Variational Inference for GPs: Presenters

Group1: Stochastic variational inference. Slides 2 - 28

- Chaoqi Wang
- Sana Tonekaboni
- Will Grathwohl

Group2: Variational inference for GPs. Slides 29 - 57

- Trefor Evans
- Kingsley Chang
- Shems Saleh
- James Lucas

Group3: PAC-Bayes. Slides 58 - 68

- Wenyuan Zeng
- Shengyang Sun

Variational Inference for GPs

CSC2541 Presentation

October 17, 2017

Stochastic Variational Inference, by Matt Hoffman, David M. Blei, Chong Wang, John Paisley

Exponential family and Latent Dirichlet Allocation

Exponential family

Exponential family plays a very important role in statistics and it has many good properties.

1. Most of the commonly used distributions are in the exponential family, like, Gaussian, multinomial, exponential, Dirichlet, Poisson, Gamma...
2. Also, some are not in the exponential family: Cauchy, uniform...

Exponential family: definition

The exponential family is defined as the following form:

$$\mathrm{p}(\boldsymbol{x}|\boldsymbol{\eta}) = \exp\{\boldsymbol{\eta}^T T(\boldsymbol{x}) - A(\boldsymbol{\eta})\}$$

1. $\boldsymbol{\eta} \in \mathbb{R}^d$, the natural parameters.
2. $T : \mathcal{X} \to \mathbb{R}^d$, the sufficient statistic.
3. $A(\boldsymbol{\eta}) = \ln \int_{\mathcal{X}} exp\{\eta^T T(x)\} d\mu(x)$, the log normalizer. ($\mu$ is the base measure on a space $\mathcal{X}$)

Sometimes, it will be convenient to use a base measure function $h(\boldsymbol{x}) : \mathcal{X} \to \mathcal{R}_+$, and define:

$$\mathrm{p}(\boldsymbol{x}|\boldsymbol{\eta}) = \mathrm{h}(\boldsymbol{x})\exp\{\boldsymbol{\eta}^T T(\boldsymbol{x}) - A(\boldsymbol{\eta})\}$$

, though h can always be included in μ.

Exponential family: examples

Categorical distribution is a discrete probability distribution that describes the possible results of a random event that can be on one of K possible outcomes. It is defined as:

1. Parameters: k (#categories); $\mu_1, ..., \mu_k$ (event probabilities, $\mu_i > 0$ and $\sum \mu_i = 1$)
2. Support set: $x \in \{1, ..., k\}$
3. PMF: $p(\boldsymbol{x}) = \mu_1^{x_1} \cdots \mu_k^{x_k}$, (here, we overload x as $([x = 1], ..., [x = k])$)
4. Mode: i when $p_i = max(\mu_1, ..., \mu_k)$

Exponential family: examples

We can write the pmf in the standard representation:

$$p(x|\boldsymbol{\mu}) = \prod_{i=1}^{k} \mu_i^{x_i} = exp\{\sum_{i=1}^{k} x_i \ln \mu_i\}$$

, where $\boldsymbol{x} = (x_1, ..., x_k)^T$, and it also can be written as:

$$p(\boldsymbol{x}|\boldsymbol{\mu}) = exp\{\sum_{i=1}^{k-1} x_i \ln \mu_i + (1 - \sum_{i=1}^{k-1} x_i) \ln(1 - \sum_{i=1}^{k-1} \mu_i)\}$$

$$= exp\{\sum_{i=1}^{k-1} x_i \ln(\frac{\mu_i}{1-\sum_{i=1}^{k-1} \mu_i}) + \ln(1 - \sum_{i=1}^{k-1} \mu_i)\}$$

Now, we can identify that:

$$\eta_i = \ln(\frac{\mu_i}{1-\sum_j \mu_j}), \quad T(\boldsymbol{x}) = \boldsymbol{x}, \quad A(\eta) = \ln(1 + \sum_{i=1}^{k-1} exp(\eta_i)), \; h(\boldsymbol{x}) = 1$$

Then,

$$p(\boldsymbol{x}|\boldsymbol{\mu}) = p(\boldsymbol{x}|\boldsymbol{\eta}) = 1 \cdot exp\{\boldsymbol{\eta}^T T(\boldsymbol{x}) - A(\boldsymbol{\eta})\}$$

Exponential family: property

Exponential family has some properties.

1. $D_{KL}(p(\boldsymbol{x}|\boldsymbol{\eta_1})||p(\boldsymbol{x}|\boldsymbol{\eta_2})) = (\boldsymbol{\eta_1} - \boldsymbol{\eta_2})^T \nabla A(\boldsymbol{\eta_1}) - A(\boldsymbol{\eta_1}) + A(\boldsymbol{\eta_2})$
2. $A(\boldsymbol{\eta})$ is convex.
3. $\nabla A(\boldsymbol{\eta}) = \mathbb{E}[T(\boldsymbol{x})] \approx \frac{1}{N} \sum_i T(\boldsymbol{x}^{(i)})$
4. $\nabla^2 A(\boldsymbol{\eta}) = \mathbb{E}[T(\boldsymbol{x})T(\boldsymbol{x})^T] - \mathbb{E}[T(\boldsymbol{x})]\mathbb{E}[T(\boldsymbol{x})^T] = Var[T(\boldsymbol{x})]$

Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is a generative probabilistic model for collections of discrete data such as text corpora.

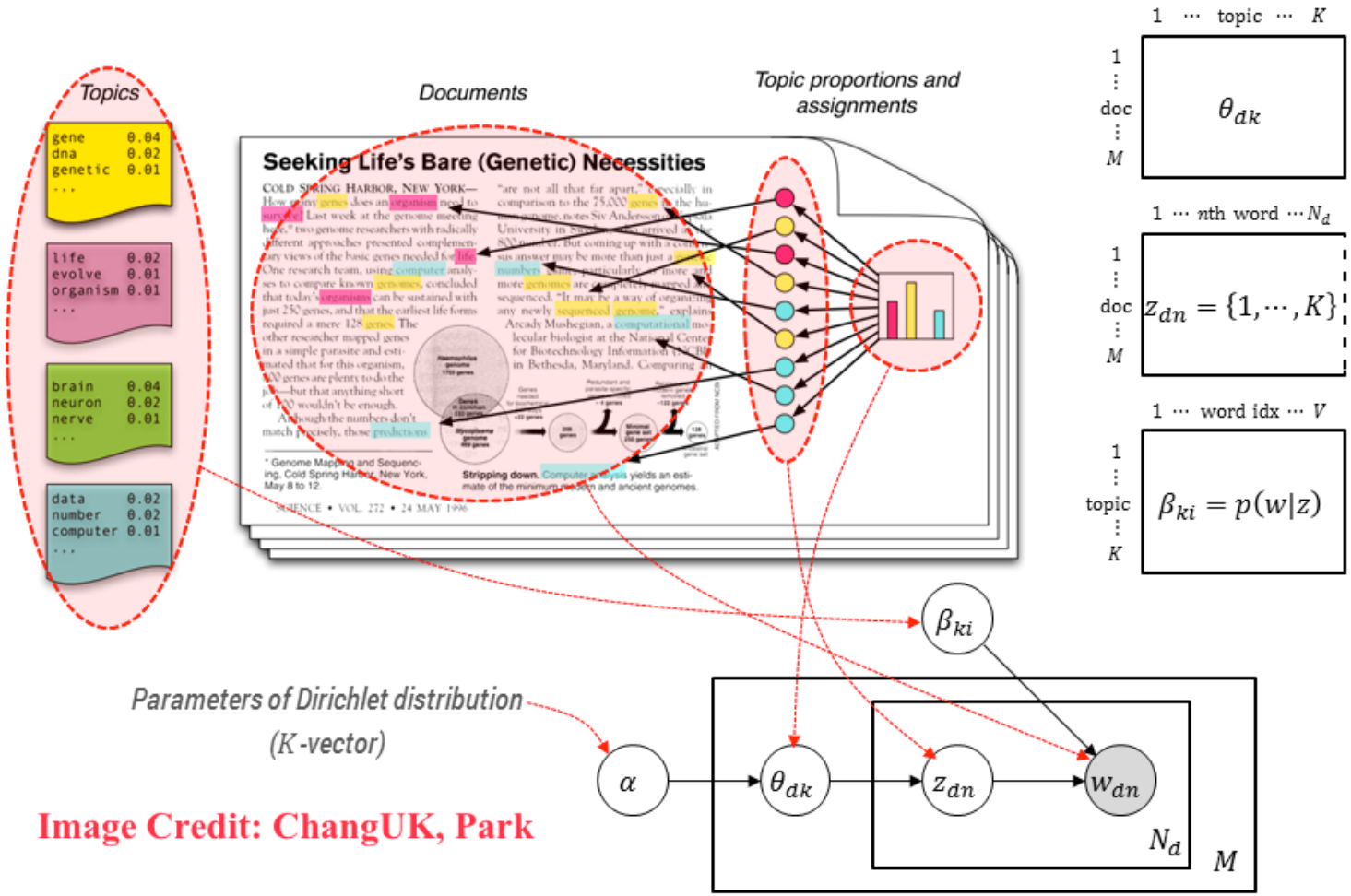


Image Credit: ChangUK, Park

LDA: process

The generative process of LDA model can be summarized as:

1. Draw topics β_k from $Dirichlet(\eta, ..., \eta)$ for $k \in \{1, ..., K\}$
2. For each document $d \in \{1, ..., D\}$:
   1. Draw topic proportions θ_d from $Dirichlet(\alpha, ..., \alpha)$
   2. For each word $w \in \{1, ..., N\}$:
      - Draw topic assignment z_{dn} from $Multinomial(\theta_d)$
      - Draw word w_{dn} from $Multinomial(\beta_{z_{dn}})$

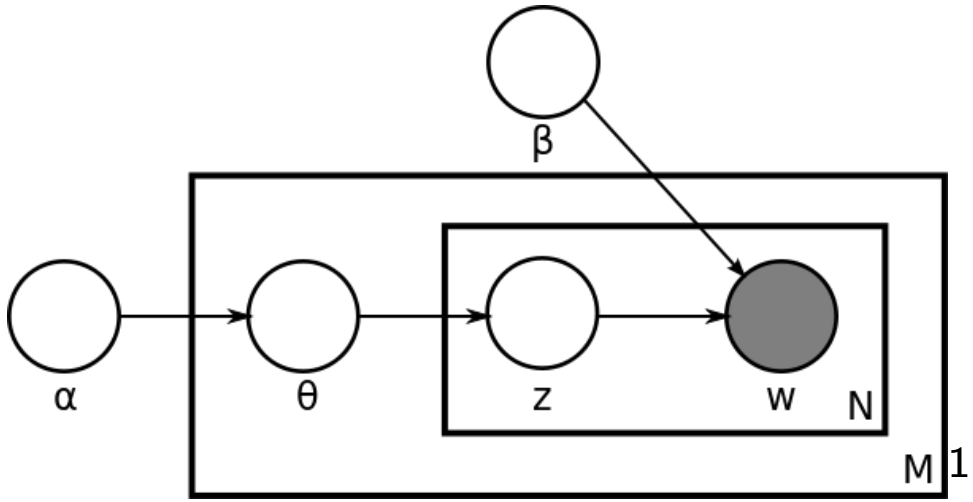
Latent Dirichlet Allocation: notations

There are some notations used in LDA model:

1. w_{dn} is the nth word in dth document. Each word is an element in the fixed vocabulary of V terms.
2. β_k is a V dimensional vector, on a $V-1$ simplex. The wth entry in topic k is β_{kw}
3. θ_d is the associated topic proportions of dth document. It is a point on the $K-1$ simplex.
4. z_{dn} indexes the topic from which w_{dn} is drawn. It is assumed that each word in each document is drawn from a single topic.

LDA: inference

Graphical model representation of LDA. The boxes are plates representing replicates. The outer plate represents documents, while the inner plate represents the repeated choice of topics and words within a document.



The joint distribution is:

$$p(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{w}|\boldsymbol{\beta}, \boldsymbol{\alpha}) = p(\boldsymbol{\theta}|\alpha) \prod_{n=1}^{N} p(z_n|\boldsymbol{\theta}) p(w_n|z_n, \boldsymbol{\beta})$$

[1]Blei, David M.; Ng, Andrew Y.; Jordan, Michael I (January 2003). Lafferty, John, ed. "Latent Dirichlet Allocation". Journal of Machine Learning Research. 3 (45): pp. 9931022. doi:10.1162/jmlr.2003.3.4-5.993

LDA: inference

The key inferential problem that we need to solve in order to use LDA is that of computing the posterior distribution of the hidden variables given a document:

$$p(\boldsymbol{\theta}, \boldsymbol{z}|\boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{p(\boldsymbol{\theta},\boldsymbol{z},\boldsymbol{w}|\boldsymbol{\beta},\boldsymbol{\alpha})}{p(\boldsymbol{w}|\boldsymbol{\alpha},\boldsymbol{\beta})}$$

However, the denominator is computationally intractable.

LDA: inference

One way to approximate the posterior is variational inference. In mean-field variational inference, the variational distributions of each variable are in the same family as the complete conditional. We have:

- $p(z_{dn} = k|\boldsymbol{\theta}_d, \boldsymbol{\beta_{1:K}}, , w_{dn}) \propto exp\{\ln \theta_{dk} + \ln \beta_{k,w_{dn}}\}$,
- $p(\boldsymbol{\theta_d}|z_d) = Dirichlet(\alpha + \sum_{n=1}^{N} z_{dn})$,
- $p(\boldsymbol{\beta_k}|\boldsymbol{z}, \boldsymbol{w}) = Dirichlet(\eta + \sum_{d=1}^{D}\sum_{n=1}^{N} z_{dn}^{k} w_{dn})$

So, the corresponding variational distributions are:

- $q(z_{dn})$ =Multinomial(ϕ_{dn}), for each update:
  $\phi_{dn} \propto exp\{\Psi(\gamma_{dk}) + \Psi(\lambda_{k,w_{dn}}) - \Psi(\sum_v \lambda_{kv})\}$ for $n \in \{1, ..., N\}$
- $q(\boldsymbol{\theta}_d) = Dirichlet(\boldsymbol{\gamma}_d)$, for each update, $\gamma_d = \alpha + \sum_{n=1}^{N} \phi_{dn}$
- $q(\boldsymbol{\beta}_k) = Dirichlet(\boldsymbol{\lambda}_k)$, for each update,
  $\lambda_k = \eta + \sum_{d=1}^{D}\sum_{n=1}^{N} \phi_{dn}^{k} w_{dn}$

LDA: inference

Before updating the topics $\lambda_{1:K}$, we need to compute the local variational parameters for every document. This is particularly wasteful in the beginning of the algorithm when, before completing the first iteration, we must analyze every document with randomly initialized topics.

Stochastic Variational Inference, by Matt Hoffman, David M. Blei, Chong Wang, John Paisley

Variational Inference

Variational Inference

- **Goal:** approximate the posterior distribution of a probabilistic model by introducing a distribution over the hidden variables, and optimizing the parameters of that distribution.

Our class of models involves:

- Obsevations $x = x_{1:N}$
- Global hidden variables β
- Local hidden variables $z = z_{1:N}$
- Fixed parameters α (For simplicity we assume that they only govern the global hidden variables)

Global vs. Local Hidden Variables

- Global hidden variables β : parameters endowed with a prior $p(\beta)$
- Local hidden variables $z = z_{1:N}$: contains the hidden structure that governs each observation

The difference is determined by **conditional dependencies**:

$$p(x_n, z_n | x_{-n}, z_{-n}, \beta, \alpha) = p(x_n, z_n | \beta, \alpha)$$

Also, the complete conditional distribution of the hidden variables are in the exponential family

$$q(\beta | x, z, \alpha) = h(\beta) \exp(\eta_g(x, z, \alpha)^T t(\beta) - a_g \eta_g(x, z, \alpha))$$
$$q(z_{nj} | x_n, z_{nj}, \beta) = h(z_{nj}) \exp(\eta_l(x_n, z_{nj}, \beta)^T t(z_{nj}) - a_l \eta_l(x_n, z_{nj}, \beta))$$

Mean-field Variational Inference

- Mean-field variational inference: a variational inference family where each hidden variable is **independent** and governed by its own **variational parameter**
  λ govern the global variables and $\phi_{\mathbf{n}}$ govern the local variables

$$\mathrm{q}(\mathrm{z},\beta) = \mathrm{q}(\beta|\lambda)\prod_{n=1}^{N}\prod_{j=1}^{J} q(z_{\mathrm{nj}}|\phi_{\mathrm{nj}})$$

- Also, we set $\mathrm{q}(\beta|\lambda)$ and $\mathrm{q}(\mathrm{z}_{\mathrm{nj}}|\phi_{\mathrm{nj}})$ to be in the same exponential family as the complete conditional distributions
  $\mathrm{p}(\beta|x,z) and p(z_{\mathrm{nj}}|x_{\mathrm{n}}, z_{\mathrm{n\text{-}j}},\beta)$

$$\mathrm{q}(\beta|\lambda) = \mathrm{h}(\beta)\exp\lambda^{T}\mathrm{t}(\beta)\text{-}\mathrm{a}_{\mathrm{g}}(\lambda)$$
$$\mathrm{q}(\mathrm{z}_{nj}|\phi_{nj}) = \mathrm{h}(\mathrm{h}_{nj})\exp\phi_{nj}^{T}\,\mathrm{t}(\mathrm{z}_{nj})\text{-}\mathrm{a}_{\mathrm{l}}(\phi_{nj})$$

Batch Variational Bayes

$$\mathcal{L} = \mathbf{E}[\log q(z,\beta)] - \mathbf{E}[\log p(x,z,\beta)]$$

- Coordinate update for λ: $\lambda = \mathrm{E}_q[\eta_g(x,z,\alpha)]$
- Coordinate update for ϕ: $\phi_{nj} = \mathrm{E}_q[\eta_l(x_n,z_{n-j},\beta)]$
- Therefore, we can optimize our objective function with an easy coordinate ascend and in closed form

Batch Variational Bayes Algorithm

1. Initialize $\lambda^{(0)}$ randomly
2. **Repeat**
3. **for** each local variational parameter ϕ_{nj} do
4. Update $\phi_{nj}, \phi_{nj}^{(t)} = \mathrm{E}_{q^{(t-1)}}[\eta_{l,j}(x_n, z_{n-j}, \beta)]$
5. **End for**
6. Update the global variational parameters $\lambda^{(t)} = \mathrm{E}_{q^{(t)}}[\eta_g(z_{1:N}, x_{1:N})]$

Stochastic Variational Inference

- **Solution:** Use a Stcohastic optimization, repeatedly subsample the data to form noisy estimates of the natural gradient of the ELBO

$$\hat{\nabla}_{\lambda}\mathcal{L} = \mathbf{E}_{\phi}[\eta_g(\mathsf{x},\mathsf{z},\alpha)] - \lambda$$

$$\hat{\nabla}_{\phi_{nj}}\mathcal{L} = \mathbf{E}_{\lambda,\phi_{n-j}}[\eta_l(\mathsf{x}_n,\mathsf{z}_{n-j},\beta)] - \phi_{nj}$$

Some benefits of Natural Gradients:

- The natural gradient points in the direction of steepest ascent in the Riemannian space
- Converges faster
- It is cheaper to compute

Stochastic Variational Inference Algorithm

1. Initialize $\lambda^{(0)}$ randomly
2. Set a step size ρ_t appropriately
3. **Repeat**
4. Sample a datapoint x_i uniformly from the dataset
5. Update the local variational parameter of the sample as if we were doing coordinate ascend

$$\phi = \mathrm{E}_{\lambda^{(t-1)}}[\eta_g(x_i^N, z_i^N)]$$

6. Update the current estimate of the global variational parameters

$$\lambda^{(t)} = \lambda^{(t-1)} + \rho_t \hat{\nabla}_\lambda \mathcal{L} = (1\text{-}\rho_t)\lambda^{(t-1)} + \rho_t \, \mathrm{E}_\phi[\eta_g(x_i^N, z_i^N)]$$

A Review Of Stochastic Gradient Variational Bayes

Last lecture instroduced Stochastic Gradient Variational Bayes (SGVB).

In SGVB, the ELBO: $\mathcal{L}(\phi) = \mathbb{E}_{x\sim D}[\mathbb{E}_{q(z;\phi)}[\log p(x,z) - \log q(z|x,\phi)]]$ is optimized via stochastic gradient decent where we estimate $\frac{\partial \mathcal{L}(\phi)}{\partial \phi}$ using monte-carlo samples.

In SGVB our estimator is produced via a 2-step hierarchical sampling procedure:

- We draw a minibatch of data x_i (or $x_i, x_j, x_k, ...$)
- We draw a minibatch of samples $z_i \sim q(z_i|x_i,\phi)$
- We estimate $\frac{\partial \mathcal{L}(\phi)}{\partial \phi} \approx \frac{\partial}{\partial \phi}(\log p(x_i,z_i) - \log q(z_i|x_i,\phi))$

Where we have reparameterized $z_i = f(x_i,\epsilon,\phi)$ with $\epsilon \sim p(\epsilon)$.

Thus: $\frac{\partial \mathcal{L}(\phi)}{\partial \phi} \approx \frac{\partial}{\partial \phi}(\log p(x_i, f(x_i,\epsilon,\phi)) - \log q(f(x_i,\epsilon,\phi)|x_i,\phi))$

Required Properties

Both SVI and SGVB require certain assumptions to hold before they can be applied

SVI:

- There exists an analytic form of $\frac{\partial \mathcal{L}(\phi)}{\partial \phi}$ for each model paraemter ϕ.
- The approximate posterior $q(z|\phi)$ must be in the same exponential family as $p(z)$

SGVB:

- The likelihood $p(x, z)$ must be differentiable wrt z.
- The approximate posterior $q(z|x, \phi)$ must be differentiabe wrt its parameters ϕ.
- There exists a differentiable reparameterization $f(x, \epsilon, \phi)$, $\epsilon \sim p(\epsilon)$ such that $z = f(x, \epsilon, \phi)$ is distributed as $q(z|x, \phi)$.

SVI

Benefits:

- Performs natural gradient decent.
- Invariant to parameterization.
- Exponential family provides a rich set of both continuous and discrete data to be modeled.
- Allows for scalable inference over large datasets.

Downsides:

- Parameters of variational approximation $q(z|\phi)$ must be exactly the exponential family parameters limiting complexity of the relationship between $q(z|\phi)$ and data x.
- Analytic Forms of ELBO derivitives are nessesary.
- $q(z|\phi)$ must be in the same exponential family as $p(z)$.

SGVB

Benefits:

- Weaker Modeling Assumptions can be made.
- $p(x, z)$ and $q(z|\phi)$ need only be differentiable wrt their parameters.
- Complex, nonlinear relationships between data and latent variables may be learned.
- Reparameterization allows for low-variance gradient estimates for all model parameters.
- Allows for scalable inference over large datasets.

SGVB Continued

Downsides:

- Naive natural gradient decent is intractible in nonlinear probablistic models (although see Dr. Grosse's recent work for exciting progress towards approximate NGD for neural network models).
- Not invariant to model parameterization so extra care must be taken to ensure proper results.
- Reparameterization limits the type of posterior approximations we can use to continuous distributions (like gaussian, laplace).
- No proof exists showing that reparameterization gradients have lower varaince than score function estimator.

Variational Inference for GPs

Sparse Gaussian Process

Gaussian Processes Review

$$\mathbf{y} = \{f(\mathbf{x}_i) + \epsilon\}_{i=1}^n, \quad \mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n, \quad \mathbf{x}_i \in \mathbb{R}^d$$

A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution. It is completely specified by its mean and covariance function.

Gaussian Processes Review

$$\mathbf{y} = \{f(\mathbf{x}_i) + \epsilon\}_{i=1}^n, \quad \mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n, \quad \mathbf{x}_i \in \mathbb{R}^d$$

A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution. It is completely specified by its mean and covariance function.

Prior $\mathbf{f} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{\mathrm{X},\mathrm{X}}), \qquad [\mathbf{K}_{\mathrm{X},\mathrm{X}}]_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$

Joint Prior $$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim p(\mathbf{y}, \mathbf{f}_*) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}_{\mathrm{X},\mathrm{X}} + \sigma_2 \mathbf{I} & \mathbf{K}_{\mathrm{X},*} \\ \mathbf{K}_{*,\mathrm{X}} & \mathbf{K}_{*,*} \end{bmatrix}\right)$$

Conditional Distribution $\mathbf{f}_* | \mathbf{X}, \mathbf{y}, \mathbf{X}_* \sim \mathcal{N}(\mathbb{E}[\mathbf{f}_*], \mathrm{cov}[\mathbf{f}_*])$

$$\mathbb{E}[\mathbf{f}_*] = (\mathbf{K}_{\mathrm{X},\mathrm{X}} + \sigma^2 \mathbf{I})^{-1} \mathbf{y}$$
$$\mathrm{cov}[\mathbf{f}_*] = \mathbf{K}_{*,*} - \mathbf{K}_{*,\mathrm{X}} (\mathbf{K}_{\mathrm{X},\mathrm{X}} + \sigma^2 \mathbf{I})^{-1} \mathbf{K}_{\mathrm{X},*}$$

Model Selection

Our kernels are typically parameterized by some hyperparameters $\boldsymbol{\theta}$. For example, the squared exponential kernel

$$k(\mathbf{x}, \mathbf{z}) = \exp(-||\mathbf{x}-\mathbf{z}||_2^2/\theta^2)$$

Log Marginal likelihood

$$\log p(\mathbf{y}|\boldsymbol{\theta}, \sigma^2, \mathbf{X}) = -\tfrac{1}{2}\log|\mathbf{K}_{\mathsf{X},\mathsf{X}}+\sigma^2\mathbf{I}| - \tfrac{1}{2}\mathbf{y}_{\mathsf{X}}^T(\mathbf{K}_{\mathsf{X},\mathsf{X}}+\sigma^2\mathbf{I})^{-1}\mathbf{y}_{\mathsf{X}} - \tfrac{N}{2}\log(2\pi)$$

Requires $\mathcal{O}(n^3)$ time and $\mathcal{O}(n^2)$ storage.

Modifying the Joint Prior

$$p(\mathbf{f}, \mathbf{f}_*) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}_{X,X} & \mathbf{K}_{X,*} \\ \mathbf{K}_{*,X} & \mathbf{K}_{*,*} \end{bmatrix}\right)$$

We want to modify this joint prior to reduce computational requirements. Assume $\mathbf{f}_*, \mathbf{f}$ conditionally independent given set of inducing point locations $\mathbf{Z} = \{\mathbf{z}_i\}_{i=1}^m$ and responses $\mathbf{u} = \{u_i\}_{i=1}^m$.

$$p(\mathbf{f}, \mathbf{f}_*) = \int p(\mathbf{f}, \mathbf{f}_*, \mathbf{u}) d\mathbf{u}$$

$$q(\mathbf{f}, \mathbf{f}_*) = \int p(\mathbf{f}_*|\mathbf{u}) q(\mathbf{f}|\mathbf{u}) p(\mathbf{u}) d\mathbf{u}$$

we make an approximation to the training conditional

$$p(\mathbf{f}|\mathbf{u}) \approx q(\mathbf{f}|\mathbf{u})$$

Fully Independent Training Conditional (FITC)

We approximate the training conditional with an independent distribution (diagonal covariance)

$$p(\mathbf{f}|\mathbf{u}) = \mathcal{N}(\mathbf{K}_{\mathrm{X},\mathrm{Z}}\mathbf{K}_{\mathrm{X},\mathrm{Z}}^{-1}\mathbf{u}, \quad \mathbf{K}_{\mathrm{X},\mathrm{X}} - \mathbf{Q}_{\mathrm{X},\mathrm{X}})$$
$$q(\mathbf{f}|\mathbf{u}) = \mathcal{N}(\mathbf{K}_{\mathrm{X},\mathrm{Z}}\mathbf{K}_{\mathrm{X},\mathrm{Z}}^{-1}\mathbf{u}, \quad \mathrm{diag}[\mathbf{K}_{\mathrm{X},\mathrm{X}} - \mathbf{Q}_{\mathrm{X},\mathrm{X}}])$$

where $\mathbf{Q}_{\mathrm{X},\mathrm{X}} = \mathbf{K}_{\mathrm{X},\mathrm{Z}}\mathbf{K}_{\mathrm{Z},\mathrm{Z}}^{-1}\mathbf{K}_{\mathrm{Z},\mathrm{X}}$. This gives

$$p(\mathbf{f}, \mathbf{f}_*) \approx \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{Q}_{\mathrm{X},\mathrm{X}} + \mathrm{diag}[\mathbf{K}_{\mathrm{X},\mathrm{X}} - \mathbf{Q}_{\mathrm{X},\mathrm{X}}] & \mathbf{Q}_{\mathrm{X},*} \\ \mathbf{Q}_{*,\mathrm{X}} & \mathbf{Q}_{*,*} + \mathrm{diag}[\mathbf{K}_{*,*} - \mathbf{Q}_{*,*}] \end{bmatrix}\right)$$

from this, the cost of computing our conditional distribution decreases from $\mathcal{O}(n^3) \to \mathcal{O}(m^2 n)$ time and from $\mathcal{O}(n^2) \to \mathcal{O}(mn)$ storage.

Variational Inference for GPs

Adapted from a presentation by "Variational Model Selection for Sparse Gaussian Process Regression" *Christopher. P. Ley* (2016) [2]

[2] http://games.cmm.uchile.cl/media/uploads/posts/SGP_presentation.pdf

Variational learning of inducing variables I

- *Titsias (2009)* proposed a variational lower bound to approximate the true posterior.
- The ideal inducing variables should serve as sufficient statistics to the observation $\mathbf{y}$.

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

- The augmented true posterior $p(\mathbf{f}, \mathbf{f}_m|\mathbf{y})$ factorises as

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

Variational learning of inducing variables II

- The key is that $q(\mathbf{f}, \mathbf{f}_m)$ must satisfy a factorisation that holds for optimal inducing variables:

$$True : p(\mathbf{f}, \mathbf{f_m}|\mathbf{y}) = p(\mathbf{f}|\mathbf{f_m})p(\mathbf{f_m}|\mathbf{y})$$

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

Variational learning of inducing variables III

- This gives rise to the variational distribution

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

  where $\phi(\boldsymbol{f}_m)$ is an unconstrained variational distribution over $\boldsymbol{f}_m$
- We now can use standard variational Bayesian inference where we minimise the Kullback-Leibler divergence

$$KL(q(\boldsymbol{f}, \boldsymbol{f}_m)||p(\boldsymbol{f}, \boldsymbol{f}_m|\boldsymbol{y}))$$

- Which gives us an equivalent maximum bound on the true log marginal likelihood:

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

Computation of the variational bound I

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

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

$$
E[\boldsymbol{f}|\boldsymbol{f}_m] = K_{nm} K_{mm}^{-1} \boldsymbol{f}_m
$$

$$
\mathrm{Cov}[\boldsymbol{f}|\boldsymbol{f}_m] = K_{nn} - K_{nm} K_{mm}^{-1} K_{mn}
$$

Bias-Variance Decomposition

$$\int_f p(\mathbf{f}|\mathbf{f}_m) log p(\mathbf{y}|\mathbf{f}) d\mathbf{f} = \overbrace{log[N(\mathbf{y}|E[\mathbf{f}|\mathbf{f}_m], \sigma^2_{noise} I)]}^{bias} - \underbrace{\frac{1}{2\sigma^2_{noise}} Tr[Cov(\mathbf{f}|\mathbf{f}_m)]}_{variance}$$

- Recall that the bias-variance decomposition in L2 loss:

$$E_{t\sim p(t|x)}[(y-t)^2] = \underbrace{(y - E_{t\sim p(t|x)}[t])^2}_{bias} + \underbrace{Var[t|x]}_{variance}$$

Computation of the variational bound II

- Merge the logs

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

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

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

where $\mathrm{Cov}[\boldsymbol{f} | \boldsymbol{f}_m] = K_{nn} - K_{nm} K_{mm}^{-1} K_{mn}$

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 $p(\mathbf{f}|\mathbf{f}_m)$)

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

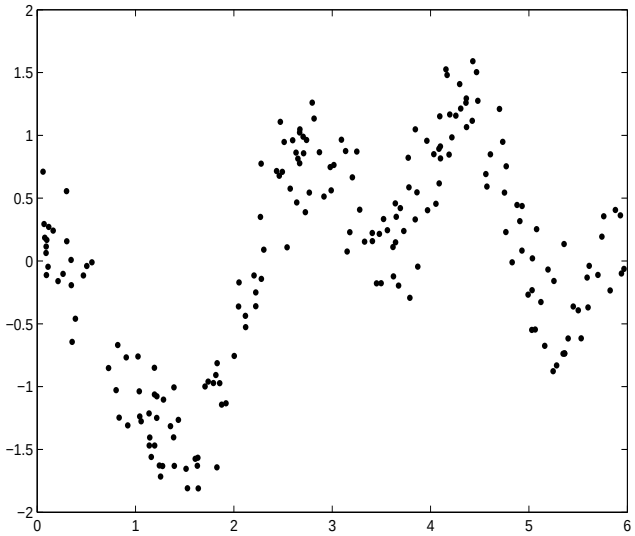
Variational bound for model selection

When the approximation is the same as the full covariance matrix, i.e.

$$K_{nn} = K_{nm} K_{mm}^{-1} K_{mn}$$

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

Illustrative comparison on Ed Snelson's toy data



We compare the traditional PP/DTC log likelihood

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

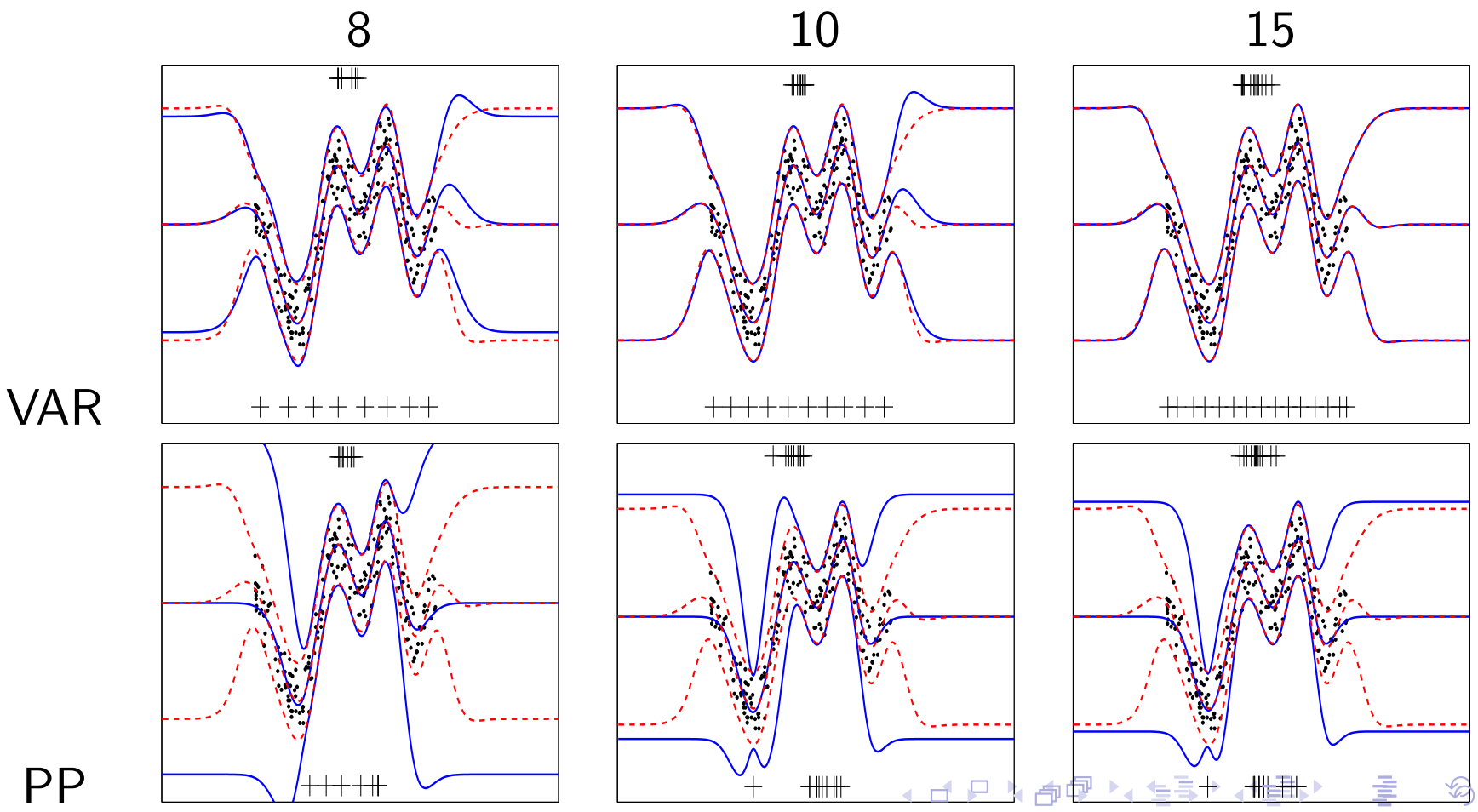
and the bound

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

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



Variational bound compared to PP likelihood

- The variational method (VFE) converges to the full GP model as we increase the number of inducing variables. But PP would not.
- VFE tends to find smoother distribution than the fill GP when the inducing vairaibles are not enough.
- PP tends to interpolate the training examples.

Variational Inference for GPs

FITC and VFE Comparison

Overview of the two methods

- Negative Log Marginal Likelihood:

$$\mathcal{F} = \frac{N}{2}\log(2\pi) + \underbrace{\frac{1}{2}\log|Q_{\mathbf{ff}} + G|}_{\text{complexity penalty}} + \underbrace{\frac{1}{2}\mathbf{y}^{\mathsf{T}}(Q_{\mathbf{ff}} + G)^{-1}\mathbf{y}}_{\text{data fit}} + \underbrace{\frac{1}{2\sigma_n^2}\operatorname{tr}(T)}_{\text{trace term}},$$

$$\begin{aligned} G_{\text{FITC}} &= \operatorname{diag}[K_{\mathbf{ff}} - Q_{\mathbf{ff}}] + \sigma_n^2 I & \qquad G_{\text{VFE}} &= \sigma_n^2 I \\ T_{\text{FITC}} &= 0 & \qquad T_{\text{VFE}} &= K_{\mathbf{ff}} - Q_{\mathbf{ff}}. \end{aligned}$$

  - **Data fit**: Penalizes data outside the covariance ellipse $Q_{ff} + G$
  - **Complexity penalty**: Characterizes the volume of possible datasets compatible with the data fit term. (Occam's Razor)
  - **Trace term**: Ensure that objective function is a true lower bound to marginal likelihood of the full GP

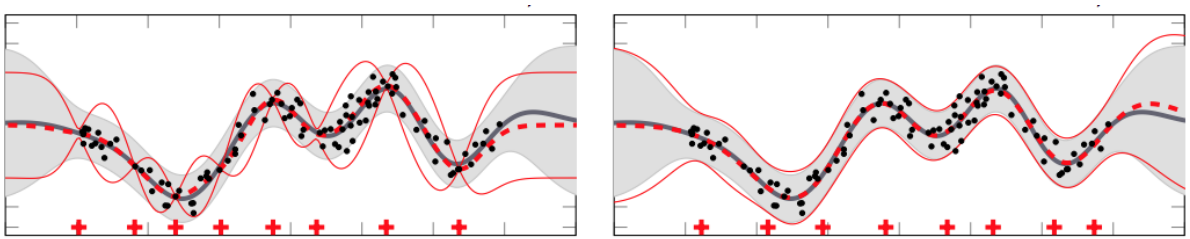
Overview

Points of comparison:

1. Noise Variance
2. Number of Inducing Inputs
3. True GP Posterior
4. Optimas

Noise Variance

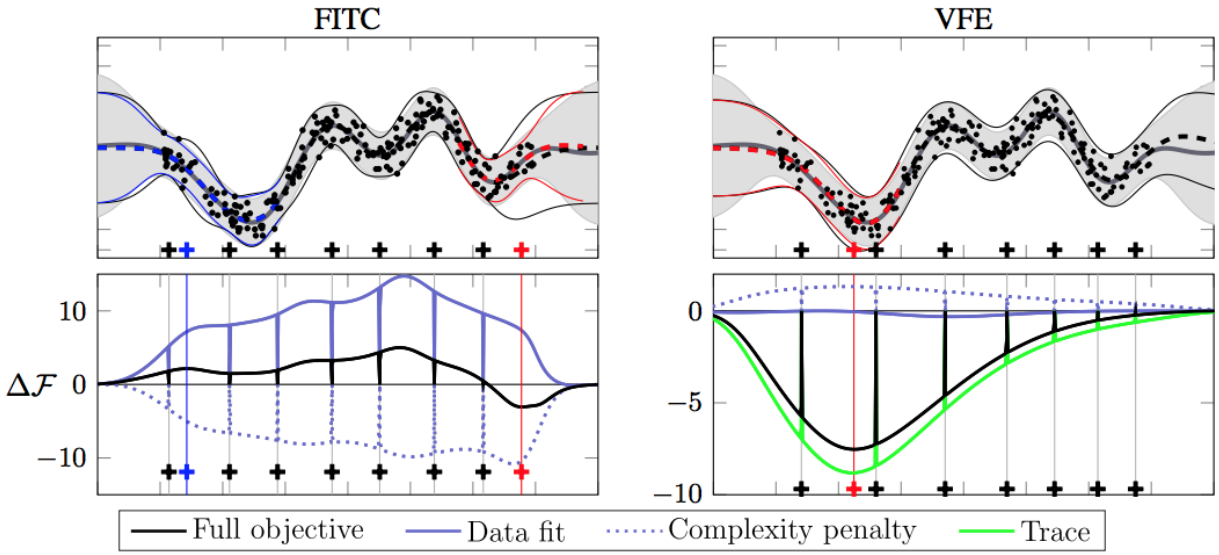
- FITC (left) underestimates the noise variance while VFE (right) overestimates it



- In full GP, we assume homoscedastic (input independent) noise with parameter σ_n^2
- FITC uses the diagonal term $diag(K_{ff} - Q_{ff})$ in G_{FITC} as heteroscedastic (input dependent) noise
- The trace and data fit terms in VFE can be reduced by increasing σ_n^2 causing a bias towards overestimation of the noise variance

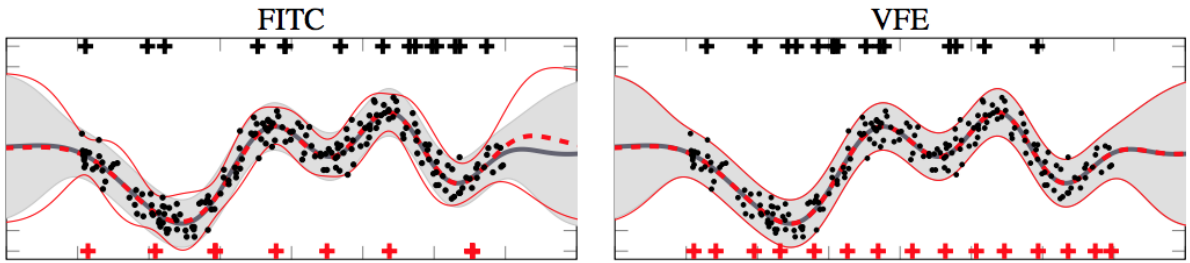
Number of Inducing Inputs

- VFE improves with additional inducing inputs while FITC may ignore them



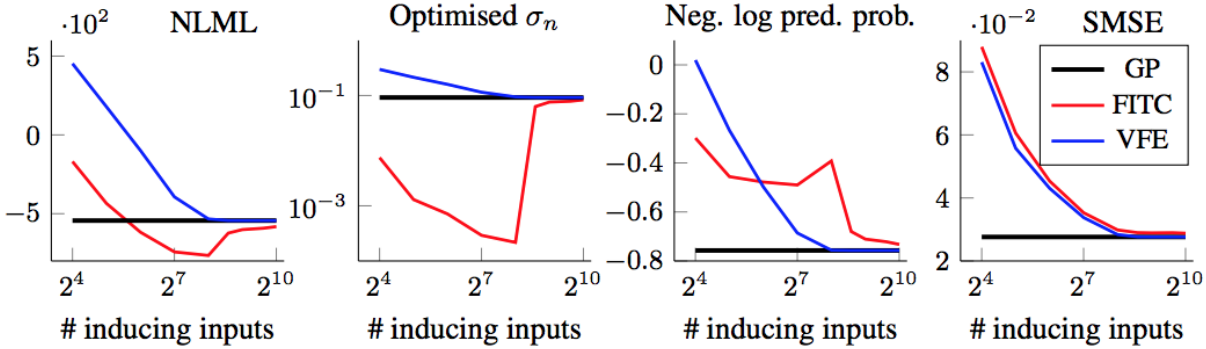
Number of Inducing Inputs

- FITC avoids the penalty of added inducing inputs by clumping them
- This also means FITC doesn't recover the full GP even when given enough resources



Local Optima

- FITC relies on local optima



- The minimum found by FITC through clumping still exists in the optimization surface of many inducing points
- Optimizing FITC is easier than VFE
- Optimizing VFE function includes initializing the inducing points with k-means and initially fixing the hyperparameters
- VFE recognizes a good solution when we initialize it with the FITC solution

Summary

- FITC Behaviour
  - Over-estimation of marginal likelihood
  - Severe under-estimation of noise variance
  - Wasting modelling resources
  - Not recovering true posterior
- VFE Behaviour
  - True bound to the marginal likelihood of full GP
  - Behaves predictably
  - Improves with extra resources
  - Recovers true posterior when possible
- FITC remains easier to optimise and gives a good local optima
- The VFE objective function is recommended since its optimization difficulties can be mitigated by careful initialization, random starts and FITC initialization
- In practice, it ends up depending on the dataset

Variational Inference for GPs

SVI for GPs

Are sparse GPs enough?

- Standard GPs require $\mathcal{O}(n^3)$ time complexity and $\mathcal{O}(n^2)$ storage.

Are sparse GPs enough?

- Standard GPs require $\mathcal{O}(n^3)$ time complexity and $\mathcal{O}(n^2)$ storage.
- Sparse GPs cut this down to $\mathcal{O}(nm^2)$ time complexity and $\mathcal{O}(nm)$ storage.

Are sparse GPs enough?

- Standard GPs require $\mathcal{O}(n^3)$ time complexity and $\mathcal{O}(n^2)$ storage.
- Sparse GPs cut this down to $\mathcal{O}(nm^2)$ time complexity and $\mathcal{O}(nm)$ storage.
- But we have huge datasets where n is on the order of millions, or billions!

How can we hope to fit (even sparse) GPs to datasets of this magnitude?

1: Initialize $\lambda^{(0)}$ randomly.
2: Set the step-size schedule ρ_t appropriately.
3: **repeat**
4: Sample a data point x_i uniformly from the data set.
5: Compute its local variational parameter,

$$\phi = \mathbb{E}_{\lambda^{(t-1)}}[\eta_g(x_i^{(N)}, z_i^{(N)})].$$

6: Compute intermediate global parameters as though x_i is replicated N times,

$$\hat{\lambda} = \mathbb{E}_{\phi}[\eta_g(x_i^{(N)}, z_i^{(N)})].$$

7: Update the current estimate of the global variational parameters,

$$\lambda^{(t)} = (1 - \rho_t)\lambda^{(t-1)} + \rho_t \hat{\lambda}.$$

8: **until** forever

Stochastic variational inference!

The GP Variational Bound

- Titsias, 2009. showed that

$$\log p(\mathbf{y}|\mathbf{X}) = \log \int p(\mathbf{y}|\mathbf{u})p(\mathbf{u})d\mathbf{u} \tag{1}$$

$$\geq \log \int \exp(\mathcal{L}_1)p(\mathbf{u})d\mathbf{u} := \mathcal{L}_2 \tag{2}$$

Where $\mathcal{L}_1 := \mathbb{E}_{p(\mathbf{f}|\mathbf{u})}[\log p(\mathbf{y}|\mathbf{f})]$. Remember:

- $\mathbf{f}$ - Function evaluated at $\mathbf{X}$
- $\mathbf{y}$ - Noisy observation of $\mathbf{f}$
- $\mathbf{u}$ - Value of function evaluated at inducing points ($\mathbf{Z}$)

The GP Variational Bound

- Titsias, 2009. showed that

$$\log p(\mathbf{y}|\mathbf{X}) = \log \int p(\mathbf{y}|\mathbf{u})p(\mathbf{u})d\mathbf{u} \tag{1}$$

$$\geq \log \int \exp(\mathcal{L}_1)p(\mathbf{u})d\mathbf{u} := \mathcal{L}_2 \tag{2}$$

  Where $\mathcal{L}_1 := \mathbb{E}_{p(\mathbf{f}|\mathbf{u})}[\log p(\mathbf{y}|\mathbf{f})]$. Remember:
  - $\mathbf{f}$ - Function evaluated at $\mathbf{X}$
  - $\mathbf{y}$ - Noisy observation of $\mathbf{f}$
  - $\mathbf{u}$ - Value of function evaluated at inducing points ($\mathbf{Z}$)
- We can compute this analytically (as shown before). Posterior can be viewed as "collapsed" over inducing points

The GP Variational Bound

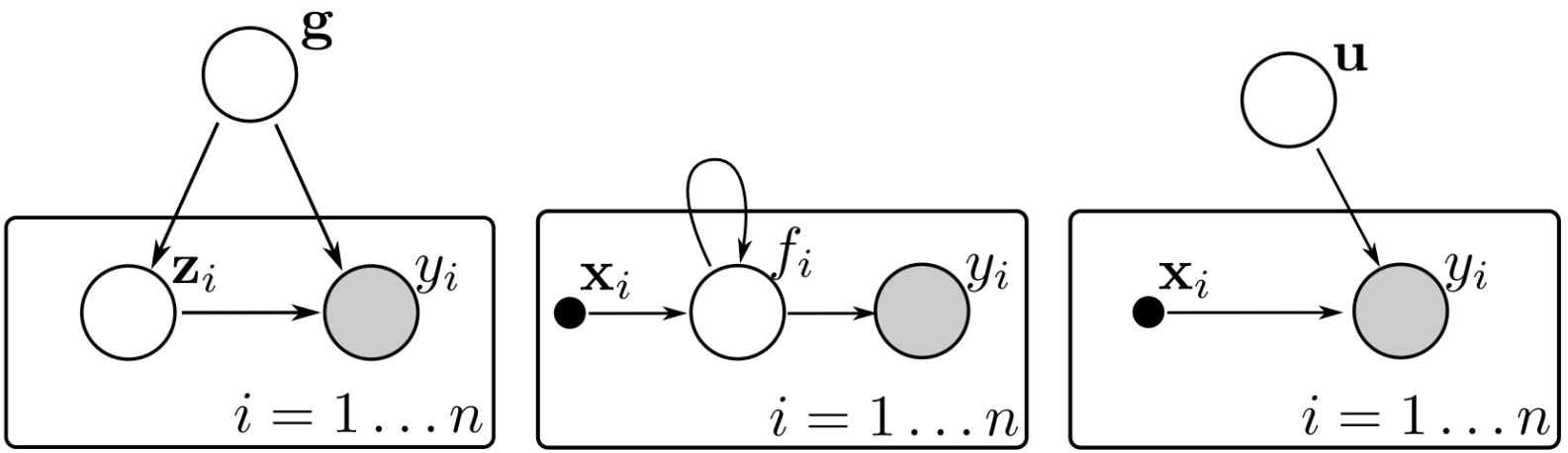
- Titsias, 2009. showed that

$$\log p(\mathbf{y}|\mathbf{X}) = \log \int p(\mathbf{y}|\mathbf{u})p(\mathbf{u})d\mathbf{u} \tag{1}$$

$$\geq \log \int \exp(\mathcal{L}_1)p(\mathbf{u})d\mathbf{u} := \mathcal{L}_2 \tag{2}$$

  Where $\mathcal{L}_1 := \mathbb{E}_{p(\mathbf{f}|\mathbf{u})}[\log p(\mathbf{y}|\mathbf{f})]$. Remember:
  - $\mathbf{f}$ - Function evaluated at $\mathbf{X}$
  - $\mathbf{y}$ - Noisy observation of $\mathbf{f}$
  - $\mathbf{u}$ - Value of function evaluated at inducing points ($\mathbf{Z}$)
- We can compute this analytically (as shown before). Posterior can be viewed as "collapsed" over inducing points
- We need to be explicit about inducing points to do SVI

Requirements for SVI



(a) Requirements for SVI (b) Gaussian Process regression (c) Variational GP regression

Marginalisation of $\mathbf{u}$ introduces dependencies in the observations. We need to adjust our VIGP regression model to allow us to use SVI...

From a collapsed posterior to global latent variables

- We instead treat the inducing points as global latent variables, with variational distribution $q(\mathbf{u})$

From a collapsed posterior to global latent variables

- We instead treat the inducing points as global latent variables, with variational distribution $q(\mathbf{u})$
- We then get a new bound which we can use for SVI.

$$\log p(\mathbf{y}|\mathbf{X}) \geq \mathbb{E}_{q(\mathbf{u})}[\mathcal{L}_1 + \log p(\mathbf{u}) - \log q(\mathbf{u})] := \mathcal{L}_3 \tag{3}$$

(Remember, $\mathcal{L}_1 := \mathbb{E}_{p(\mathbf{f}|\mathbf{u})}[\log p(\mathbf{y}|\mathbf{f})]$)

From a collapsed posterior to global latent variables

- We instead treat the inducing points as global latent variables, with variational distribution $q(\mathbf{u})$
- We then get a new bound which we can use for SVI.

$$\log p(\mathbf{y}|\mathbf{X}) \geq \mathbb{E}_{q(\mathbf{u})}[\mathcal{L}_1 + \log p(\mathbf{u}) - \log q(\mathbf{u})] := \mathcal{L}_3 \tag{3}$$

(Remember, $\mathcal{L}_1 := \mathbb{E}_{p(\mathbf{f}|\mathbf{u})}[\log p(\mathbf{y}|\mathbf{f})]$)

- The optimal $q(\mathbf{u})$ is Gaussian, which leads to,

$$\mathcal{L}_3 = \sum_{i=1}^{n} \left\{ \log \mathcal{N}(y_i|\hat{\boldsymbol{\mu}}, \beta^{-1}) + \cdots \right\} - KL(q(\mathbf{u})||p(\mathbf{u})) \tag{4}$$

Omitting some terms for brevity (see paper).

From a collapsed posterior to global latent variables

- We instead treat the inducing points as global latent variables, with variational distribution $q(\mathbf{u})$
- We then get a new bound which we can use for SVI.

$$\log p(\mathbf{y}|\mathbf{X}) \geq \mathbb{E}_{q(\mathbf{u})}[\mathcal{L}_1 + \log p(\mathbf{u}) - \log q(\mathbf{u})] := \mathcal{L}_3 \tag{3}$$

(Remember, $\mathcal{L}_1 := \mathbb{E}_{p(\mathbf{f}|\mathbf{u})}[\log p(\mathbf{y}|\mathbf{f})]$)

- The optimal $q(\mathbf{u})$ is Gaussian, which leads to,

$$\mathcal{L}_3 = \sum_{i=1}^{n} \left\{ \log \mathcal{N}(y_i|\hat{\boldsymbol{\mu}}, \beta^{-1}) + \cdots \right\} - KL(q(\mathbf{u})||p(\mathbf{u})) \tag{4}$$

Omitting some terms for brevity (see paper).

- We can write this as a sum over data points allowing SVI!

Inference with this new bound

- We perform SVI using natural gradient updates

Inference with this new bound

- We perform SVI using natural gradient updates
- Exponential family leads to a nice form of the updates
- See the paper for the derived update rules

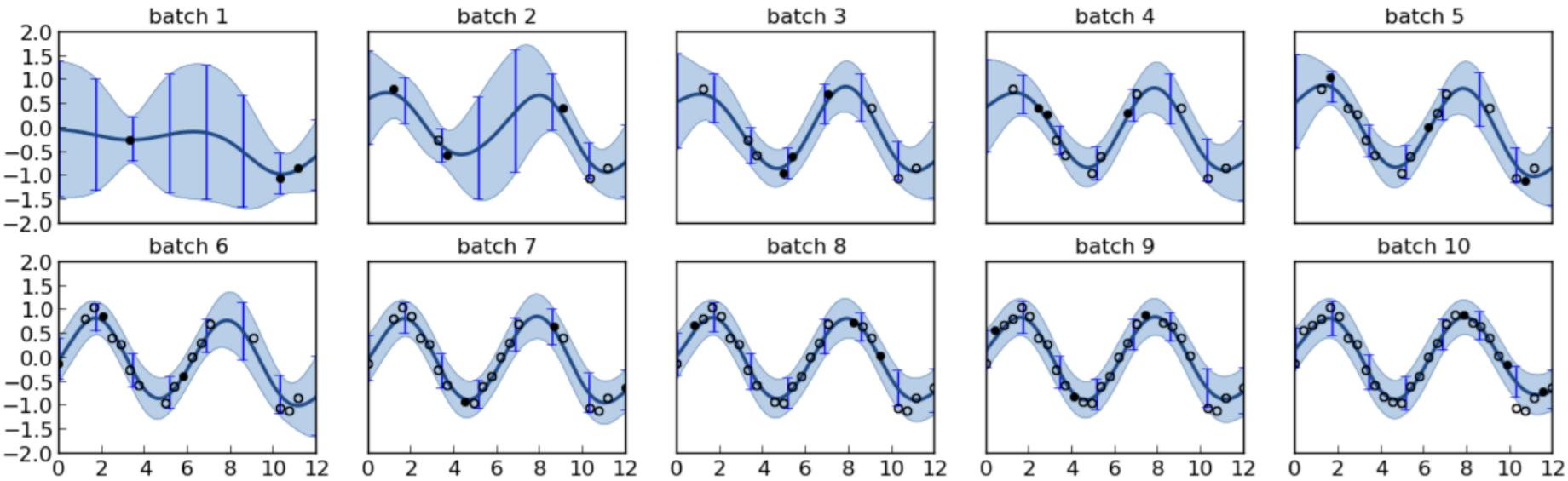
Inference with this new bound

- We perform SVI using natural gradient updates
- Exponential family leads to a nice form of the updates
- See the paper for the derived update rules
- Training updates are now $\mathcal{O}(m^3)$!

Inference with this new bound

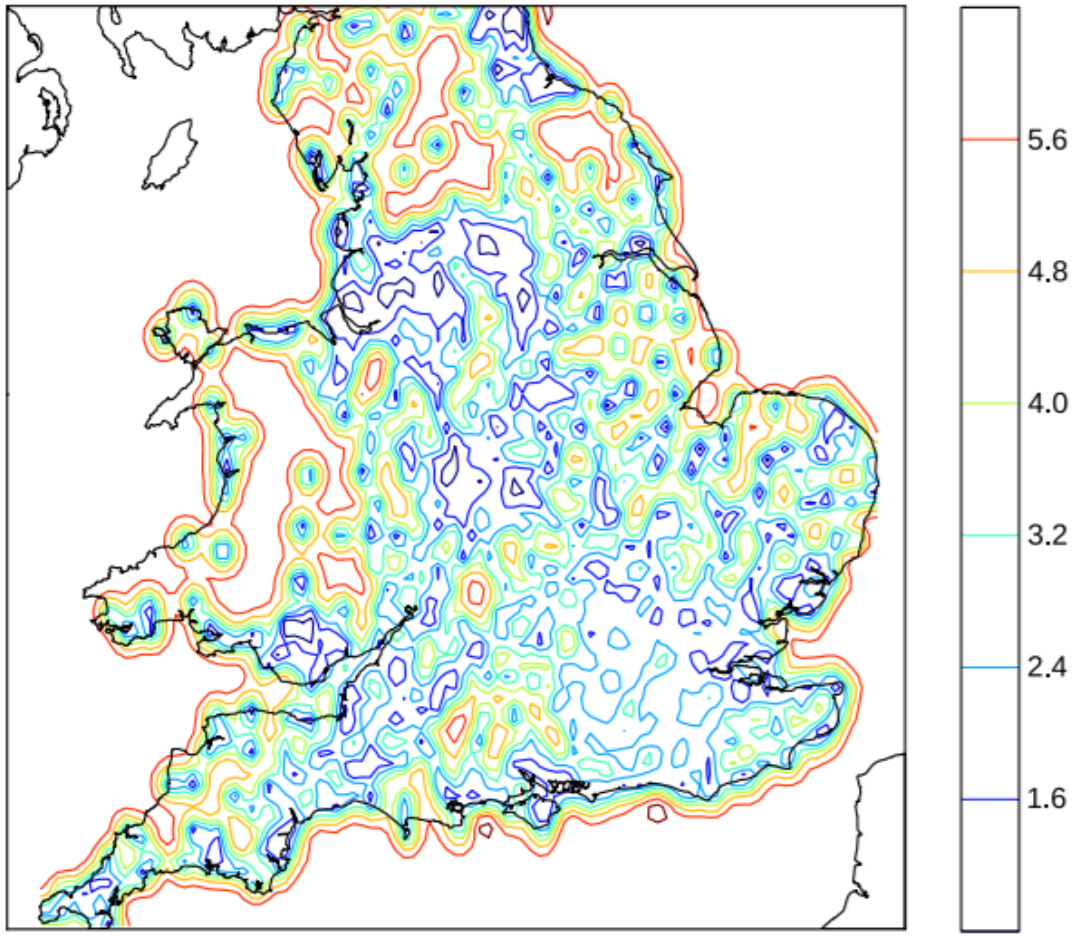
- We perform SVI using natural gradient updates
- Exponential family leads to a nice form of the updates
- See the paper for the derived update rules
- Training updates are now $\mathcal{O}(m^3)$!
- Can also use non-Gaussian likelihoods because of the $\mathcal{L}_3$ factorisation. This normally requires approximations.

Experiments I



Each pane shows the posterior of the GP updated per batch. The variational distribution $q(\mathbf{u})$ is shown by the error bars.

Experiments II



Posterior variance of apartment price by postal region.

Summary

- We cannot do SVI on the collapsed VI bound.

Summary

- We cannot do SVI on the collapsed VI bound.
- But we can refactorize it.

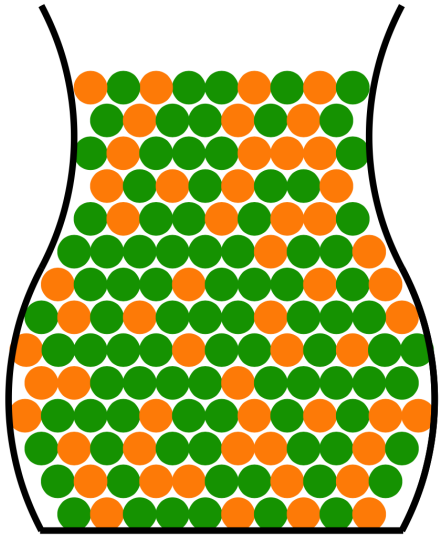
Summary

- We cannot do SVI on the collapsed VI bound.
- But we can refactorize it.
- SVI lets us handle **big data** *and* can attain the same optimum parameters.

PAC-Bayes

PAC-Bayes

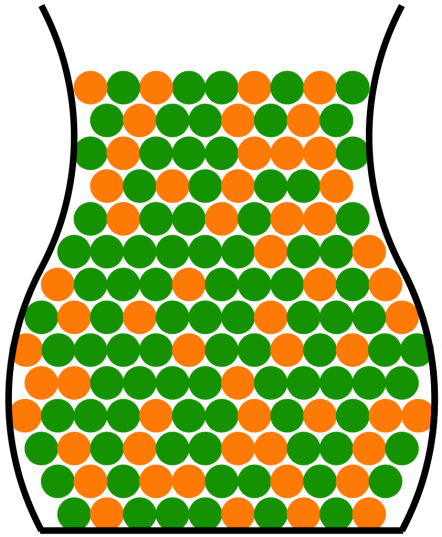
Hoeffding's Inequality



- How can you infer the probability of orange balls?

[3] Thanks to Hsuan-Tien Lin in National Taiwan University for his example.

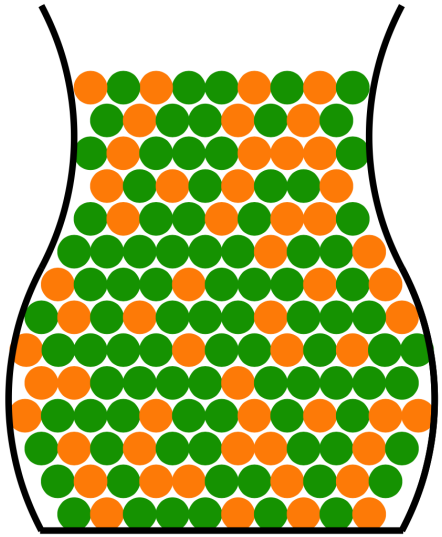
Hoeffding's Inequality



- How can you infer the probability of orange balls? **Sampling!!!**

[3] Thanks to Hsuan-Tien Lin in National Taiwan University for his example.

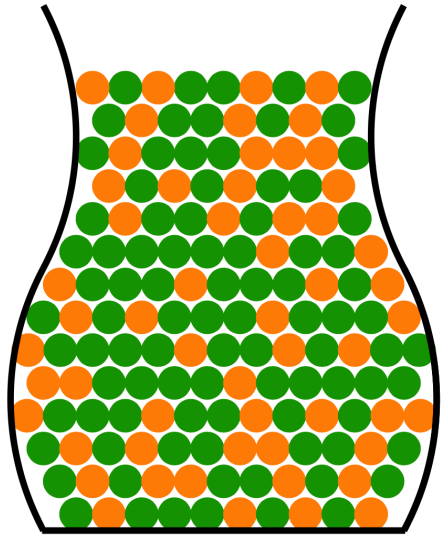
Hoeffding's Inequality



- How can you infer the probability of orange balls? **Sampling!!!**
- Assume the real orange probability is μ, the number of balls sampled is N, the sampling orange probability is ν.

[3] Thanks to Hsuan-Tien Lin in National Taiwan University for his example.

Hoeffding's Inequality



- How can you infer the probability of orange balls? **Sampling!!!**
- Assume the real orange probability is μ, the number of balls sampled is N, the sampling orange probability is ν.
  How accurate is your estimation by sampling?

3

[3] Thanks to Hsuan-Tien Lin in National Taiwan University for his example.

Hoeffding's Inequality

- Hoeffding's Inequality

$$p\left[|\nu - \mu| \geq \epsilon\right] \leq 2\exp(-2\epsilon^2 N) \tag{5}$$

Hoeffding's Inequality

- Hoeffding's Inequality

$$p\left[|\nu - \mu| \geq \epsilon\right] \leq 2\exp(-2\epsilon^2 N) \tag{5}$$

- The statement $\nu = \mu$ is probably approximately correct (PAC)

Hoeffding's Inequality

- Hoeffding's Inequality

$$p\left[|\nu - \mu| \geq \epsilon\right] \leq 2\exp(-2\epsilon^2 N) \tag{5}$$

- The statement $\nu = \mu$ is probably approximately correct (PAC)
- When the number of samples is big, your estimation can be very accurate. Therefore,

you can LEARN from training set! bigger, better!

Hoeffding's Inequality

- Moving on, supposing we have K hypothesis, whose generalization errors are $\{e_i\}_{i=1}^K$, whose empirical errors are $\{\hat{e}_i\}_{i=1}^K$, the probability for discrepency ϵ

$$p(\exists i, |\hat{e}_i - e_i| \geq \epsilon) \leq \sum_i \mathbb{P}(|\hat{e}_i - e_i| \geq \epsilon) \leq 2K \exp(-\epsilon^2 N) \tag{6}$$

Hoeffding's Inequality

- Moving on, supposing we have K hypothesis, whose generalization errors are $\{e_i\}_{i=1}^K$, whose empirical errors are $\{\hat{e}_i\}_{i=1}^K$, the probability for discrepency ϵ

$$p(\exists i, |\hat{e}_i - e_i| \geq \epsilon) \leq \sum_i \mathbb{P}(|\hat{e}_i - e_i| \geq \epsilon) \leq 2K \exp(-\epsilon^2 N) \tag{6}$$

- Equivently, we have

$$\log p(\exists i, |\hat{e}_i - e_i| \geq \sqrt{\frac{\log(2K) - \log(\delta)}{N}}) \leq \delta \tag{7}$$

Gibbs Classifier Bayes Classifier

- Suppose we're doing a binary classification task. Let $x \in X$ and $t \in \{-1, 1\}$.
- The model is composed of a latent function $\mathbf{x} \rightarrow \mathbf{y}(\mathbf{x})$, and a classification model $P(t|\mathbf{y})$.

Gibbs Classifier Bayes Classifier

- Suppose we're doing a binary classification task. Let $x \in X$ and $t \in \{-1, 1\}$.
- The model is composed of a latent function $\mathbf{x} \rightarrow \mathbf{y}(\mathbf{x})$, and a classification model $P(t|\mathbf{y})$.
- From the Bayesian viewpoint, the latent function could be parametrized as $\mathbf{y}(\mathbf{x}|\mathbf{w})$, where $\mathbf{w} \sim Q(\mathbf{w})$

Gibbs Classifier Bayes Classifier

- **Gibbs Classifier** predicts the output by first sampling $\mathbf{w} \sim Q(\mathbf{w})$, then returning

$$t_* = \text{sgn}\mathbf{y}(\mathbf{x}_*|\mathbf{w}) \tag{8}$$

Gibbs Classifier Bayes Classifier

- **Gibbs Classifier** predicts the output by first sampling $\mathbf{w} \sim Q(\mathbf{w})$, then returning

$$t_* = \text{sgn}\mathbf{y}(\mathbf{x}_*|\mathbf{w}) \tag{8}$$

- **Bayes classifier** predicts the output by integrating the distribution of $\mathbf{w}$, namely

$$t_* = \text{sgn}\mathbb{E}_{\mathbf{w}\sim Q}[\mathbf{y}(\mathbf{x}_*|\mathbf{w})] \tag{9}$$

Gibbs Classifier Bayes Classifier

- **Gibbs Classifier** predicts the output by first sampling $\mathbf{w} \sim Q(\mathbf{w})$, then returning

$$t_* = \text{sgn}\mathbf{y}(\mathbf{x}_*|\mathbf{w}) \tag{8}$$

- **Bayes classifier** predicts the output by integrating the distribution of $\mathbf{w}$, namely

$$t_* = \text{sgn}\mathbb{E}_{\mathbf{w}\sim Q}[\mathbf{y}(\mathbf{x}_*|\mathbf{w})] \tag{9}$$

- **Bayes voting classifier** predicts the output by integrating the distribution of $\mathbf{w}$ and votes, namely

$$t_* = \text{sgn}\mathbb{E}_{\mathbf{w}\sim Q}[\text{sgn}\mathbf{y}(\mathbf{x}_*|\mathbf{w})] \tag{10}$$

PAC-Bayesian theorem (McAllester, 1999)

[4] For any data distribution over $X \times \{-1, +1\}$ and an arbitrary posterior distribution $Q(\mathbf{w})$, we have that the following bound holds, where the probability is over random i.i.d. samples of size n
$S = \{(x_i^S, t_i^S) | i = 1, \cdots, n\}$ drawn from the true data distribution:

$$p\left[gen(Q) > emp(S, Q) + f(KL[Q||P], n, \delta, emp(S, Q)\right] \leq \delta \tag{11}$$

[4] PAC-Bayesian Generalisation Error Bounds for Gaussian Process Classification

PAC-Bayesian theorem (McAllester, 1999)

[4] For any data distribution over $X \times \{-1, +1\}$ and an arbitrary posterior distribution $Q(\mathbf{w})$, we have that the following bound holds, where the probability is over random i.i.d. samples of size n
$S = \{(x_i^S, t_i^S) | i = 1, \cdots, n\}$ drawn from the true data distribution:

$$p\left[gen(Q) > emp(S, Q) + f(KL[Q||P], n, \delta, emp(S, Q)\right] \leq \delta \tag{11}$$

[4] PAC-Bayesian Generalisation Error Bounds for Gaussian Process Classification

Variational Inference and PAC-Bayesian

- Recall variational inference, which training Evidence Lower Bound(ELBO) to optimize the variational posterior.

$$\text{ELBO} = \mathbb{E}_{Q(z|x)}[\log p(\mathbf{x}|\mathbf{z})] - KL[Q(z|x)||P(z)] \tag{12}$$

- With the reconstruction error, empirical loss tends to be small.
- With the regularization term, KL divergence cannot be very big.

$$\Downarrow$$

VI tends to have good generalization.

Gaussian Process Classification

- Given dataset $S = \{(x_i, t_i)\}_{i=1}^N$, a Gaussian Process models the dependency,

$$\begin{aligned} p(\mathbf{y}) &\sim \mathcal{N}(0, \mathbf{K}) \\ p(\mathbf{t}|\mathbf{y}) &= \prod_i p(t_i|y_i) \end{aligned} \tag{13}$$

- According to Bayes formula, the true posterior is as

$$p(\mathbf{y}|S) \propto p(\mathbf{t}|\mathbf{y})p(\mathbf{y}) \tag{14}$$

- Different with GP regression, classification posterior is intractable, which promotes using variational posterior $q(\mathbf{y})$ (Laplace approximation, for example).

$$q(\mathbf{y}) = \mathcal{N}(\mathbf{K}^{-1}\alpha, \Sigma) \tag{15}$$

Gaussian Process Classification

- Given $q(\mathbf{y}|S)$, predication for testing data $x_\star$? Assume $\mathbf{k} = k(x_\star, \mathbf{X})$, $k_\star = k(x_\star, x_\star)$.

$$\begin{aligned} p(y_\star, \mathbf{y}|S) &= p(y_\star|\mathbf{y})\, q(\mathbf{y}|S) \\ p(y_\star|\mathbf{y}) &= \mathcal{N}\left(\mathbf{k}^T\mathbf{K}^{-1}\mathbf{y}, k_\star - \mathbf{k}^T\mathbf{K}^{-1}\mathbf{k}\right) \end{aligned} \tag{16}$$

- Using conditional Gaussian distribution, we have the predicative distribution

$$q(y_\star|\mathbf{y}, S) = \mathcal{N}\left(\mathbf{k}^T\alpha, k_\star - \mathbf{k}^T\left(\mathbf{K}^{-1} - \mathbf{K}^{-1}\Sigma\mathbf{K}^{-1}\right)\mathbf{k}\right) \tag{17}$$

- KL divergence between $q(\mathbf{y})$ and $p(\mathbf{y})$ gives PAC bound for GP binary classification.

What can PAC do?

- Model Selection. (Not accurate, only as a reference)

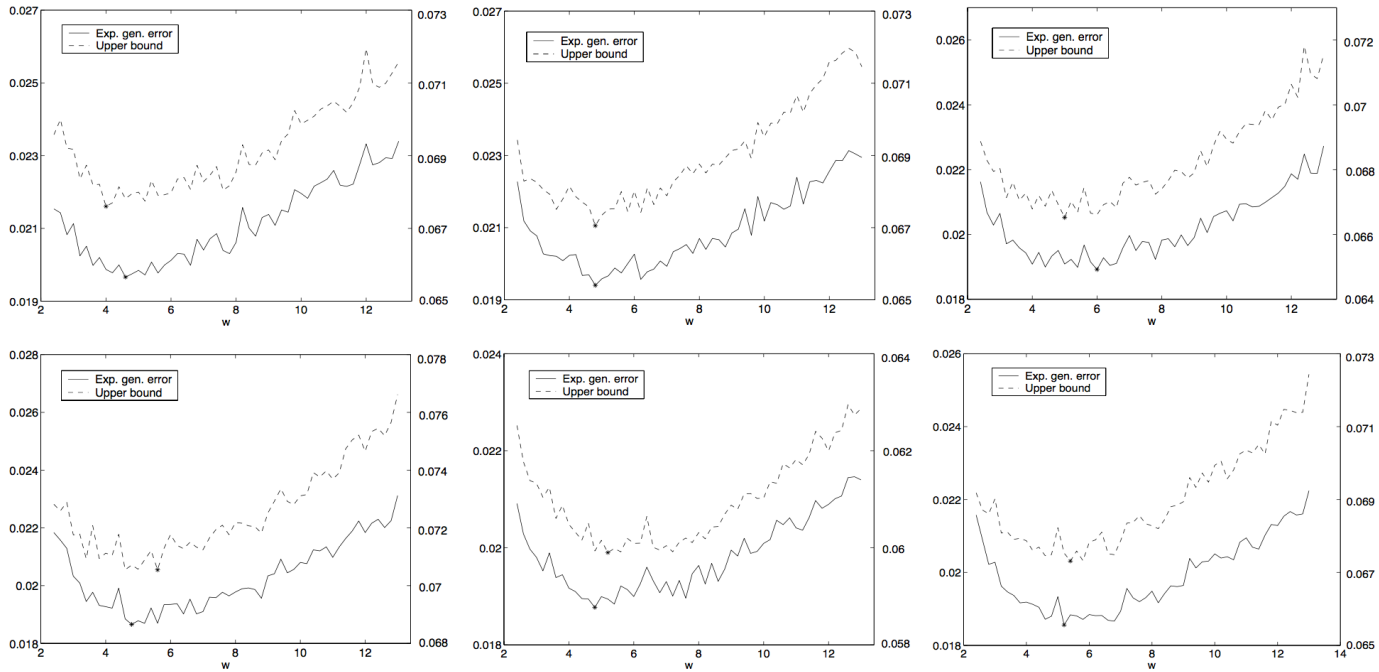


Figure 2: Comparing upper bound values with expected test errors. Solid line: expected test error (scale on left side). Dashed line: upper bound value (*translated*, scale on the right). Respective minimum points marked by an asterisk.