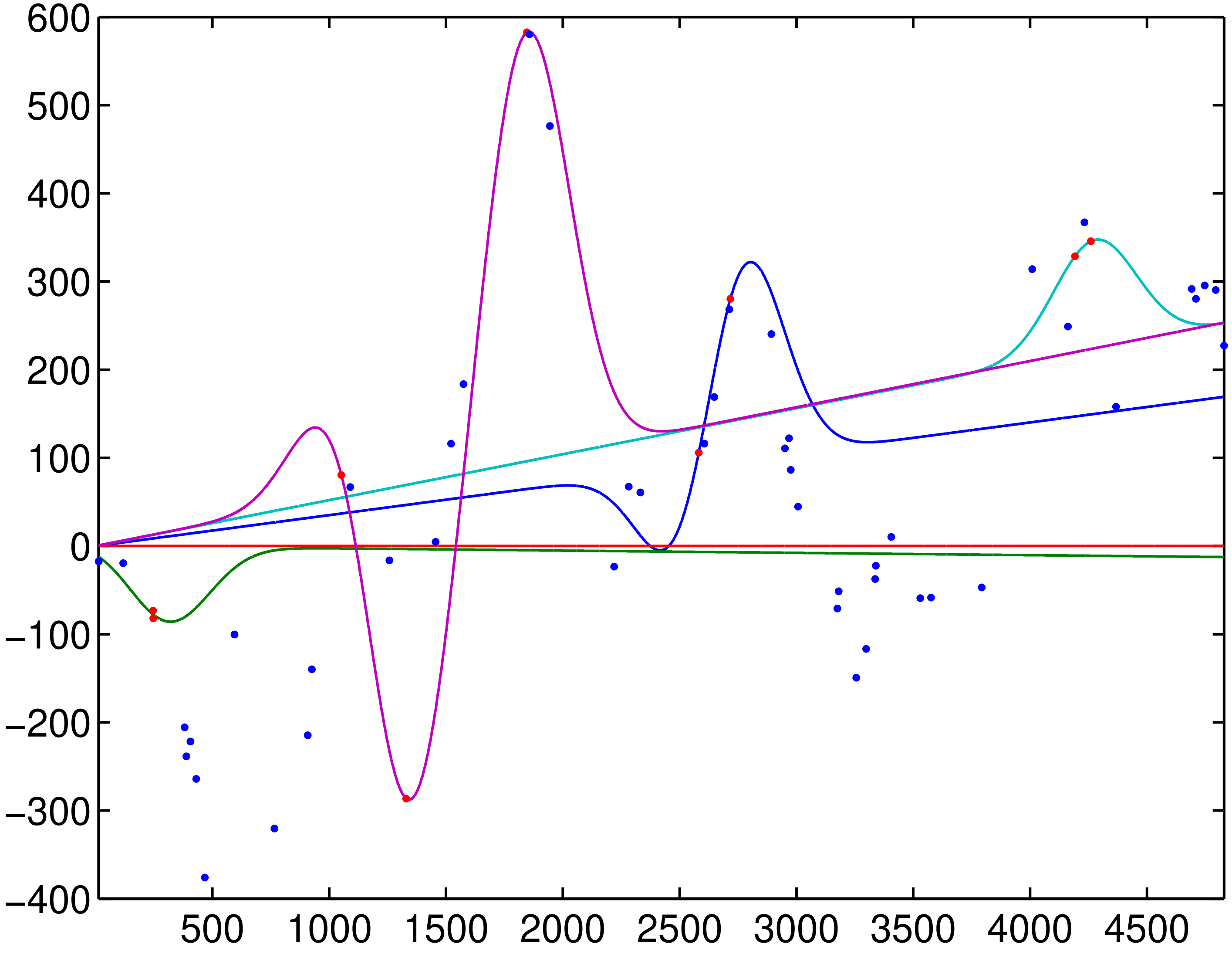
Smaller data set Empirical Bayes Gaussian covariance + linear covariance function



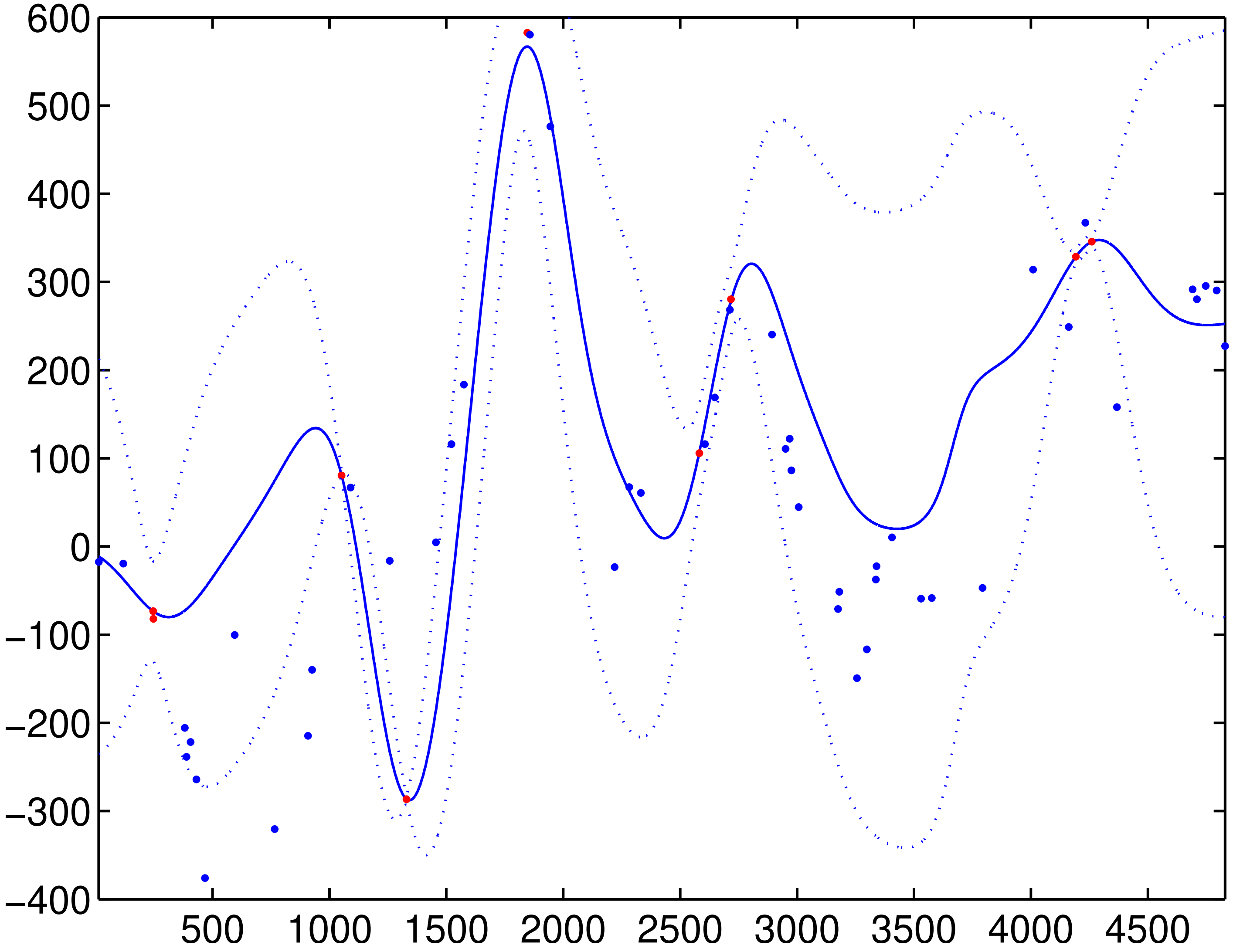
Clustering both labelled and unlabelled examples with Variational Bayes mixture of Gaussians.



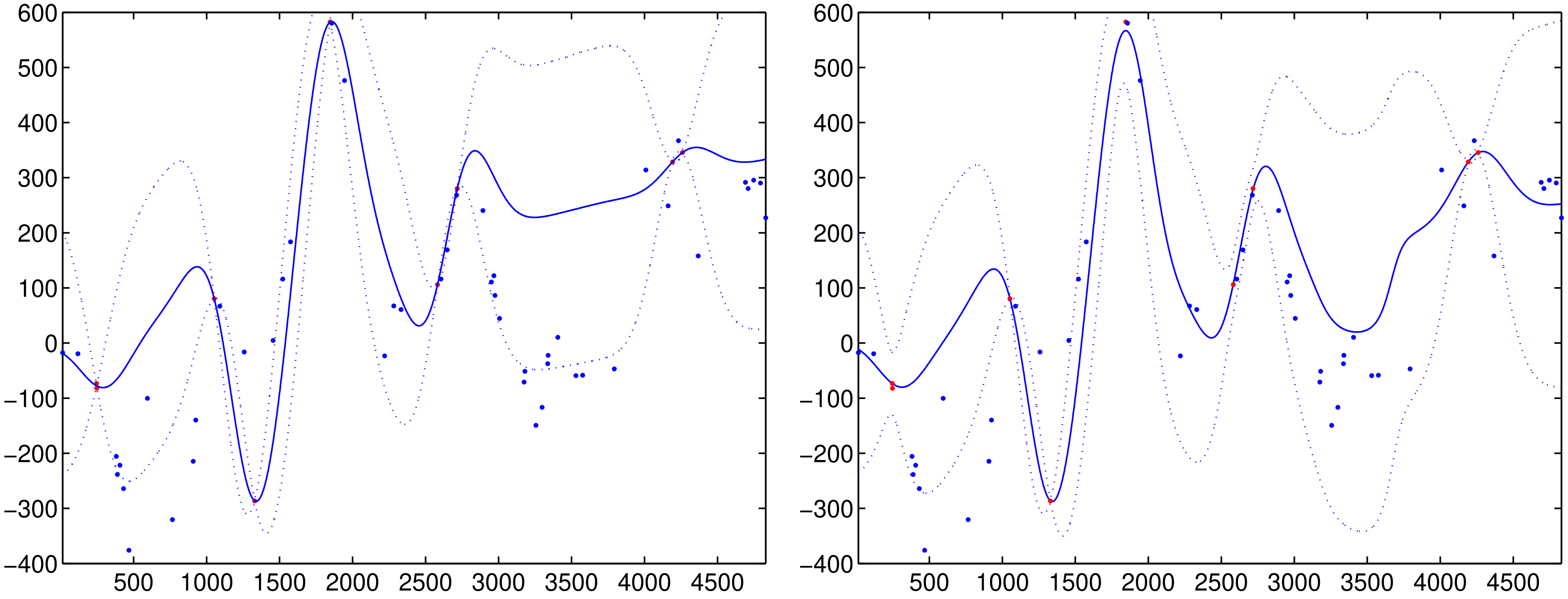
The prediction of each individual member Another approximation did not use predictive distribution, but rather ML.



The averaged prediction



Comparing Classic and Mixture



Variational Approach to GP-Density Model

Introduce latent assignment variables $\mathbf{z}_n$: $z_{nk} = \{0, 1\}$, $\sum_k z_{nk} = 1$,

$$\langle z_{nk} \rangle = p(k|y_n, \mathbf{x}_n, \mathcal{D}) \ .$$

Probability of observations, GPs and $\mathbf{z}_n$:

$$p(\mathbf{X}, \mathbf{y}, \mathbf{Z}, \mathbf{f}|\theta) \;=\; \prod_n \prod_k \left\{ p\left(y_n | f_k(\mathbf{x}_n), \theta_{y|x}^{(k)} \right) p\left(\mathbf{x}_n | \theta_x^{(k)} \right) p(k) \right\}^{z_{nk}}$$
$$\prod_k p(\mathbf{f}_k | K_k)$$

For simplicity ML in all other variables.

Variational distribution (regression)

$$q(\mathbf{Z}, \mathbf{f}) = \prod_k q(\mathbf{f}_k) \prod_n q(\mathbf{z}_n)$$

Variational Solution

Minimize KL-divergence $\mathrm{KL}(q||p) = \int d\mathbf{f} d\mathbf{Z}\, q(\mathbf{Z}, \mathbf{f}) \ln \frac{p(\mathbf{Z},\mathbf{f}|\mathbf{X},\mathbf{y},\theta)}{q(\mathbf{Z},\mathbf{f})}$:

$$q(\mathbf{f}_k) \;\propto\; p(\mathbf{f}_k|K_k) \prod_n \left[p\left(y_n | f_k(\mathbf{x}_n), \theta^{(k)}_{y|x} \right) \right]^{\langle z_{nk} \rangle_{q(\mathbf{z}_n)}}$$

Examples that have $\langle z_{nk} \rangle = 0$ can be marginalized out!

$$\langle z_{nk} \rangle_{q(\mathbf{z}_n)} \;=\; \frac{e^{\left\langle \ln p\left(y_n | f_k(\mathbf{x}_n), \theta^{(k)}_{y|x} \right) \right\rangle_{q(\mathbf{f}_k)}} p\left(\mathbf{x}_n | \theta^{(k)} \right) p(k)}{\sum_{k'} e^{\left\langle \ln p\left(y_n | f_{k'}(\mathbf{x}_n), \theta^{(k')}_{y|x} \right) \right\rangle_{q(\mathbf{f}_{k'})}} p\left(\mathbf{x}_n | \theta^{(k')} \right) p(k')} .$$

Regression:

$$\begin{aligned} \left\langle \ln p\left(y_n | f_k(\mathbf{x}_n), \theta^{(k)}_{y|x} \right) \right\rangle &= -\frac{d}{2} \ln 2\pi\sigma_\epsilon^2 - \frac{1}{2\sigma_\epsilon^2} \langle (y_n - f_k(\mathbf{x}_n))^2 \rangle \\ &= -\frac{d}{2} \ln 2\pi\sigma_\epsilon^2 - \frac{1}{2\sigma_\epsilon^2} \left((y_n - m_k(\mathbf{x}_n))^2 + v_k(\mathbf{x}) \right) \end{aligned}$$

Classification

Likelihood for label $y = \pm 1$:

$$p(y|f(\mathbf{x}), \sigma) = \Phi(yf(\mathbf{x})/\sigma) \to \Theta(yf(\mathbf{x})) \quad \text{for} \quad \sigma \to 0 \ .$$

Solution: $\langle z_{nk} \rangle = 0$ goes out.

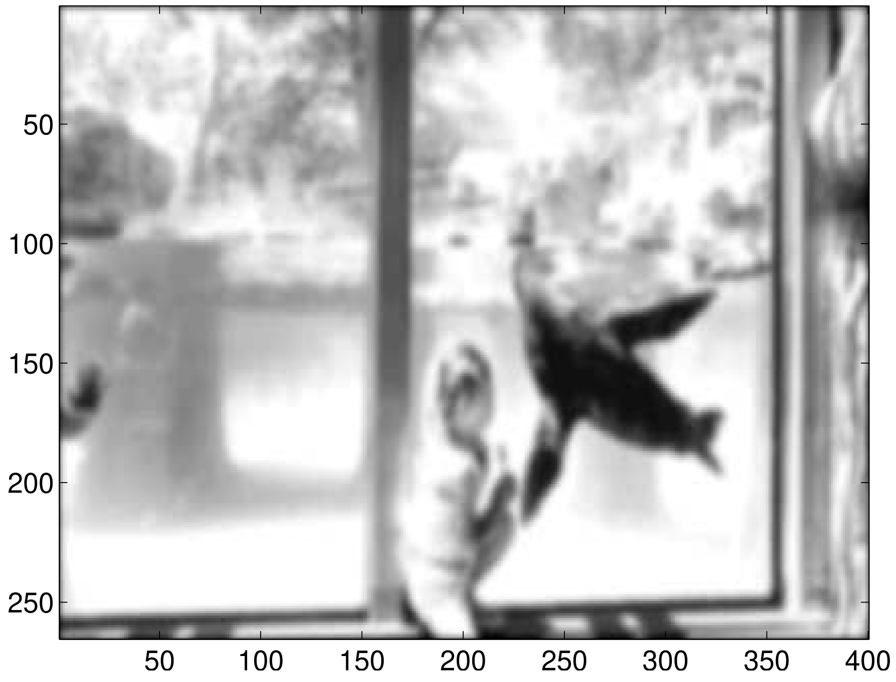
Variational distribution: $q(\mathbf{Z}, \mathbf{f}) = \prod_n \left[\prod_k q(f_k(\mathbf{x}_n)) q(\mathbf{z}_n)\right]$

$$q(f_k(\mathbf{x}_n)) \propto \exp\left(-\frac{1}{2}\lambda_{nk} f_k^2(\mathbf{x}_n) + \gamma_{nk} f_k(\mathbf{x}_n)\right) p^{\langle z_{nk} \rangle}(y_n | f_k(\mathbf{x}_n), \sigma) \ .$$

Speculation: should work well for digit recognition, data clustered with different characteristics.

Image Deconvolution

Image $\mathbf{f}$



N is the number of pixels now.

Observation blurred noise image:

$$\mathbf{y} = \mathbf{G}\mathbf{f} + \epsilon$$

Prior knowledge: $\mathbf{f} \in [0;255]^N$

$\mathbf{G}$, follows from the camera point spread function, probably approximately known from visual inspection, varying across image.

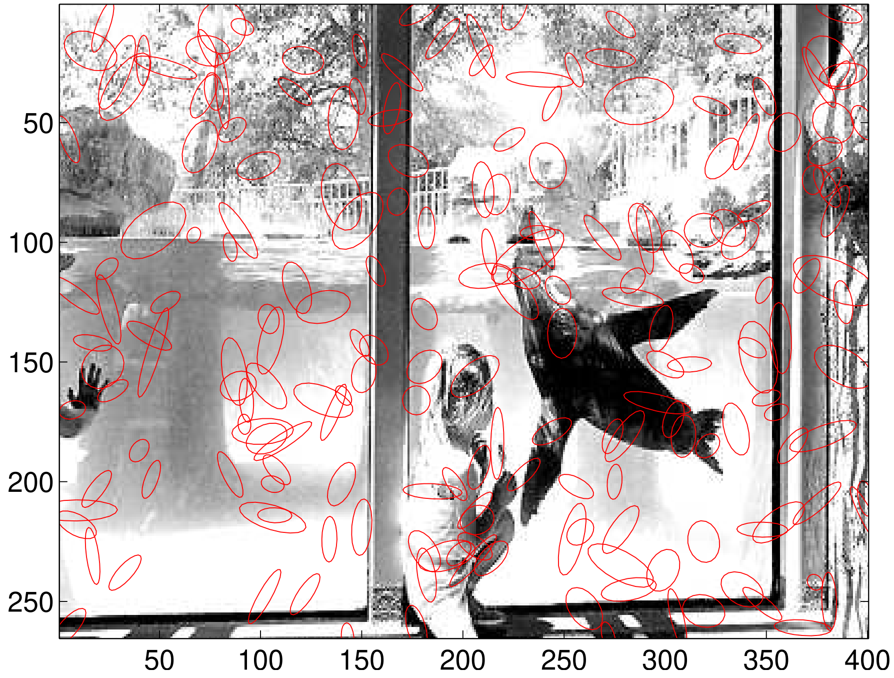
Within patch values varies smoothly and are almost constant, smoothness parameter known!

GP-Density Mixture Deconvolution

Infer image:

$$\langle \mathbf{f} \rangle \pm \sqrt{\mathrm{var}(\mathbf{f})} \ .$$

GP-density mixture model for image



$$p(\mathbf{X}, \mathbf{y}, \mathbf{Z}, \mathbf{f}|\theta) \propto \prod_n \prod_k \left\{ p\left(y_n | [\mathbf{G}\mathbf{f}_k]_n, \theta_{y|x}^{(k)} \right) p\left(\mathbf{x}_n | \theta_x^{(k)} \right) p(k) \right\}^{z_{nk}}$$
$$\prod_k P(\mathbf{f}_k | K_k) \prod_{k,n} 1[0 \leq f_k(\mathbf{x}_n) \leq 255]$$

Basically a classification model – density modelling to localize patches

Outlook

Joint Gaussian Process-Density Mixtures to address

- Flexbility
- Semi-supervised learning
- Complexity

Approximate two-stage approximation presented.

Variational Inference, EC and EP possible for regression and classification

Better models by joint estimation?! Or will inherent computational problems of density–mixture modelling spoil the day?