Variational Model Selection for Sparse Gaussian Process Regression

Michalis K. Titsias
School of Computer Science
University of Manchester

7 September 2008

Outline

- Gaussian process regression and sparse methods
- Variational inference based on inducing variables
  - Auxiliary inducing variables
  - The variational bound
  - Comparison with the PP/DTC and SPGP/FITC marginal likelihood
  - Experiments in large datasets
  - Inducing variables selected from training data
- Variational reformulation of SD, FITC and PITC
- Related work/Conclusions

Gaussian process regression

Regression with Gaussian noise

- Data: $\{(\mathbf{x}_i, y_i), i = 1, \ldots, n\}$ where $\mathbf{x}_i$ is a vector and y_i scalar
- Likelihood:

$$y_i = f(\mathbf{x}_i) + \epsilon, \quad \epsilon \sim N(0, \sigma^2)$$

$$p(\mathbf{y}|\mathbf{f}) = N(\mathbf{y}|\mathbf{f}, \sigma^2 I), \quad f_i = f(\mathbf{x}_i)$$

- GP prior on $\mathbf{f}$:

$$p(\mathbf{f}) = N(\mathbf{f}|\mathbf{0}, K_{nn})$$

  K_{nn} is the $n \times n$ covariance matrix on the training data computed using a kernel that depends on $\boldsymbol{\theta}$

- Hyperparameters: $(\sigma^2, \boldsymbol{\theta})$

Gaussian process regression

Maximum likelihood II inference and learning

- Prediction: Assume hyperparameters $(\sigma^2, \boldsymbol{\theta})$ are known
  - Infer the latent values $\mathbf{f}_*$ at test inputs X_*:

$$p(\mathbf{f}_*|\mathbf{y}) = \int_{\mathbf{f}} p(\mathbf{f}_*|\mathbf{f})p(\mathbf{f}|\mathbf{y})d\mathbf{f}$$

    $p(\mathbf{f}_*|\mathbf{f})$ test conditional, $p(\mathbf{f}|\mathbf{y})$ posterior on training latent values

- Learning $(\sigma^2, \boldsymbol{\theta})$: Maximize the marginal likelihood

$$p(\mathbf{y}) = \int_{\mathbf{f}} p(\mathbf{y}|\mathbf{f})p(\mathbf{f})d\mathbf{f} = N(\mathbf{y}|\mathbf{0}, \sigma^2 I + K_{nn})$$

Time complexity is $O(n^3)$

Sparse GP regression

Time complexity is $O(n^3)$: Intractability for large datasets

- Exact prediction and training is intractable
  - We can neither compute the predictive distribution $p(\mathbf{f}_*|\mathbf{y})$ nor the marginal likelihood $p(\mathbf{y})$
- Approximate/sparse methods:
  - Subset of data: Keep only m training points, complexity is $O(m^3)$
  - Inducing/active/support variables: Complexity $O(nm^2)$
  - Other methods: Iterative methods for linear systems

Sparse GP regression using inducing variables

Inducing variables

- Subset of training points (Csato and Opper, 2002; Seeger et al. 2003, Smola and Bartlett, 2001)
- Test points (BCM; Tresp, 2000)
- Auxiliary variables (Snelson and Ghahramani, 2006; Quiñonero-Candela and Rasmussen, 2005)

Training the sparse GP regression system

- Select inducing inputs
- Select hyperparameters $(\sigma^2, \boldsymbol{\theta})$
- Which objective function is going to do all that?
  - The approximate marginal likelihood
  - But which approximate marginal likelihood?

Sparse GP regression using inducing variables

Approximate marginal likelihoods currently used are derived

- by changing/approximating the likelihood $p(\mathbf{y}|\mathbf{f})$
- by changing/approximating the prior $p(\mathbf{f})$ (Quiñonero-Candela and Rasmussen, 2005)
- all have the form

$$F_P = N(\mathbf{y}|\mathbf{0}, \widetilde{K})$$

where $\widetilde{K}$ is some approximation to the true covariance $\sigma^2 I + K_{nn}$

Overfitting can often occur

- The approximate marginal likelihood is not a lower bound
- Joint learning of the inducing points and hyperparameters easily leads to overfitting

Sparse GP regression using inducing variables

What we wish to do here

- Do model selection in a different way
  - Never think about approximating the likelihood $p(\mathbf{y}|\mathbf{f})$ or the prior $p(\mathbf{f})$
  - Apply standard variational inference
    - Just introduce a variational distribution to approximate the true posterior
  - That will give us a lower bound
- We will propose the bound for model selection
  - jointly handle inducing inputs and hyperparameters

Auxiliary inducing variables (Snelson and Ghahramani, 2006)

- Auxiliary inducing variables: m latent function values $\mathbf{f}_m$ associated with arbitrary inputs X_m
- Model augmentation: We augment the GP prior $p(\mathbf{f}, \mathbf{f}_m) = p(\mathbf{f}|\mathbf{f}_m)p(\mathbf{f}_m)$

$$\textbf{joint} \qquad p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{f}_m)p(\mathbf{f}_m)$$

$$\textbf{marginal likelihood} \qquad \int_{\mathbf{f},\mathbf{f}_m} p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{f}_m)p(\mathbf{f}_m)d\mathbf{f}d\mathbf{f}_m$$

- The model is unchanged! The predictive distribution and the marginal likelihood are the same

The parameters X_m play no active role (at the moment)...and there is no any fear about overfitting when we specify X_m

Auxiliary inducing variables

What we wish: *To use the auxiliary variables* $(\mathbf{f}_m, X_m)$ *to facilitate inference about the training function values* $\mathbf{f}$

- Before we get there: Let's specify the ideal inducing variables
- **Definition**: We call $(\mathbf{f}_m, X_m)$ *optimal* when $\mathbf{y}$ and $\mathbf{f}$ are conditionally independent given $\mathbf{f}_m$

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

- At optimality: The augmented true posterior $p(\mathbf{f}, \mathbf{f}_m|\mathbf{y})$ factorizes as

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

Auxiliary inducing variables

What we wish: *To use the auxiliary variables* $(\mathbf{f}_m, X_m)$ *to facilitate inference about the training function values* $\mathbf{f}$

- Question: How can we discover optimal inducing variables?
- Answer: Minimize a distance between the true $p(\mathbf{f}, \mathbf{f}_m|\mathbf{y})$ and an approximate $q(\mathbf{f}, \mathbf{f}_m)$ wrt to X_m and (optionally) the number m
- The key: $q(\mathbf{f}, \mathbf{f}_m)$ must satisfy the factorization that holds for optimal inducing variables:

$$\text{True} \qquad p(\mathbf{f}, \mathbf{f}_m|\mathbf{y}) = p(\mathbf{f}|\mathbf{f}_m, \mathbf{y})p(\mathbf{f}_m|\mathbf{y})$$

$$\text{Approximate} \quad q(\mathbf{f}, \mathbf{f}_m) = p(\mathbf{f}|\mathbf{f}_m)\phi(\mathbf{f}_m)$$

Variational learning of inducing variables

- Variational distribution:

$$q(\mathbf{f}, \mathbf{f}_m) = p(\mathbf{f}|\mathbf{f}_m)\phi(\mathbf{f}_m)$$

  $\phi(\mathbf{f}_m)$ is an unconstrained variational distribution over $\mathbf{f}_m$
- Standard variational inference: We minimize the divergence $\mathrm{KL}(q(\mathbf{f}, \mathbf{f}_m)||p(\mathbf{f}, \mathbf{f}_m|\mathbf{y}))$
- Equivalently we maximize a bound on the true log marginal likelihood:

$$F_V(X_m, \phi(\mathbf{f}_m)) = \int_{\mathbf{f},\mathbf{f}_m} q(\mathbf{f}, \mathbf{f}_m) \log \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{f}_m)p(\mathbf{f}_m)}{q(\mathbf{f}, \mathbf{f}_m)} d\mathbf{f}\, d\mathbf{f}_m$$

Let's compute this

Computation of the variational bound

$$
\begin{aligned}
F_V(X_m, \phi(\mathbf{f}_m)) &= \int_{\mathbf{f},\mathbf{f}_m} p(\mathbf{f}|\mathbf{f}_m)\phi(\mathbf{f}_m) \log \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{f}_m)p(\mathbf{f}_m)}{p(\mathbf{f}|\mathbf{f}_m)\phi(\mathbf{f}_m)} d\mathbf{f} d\mathbf{f}_m \\
&= \int_{\mathbf{f},\mathbf{f}_m} p(\mathbf{f}|\mathbf{f}_m)\phi(\mathbf{f}_m) \log \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f}_m)}{\phi(\mathbf{f}_m)} d\mathbf{f} d\mathbf{f}_m \\
&= \int_{\mathbf{f}_m} \phi(\mathbf{f}_m) \left\{ \int_{\mathbf{f}} p(\mathbf{f}|\mathbf{f}_m) \log p(\mathbf{y}|\mathbf{f}) d\mathbf{f} + \log \frac{p(\mathbf{f}_m)}{\phi(\mathbf{f}_m)} \right\} d\mathbf{f}_m \\
&= \int_{\mathbf{f}_m} \phi(\mathbf{f}_m) \left\{ \log G(\mathbf{f}_m, \mathbf{y}) + \log \frac{p(\mathbf{f}_m)}{\phi(\mathbf{f}_m)} \right\} d\mathbf{f}_m
\end{aligned}
$$

$$
\log G(\mathbf{f}_m, \mathbf{y}) = \log \left[N(\mathbf{y}|E[\mathbf{f}|\mathbf{f}_m], \sigma^2 I) \right] - \frac{1}{2\sigma^2} Tr[\mathrm{Cov}(\mathbf{f}|\mathbf{f}_m)]
$$

$$
E[\mathbf{f}|\mathbf{f}_m] = K_{nm}K_{mm}^{-1}\mathbf{f}_m, \; \mathrm{Cov}(\mathbf{f}|\mathbf{f}_m) = K_{nn} - K_{nm}K_{mm}^{-1}K_{mn}
$$

Computation of the variational bound

- Merge the logs

$$F_V(X_m, \phi(\mathbf{f}_m)) = \int_{\mathbf{f}_m} \phi(\mathbf{f}_m) \left\{ \log \frac{G(\mathbf{f}_m, \mathbf{y}) p(\mathbf{f}_m)}{\phi(\mathbf{f}_m)} \right\} d\mathbf{f}_m$$

- Reverse Jensen's inequality to maximize wrt $\phi(\mathbf{f}_m)$:

$$\begin{aligned} F_V(X_m) &= \log \int_{\mathbf{f}_m} G(\mathbf{f}_m, \mathbf{y}) p(\mathbf{f}_m) d\mathbf{f}_m \\ &= \log \int_{\mathbf{f}_m} N(\mathbf{y}|\boldsymbol{\alpha}_m, \sigma^2 I) p(\mathbf{f}_m) d\mathbf{f}_m - \frac{1}{2\sigma^2} Tr[Cov(\mathbf{f}|\mathbf{f}_m)] \\ &= \log \left[N(\mathbf{y}|\mathbf{0}, \sigma^2 I + K_{nm} K_{mm}^{-1} K_{mn}) \right] - \frac{1}{2\sigma^2} Tr[Cov(\mathbf{f}|\mathbf{f}_m)] \end{aligned}$$

where $Cov(\mathbf{f}|\mathbf{f}_m) = K_{nn} - K_{nm} K_{mm}^{-1} K_{mn}$

Variational bound versus PP log likelihood

- The traditional projected process (PP or DTC) log likelihood is

$$F_P = \log \left[N(\mathbf{y}|\mathbf{0}, \sigma^2 I + K_{nm} K_{mm}^{-1} K_{mn}) \right]$$

- What we obtained is

$$F_V = \log \left[N(\mathbf{y}|\mathbf{0}, \sigma^2 I + K_{nm} K_{mm}^{-1} K_{mn}) \right] - \frac{1}{2\sigma^2} Tr[K_{nn} - K_{nm} K_{mm}^{-1} K_{mn}]$$

- We got this extra trace term (the total variance of $p(\mathbf{f}|\mathbf{f}_m)$)

Optimal $\phi^*(\mathbf{f}_m)$ and predictive distribution

- The optimal $\phi^*(\mathbf{f}_m)$ that corresponds to the above bound gives rise to the PP predictive distribution (Csato and Opper, 2002; Seeger and Williams and Lawrence, 2003)
- The approximate predictive distribution is identical to PP

Variational bound for model selection

Learning inducing inputs X_m and (σ^2, θ) using continuous optimization

- Maximize the bound wrt to (X_m, σ^2, θ)

$$F_V = \log\left[N(\mathbf{y}|\mathbf{0}, \sigma^2 I + K_{nm}K_{mm}^{-1}K_{mn})\right] - \frac{1}{2\sigma^2} Tr[K_{nn} - K_{nm}K_{mm}^{-1}K_{mn}]$$

- The first term encourages fitting the data $\mathbf{y}$
- The second trace term says to minimize the total variance of $p(\mathbf{f}|\mathbf{f}_m)$

The trace $Tr[K_{nn} - K_{nm}K_{mm}^{-1}K_{mn}]$ can stand on its own as an objective function for sparse GP learning

Variational bound for model selection

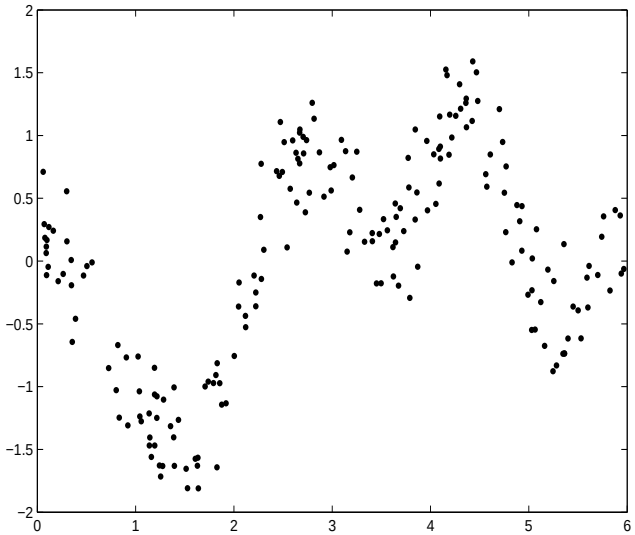
When the bound becomes equal to the true marginal log likelihood, i.e

$$F_V = \log p(\mathbf{y}),$$

then:

- $Tr[K_{nn} - K_{nm}K_{mm}^{-1}K_{mn}] = 0$
- $K_{nn} = K_{nm}K_{mm}^{-1}K_{mn}$
- $p(\mathbf{f}|\mathbf{f}_m)$ becomes a delta function
- We can reproduce the full/exact GP prediction

Illustrative comparison on Ed Snelson's toy data



We compare the traditional PP/DTC log likelihood

$$F_P = \log\left[N(\mathbf{y}|\mathbf{0}, \sigma^2 I + K_{nm}K_{mm}^{-1}K_{mn})\right]$$

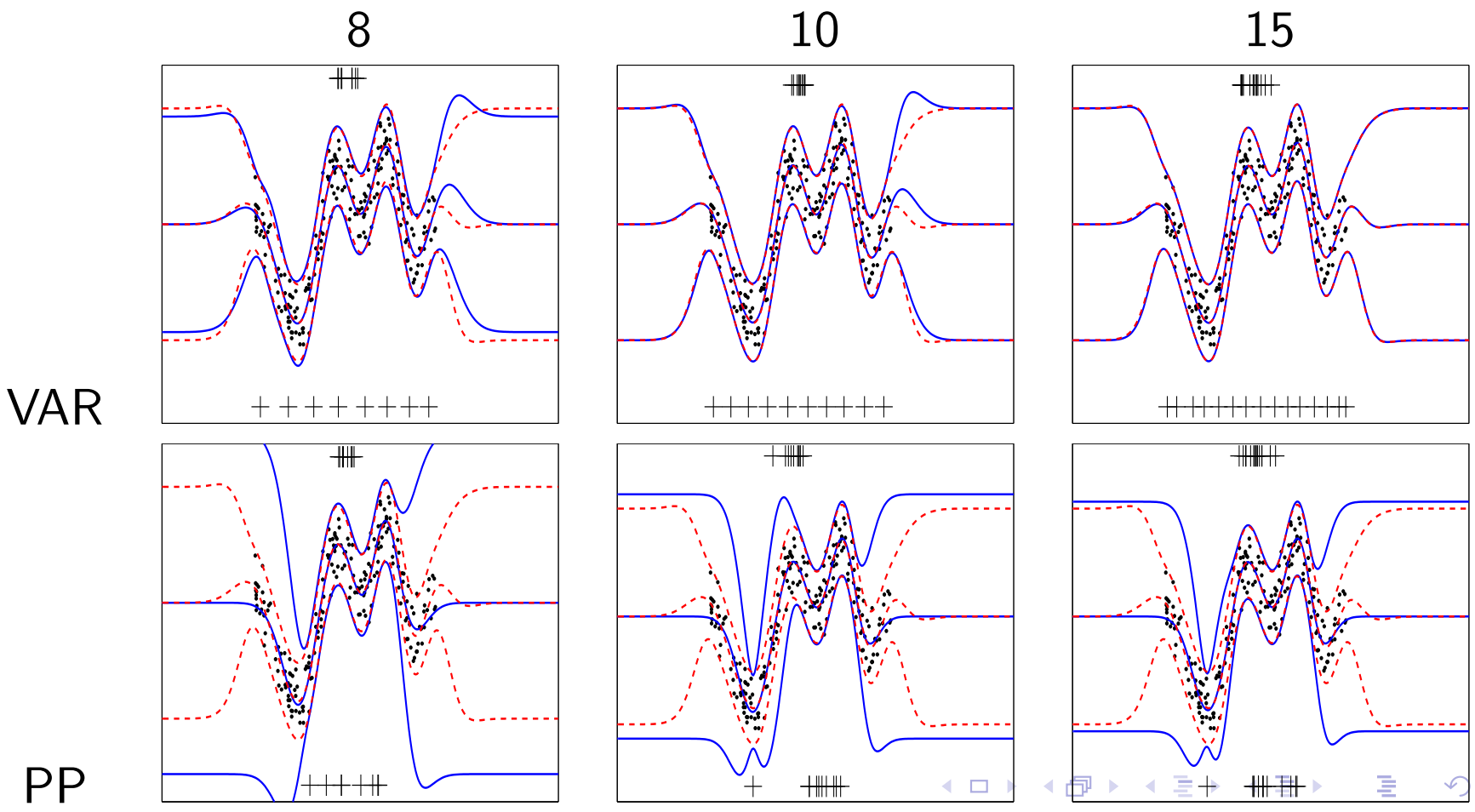
and the bound

$$F_V = \log\left[N(\mathbf{y}|\mathbf{0}, \sigma^2 I + K_{nm}K_{mm}^{-1}K_{mn})\right] - \frac{1}{2\sigma^2} Tr[K_{nn} - K_{nm}K_{mm}^{-1}K_{mn}]$$

We will jointly maximize over $(X_m, \sigma^2, \boldsymbol{\theta})$

Illustrative comparison

200 training points, red line is the full GP, blue line the sparse GP.
We used 8, 10 and 15 inducing points



Illustrative comparison

exponential kernel $\sigma_f^2 \exp\left(-\frac{(\mathbf{x}_m - \mathbf{x}_n)^2}{2\ell^2}\right)$

Table: Model parameters found by variational training

	8	10	15	FULL GP
ℓ^2	0.5050	0.4327	0.3573	**0.3561**
σ_f^2	0.5736	0.6820	0.6854	**0.6833**
σ^2	0.0859	0.0817	0.0796	**0.0796**
MARGL	-63.5282	-57.6909	-55.5708	**-55.5647**

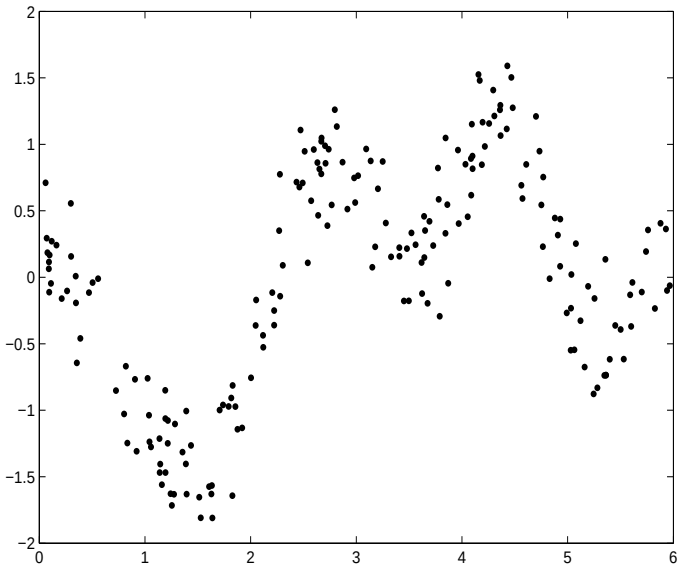
There is a pattern here (observed in many datasets)

- The noise σ^2 decreases with the number of inducing points, until full GP is matched
- This is desirable: The method prefers to explain some signal as noise when the number of inducing variables is not enough

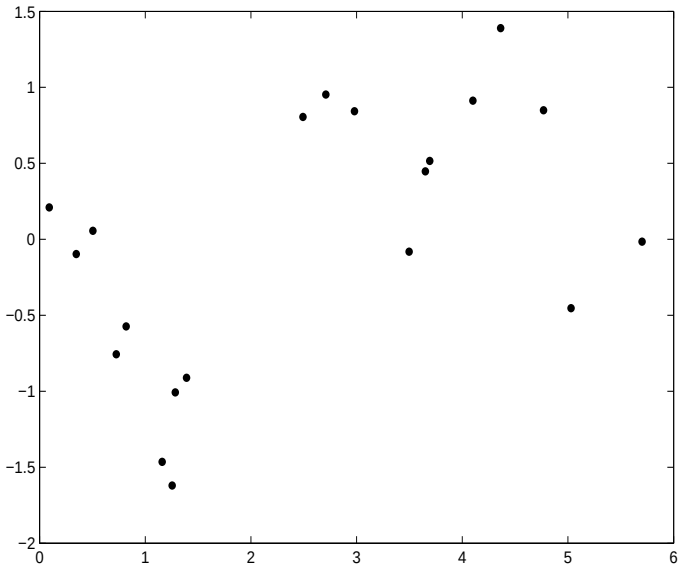
Illustrative comparison

A more challenging problem

From the original 200 training points

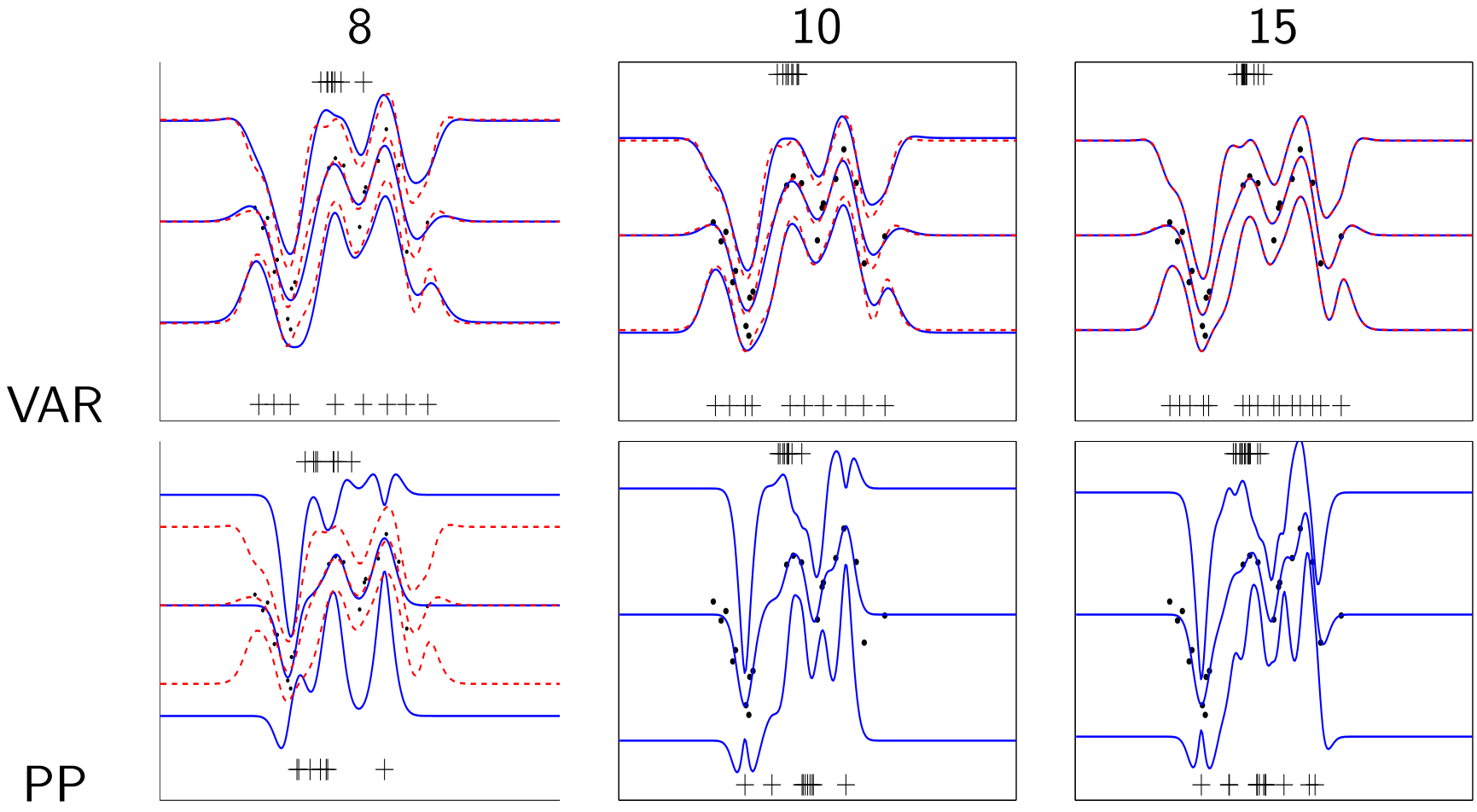


keep[1] only 20



[1] using the MATLAB command $X = X(1 : 10 : end)$

Illustrative comparison



Illustrative comparison

exponential kernel $\sigma_f^2 \exp\left(-\frac{(\mathbf{x}_m - \mathbf{x}_n)^2}{2\ell^2}\right)$

Table: Model parameters found by variational training

	8	10	15	FULL GP
ℓ^2	0.2621	0.2808	0.1804	**0.1798**
σ_f^2	0.3721	0.5334	0.5209	**0.5209**
σ^2	0.1163	0.0846	0.0647	**0.0646**
MARGL	-16.0995	-14.8373	-14.3473	**-14.3461**

Table: Model parameters found by PP marginal likelihood

	8	10	15	FULL GP
ℓ^2	0.0766	0.0632	0.0593	**0.1798**
σ_f^2	1.0846	1.1353	1.1939	**0.5209**
σ^2	0.0536	0.0589	0.0531	**0.0646**
MARGL	-8.7969	-8.3492	-8.0989	**-14.3461**

Variational bound compared to PP likelihood

- The variational method converges to the full GP model in a systematic way as we increase the number of inducing variables
- It tends to find smoother predictive distributions than the full GP (the decreasing σ^2 pattern) when the amount of inducing variables is not enough
- The PP marginal likelihood will not converge to the full GP as we increase the number of inducing inputs and maximize over them
- PP tends to interpolate the training examples

SPGP/FITC marginal likelihood (Snelson and Ghahramani 2006)

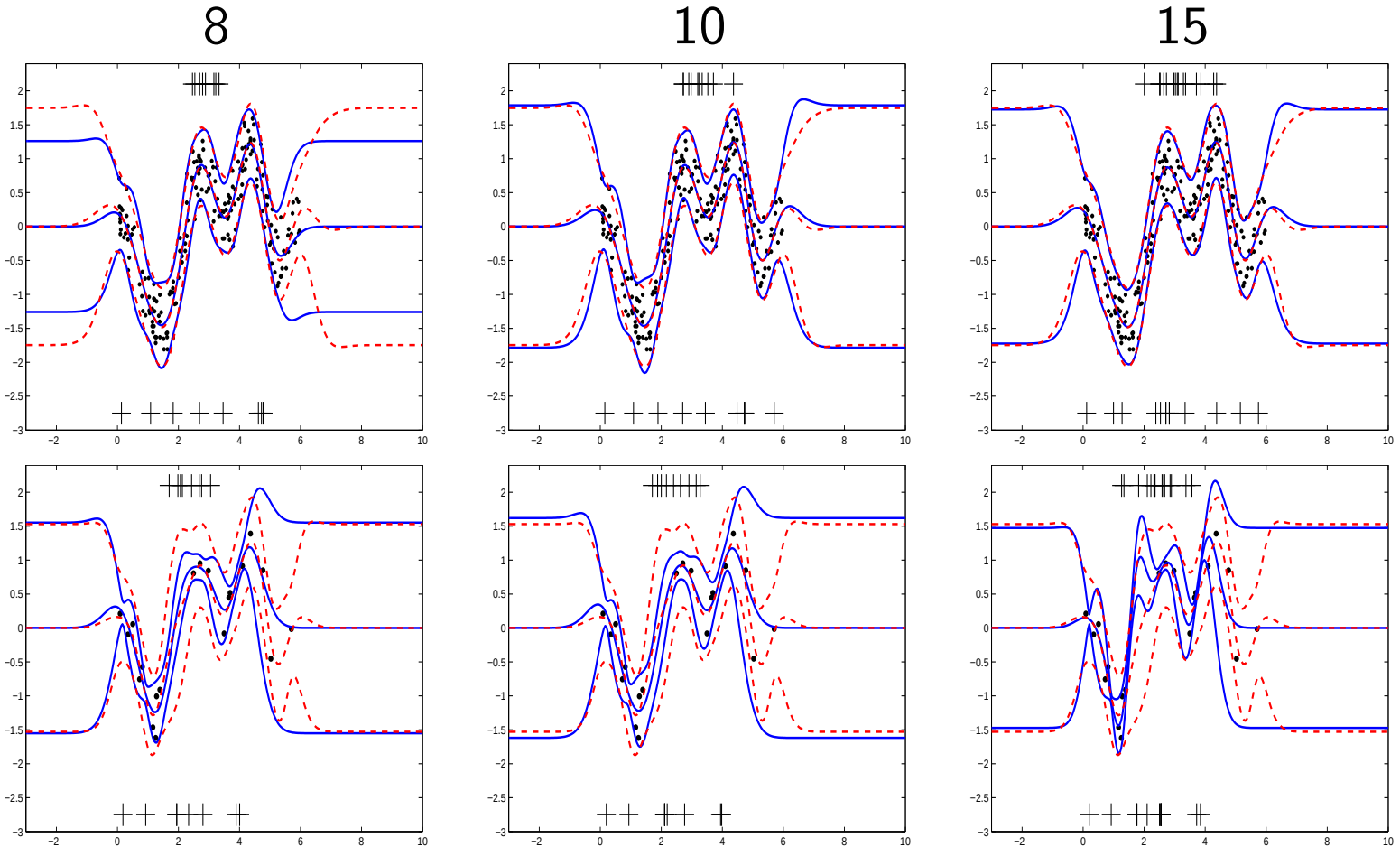
- SPGP uses the following marginal likelihood

$$N(\mathbf{y}|\mathbf{0}, \sigma^2 I + \text{diag}[K_{nn} - K_{nm}K_{mm}^{-1}K_{mn}] + K_{nm}K_{mm}^{-1}K_{mn})$$

- The covariance used is closer to the true thing $\sigma^2 + K_{nn}$ compared to PP
- SPGP uses a non-stationary covariance matrix that can model input-dependent noise
- SPGP is significantly better for model selection than the PP marginal likelihood (Snelson and Ghahramani, 2006, Snelson, 2007)

SPGP/FITC marginal likelihood on toy data

First row is for 200 training points and second row for 20 training points



SPGP/FITC on toy data

Model parameters found by SPGP/FITC marginal likelihood

Table: 200 training points

	8	10	15	
ℓ^2	0.2531	0.3260	0.3096	**0.3561**
σ_f^2	0.3377	0.7414	0.6761	**0.6833**
σ^2	0.0586	0.0552	0.0674	**0.0796**
MargL	-56.4397	-50.3789	-52.7890	**-55.5647**

Table: 20 training points

	8	10	15	
ℓ^2	0.2622	0.2664	0.1657	**0.1798**
σ_f^2	0.5976	0.6489	0.5419	**0.5209**
σ^2	0.0046	0.0065	0.0008	**0.0646**
MargL	-11.8439	-11.8636	-11.4308	**-14.3461**

SPGP/FITC marginal likelihood

- It can be much more robust to overfitting than PP
- Joint learning of inducing points and hyperparameters can cause overfitting
- It is able to model input-dependent noise
  - That is a great advantage in terms of performance measures that involve the predictive variance (like average negative log probability density)
- It will not converge to the full GP as we increase the number of inducing points and optimize over them

Boston-housing dataset

13 inputs, 455 training points, 51 test points. Optimizing only over inducing points X_m. $(\sigma^2, \boldsymbol{\theta})$ fixed to those obtained from full GP

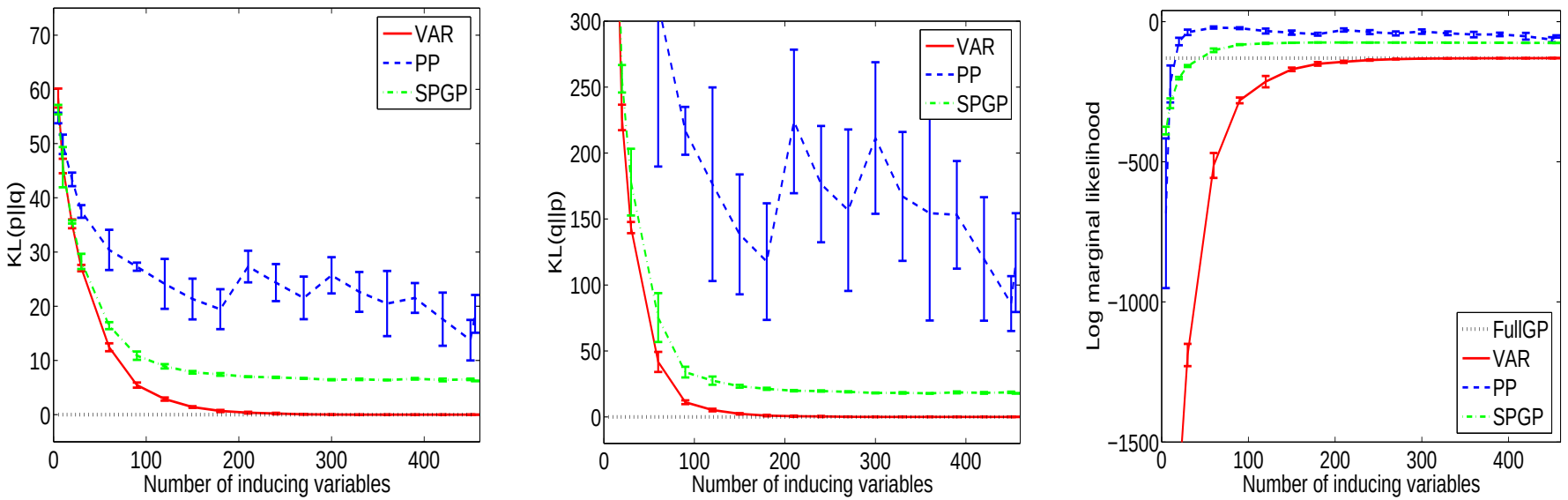


Figure: KLs between full GP predictive distribution (51-dimensional Gaussian) and sparse ones and the marginal likelihood

Only the variational method drops the KLs to zero

Boston-housing dataset

Joint learning of inducing inputs and hyperparameters

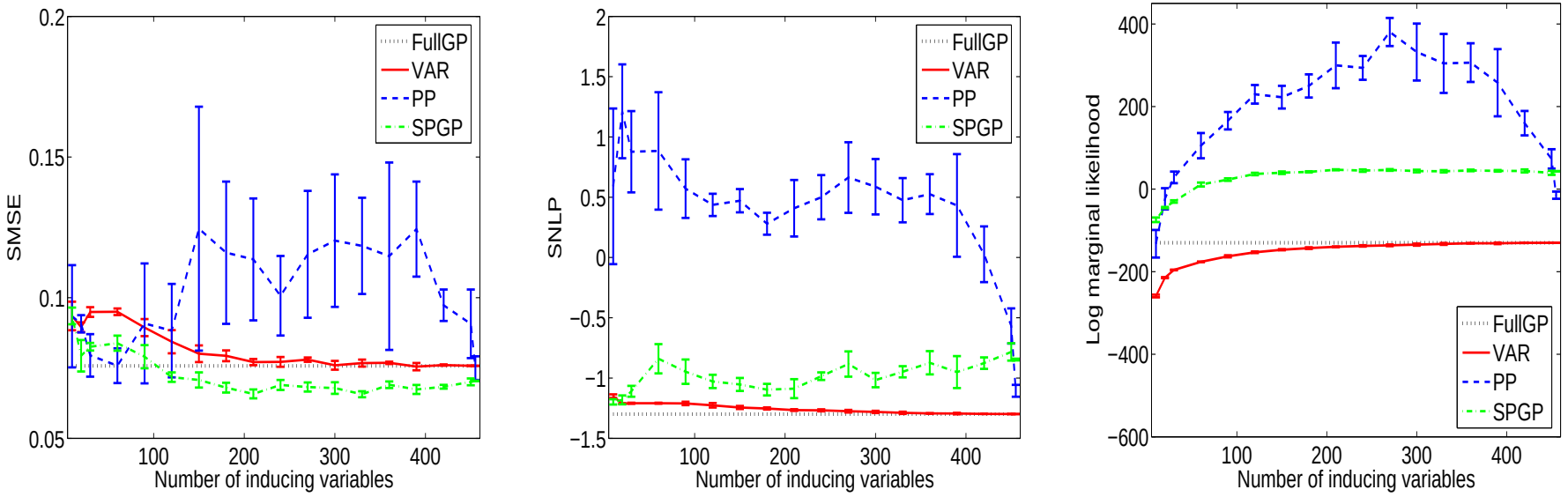


Figure: Standardised mean squared error (SMSE), standardized negative log probability density (SNLP) and the marginal likelihood wrt to the number of inducing points

For 250 points the variational method is very close to full GP

Large datasets

Two large datasets:

- KIN40K dataset: 10000 training, 30000 test, 8 attributes, *http://ida.first.fraunhofer.de/∼anton/data.html*
- SARCOS dataset: 44, 484 training, 4, 449 test, 21 attributes, *http://www.gaussianprocess.org/gpml/data/*

The inputs were normalized to have zero mean and unit variance on the training set and the outputs were centered so as to have zero mean on the training set

kin40k

Joint learning of inducing points and hyperparameters. The subset of data (SD) uses 2000 training points

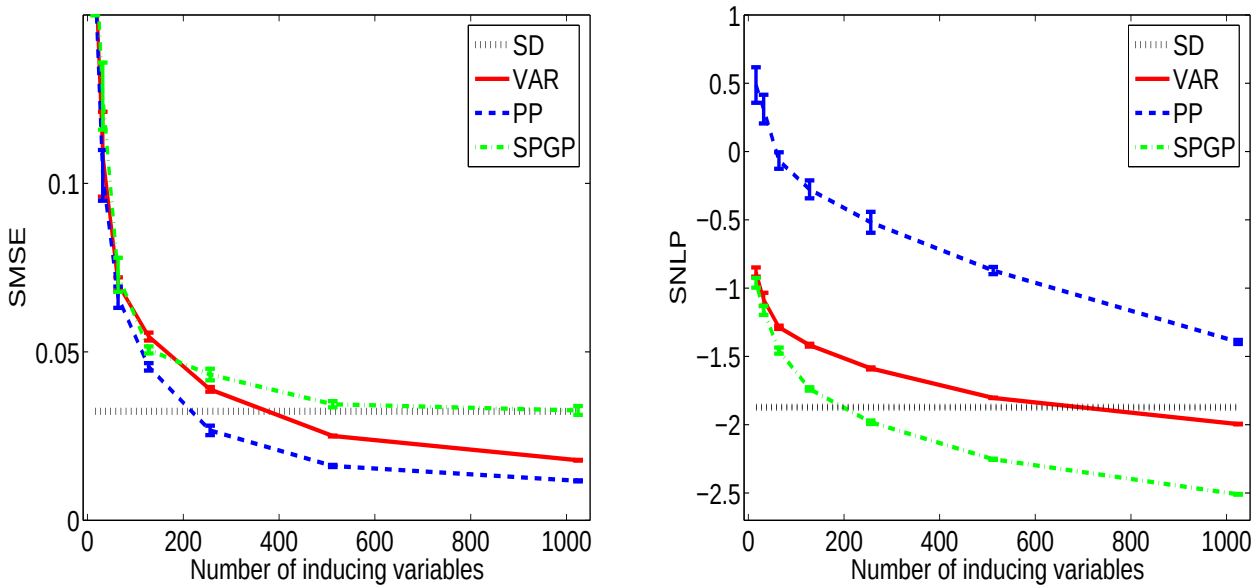


Figure: Standardised mean squared error (SMSE) and standardized negative log probability density (SNLP) wrt to the number of inducing points

sarcos

Joint learning of inducing points and hyperparameters. The subset of data (SD) uses 2000 training points

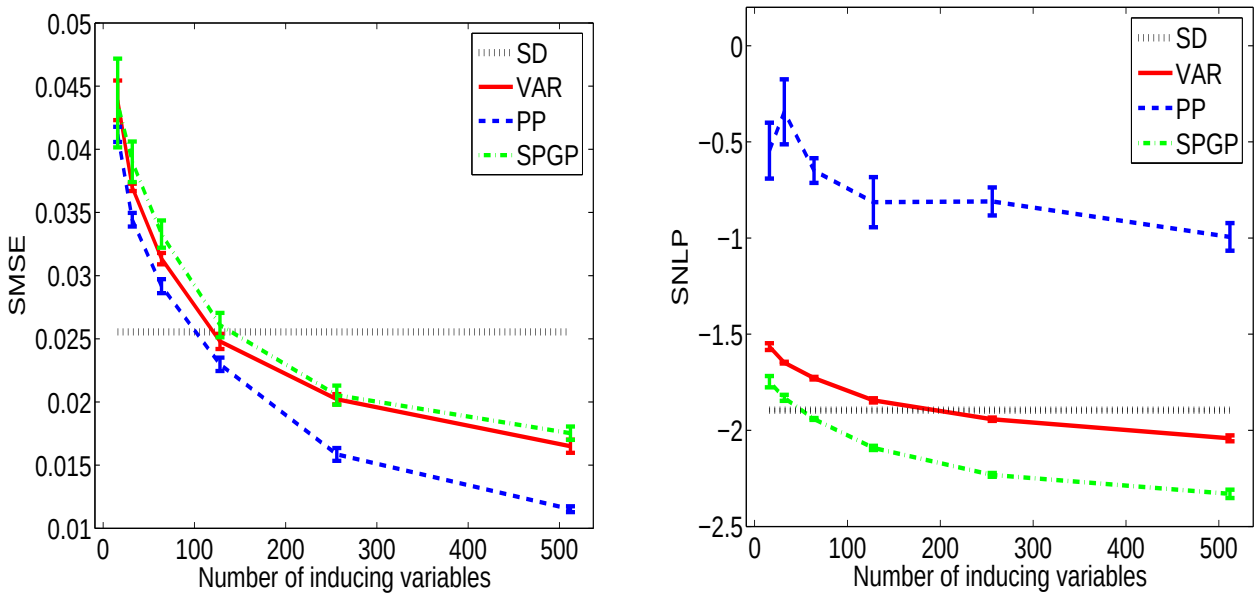


Figure: Standardised mean squared error (SMSE) and standardized negative log probability density (SNLP) wrt to the number of inducing points

Variational bound for greedy model selection

Inducing inputs X_m selected from the training set

- $m \subset \{1, \ldots, n\}$ be indices of the subset of data used as inducing/active variables.
- $n - m$ denotes the remaining training points
- Optimal active latent values $\mathbf{f}_m$ satisfy

$$\begin{aligned} p(\mathbf{f}|\mathbf{y}) &= p(\mathbf{f}_{n-m}|\mathbf{f}_m, \mathbf{y}_{n-m}) p(\mathbf{f}_m|\mathbf{y}) \\ &= p(\mathbf{f}_{n-m}|\mathbf{f}_m) p(\mathbf{f}_m|\mathbf{y}) \end{aligned}$$

- Variational distribution: $q(\mathbf{f}) = p(\mathbf{f}_{n-m}|\mathbf{f}_m)\phi(\mathbf{f}_m)$
- Variational bound:

$$F_V = \log\left[N(\mathbf{y}|\mathbf{0}, \sigma^2 I + K_{nm} K_{mm}^{-1} K_{mn})\right] - \frac{1}{2\sigma^2} Tr[Cov(\mathbf{f}_{n-m}|\mathbf{f}_m)]$$

Variational bound for greedy model selection

Greedy selection with hyperparameters adaption (Seeger, et. al., 2003)

1. Initialization: $m = \emptyset$, $n - m = \{1, \ldots, n\}$
2. Point insertion and adaption:
   - E-like step: Add $j \in J \subset n - m$, into m so as a criterion Δ_j is maximised
   - M-like step: Update (σ^2, θ) by maximizing the approximate marginal likelihood
3. Go to step 2 or stop

For the PP marginal likelihood this is problematic

- Non smooth convergence: The algorithm is not an EM

The variational bound solves this problem. The above procedure becomes precisely a **variational EM algorithm**

Variational bound for greedy model selection

The variational EM property comes out of the Proposition 1

- **Proposition 1.** Let $(m, X_m, \mathbf{f}_m)$ be the current set of active points. Any training point $i \in n - m$ added into the active set can never decrease the lower bound.
- In other words: Any point inserted cannot decrease the divergence $\mathrm{KL}(q(\mathbf{f})||p(\mathbf{f}|\mathbf{y}))$
- E-step (point insertion): Corresponds to an update of the variational distribution

$$q(\mathbf{f}) = p(\mathbf{f}_{n-m}|\mathbf{f}_m)\phi(\mathbf{f}_m)$$

- M-step: Updates the parameters by maximizing the bound

Monotonic increase of the variational bound is guaranteed for any possible criterion Δ

Variational formulation for sparse GP regression

- Define a full GP regression model
- Define a variational distribution of the form

$$q(\mathbf{f}, \mathbf{f}_m) = p(\mathbf{f}|\mathbf{f}_m)\phi(\mathbf{f}_m)$$

- Get the approximate predictive distribution

$$\text{true} \quad p(\mathbf{f}_*|\mathbf{y}) = \int_{\mathbf{f},\mathbf{f}_m} p(\mathbf{f}_*|\mathbf{f}, \mathbf{f}_m)p(\mathbf{f}, \mathbf{f}_m|\mathbf{y})$$

$$\text{approx.} \quad q(\mathbf{f}_*|\mathbf{y}) = \int_{\mathbf{f},\mathbf{f}_m} p(\mathbf{f}_*|\mathbf{f}_m)p(\mathbf{f}|\mathbf{f}_m)\phi(\mathbf{f}_m) = \int_{\mathbf{f}_m} p(\mathbf{f}_*|\mathbf{f}_m)\phi(\mathbf{f}_m)d\mathbf{f}_m$$

- Compute the bound and use it for model selection

Regarding the predictive distribution, what differentiates between SD, PP/DTC, FITC and PITC is the $\phi(\mathbf{f}_m)$ distribution

Variational bound for FITC (similarly for PITC)

- The full GP model that variationally reformulates FITC models input-dependent noise

$$p(\mathbf{y}|\mathbf{f}) = N(\mathbf{y}|\mathbf{f}, \sigma^2 I + \text{diag}[K_{nn} - K_{nm}K_{mm}^{-1}K_{mn}])$$

- FITC log marginal likelihood

$$F_{SPGP}(X_m) = \log\left[N(\mathbf{y}|\mathbf{0}, \Lambda + K_{nm}K_{mm}^{-1}K_{mn})\right]$$

where $\Lambda = \sigma^2 I + \text{diag}[K_{nn} - K_{nm}K_{mm}^{-1}K_{mn}]$

- The corresponding variational bound

$$F_V(X_m) = \log\left[N(\mathbf{y}|\mathbf{0}, \Lambda + K_{nm}K_{mm}^{-1}K_{mn})\right] - \frac{1}{2}Tr[\Lambda^{-1}\widetilde{K}]$$

where $\widetilde{K} = K_{nn} - K_{nm}K_{mm}^{-1}K_{mn}$

- Again a trace term is added

Related work/Conclusion

Related work

- There is an unpublished draft of Lehel Csato and Manfred Opper about variational learning of hyperparameters in sparse GPs
- Seeger (2003) uses also variational methods for sparse GP classification problems

Conclusions

- The variational method can provide us with lower bounds
- This can be very useful for joint learning of inducing inputs and hyperparameters
- Future extensions: classification, differential equations

Acknowledgements

Thanks for feedback to: **Neil Lawrence**, Magnus Rattray, Chris Williams, Joaquin Quiñonero-Candela, Ed Snelson, Manfred Opper, Mauricio Alvarez and Kevin Sharp