PAC-Bayes Analysis: Links to Luckiness and Applications

John Shawe-Taylor
University College London

PAC-Bayes Workshop
March, 2010

Including joint work with Amiran Ambroladze and Emilio Parrado-Hernández, Cédric Archambeau, Matthew Higgs, Manfred Opper

1. Luckiness
   - Definitions
   - Priors from data distributions
2. Maximum entropy classification
   - Generalisation
   - Optimisation
3. GPs and SDEs
   - Gaussian Process regression
   - Variational approximation
   - Generalisation

Luckiness definitions

- Based on a function from samples and hypotheses:

$$L : X^m \times H \to \mathbb{R}^+,$$

which measures the luckiness of a particular hypothesis with respect to the training examples.

- The level of luckiness is measured by seeing the number of functions that are luckier:

$$\ell(\mathbf{x}, h) = |\{b \in \{0,1\}^m : \exists g \in H, g(\mathbf{x}) = b, L(\mathbf{x}, g) \geq L(\mathbf{x}, h)\}|.$$

Luckiness definitions

- Based on a function from samples and hypotheses:

$$L : X^m \times H \to \mathbb{R}^+,$$

  which measures the luckiness of a particular hypothesis with respect to the training examples.
- The level of luckiness is measured by seeing the number of functions that are luckier:

$$\ell(\mathbf{x}, h) = |\{b \in \{0,1\}^m : \exists g \in H, g(\mathbf{x}) = b, L(\mathbf{x}, g) \geq L(\mathbf{x}, h)\}|.$$

Example

- Motivating example was the case of large margin classifiers: luckiness measured margin on the sample
- Note that hyperplanes as classifiers cannot be ranked by margin until sample is seen
- Can overcome this difficulty if we consider real valued functions, require outputs to be ± 1 and measure complexity by the norm of the weight vector
- This obscures the role of luckiness to capture alignment of hypotheses with the data generating distribution: eg density of distribution close to hyperplane
- Similar to idea of compatibility of Blum and Balcan (2005)
- Related to the local PAC-Bayes analysis of Catoni

Example

- Motivating example was the case of large margin classifiers: luckiness measured margin on the sample
- Note that hyperplanes as classifiers cannot be ranked by margin until sample is seen
- Can overcome this difficulty if we consider real valued functions, require outputs to be ± 1 and measure complexity by the norm of the weight vector
- This obscures the role of luckiness to capture alignment of hypotheses with the data generating distribution: eg density of distribution close to hyperplane
- Similar to idea of compatibility of Blum and Balcan (2005)
- Related to the local PAC-Bayes analysis of Catoni

Example

- Motivating example was the case of large margin classifiers: luckiness measured margin on the sample
- Note that hyperplanes as classifiers cannot be ranked by margin until sample is seen
- Can overcome this difficulty if we consider real valued functions, require outputs to be ± 1 and measure complexity by the norm of the weight vector
- This obscures the role of luckiness to capture alignment of hypotheses with the data generating distribution: eg density of distribution close to hyperplane
- Similar to idea of compatibility of Blum and Balcan (2005)
- Related to the local PAC-Bayes analysis of Catoni

Example

- Motivating example was the case of large margin classifiers: luckiness measured margin on the sample
- Note that hyperplanes as classifiers cannot be ranked by margin until sample is seen
- Can overcome this difficulty if we consider real valued functions, require outputs to be ± 1 and measure complexity by the norm of the weight vector
- This obscures the role of luckiness to capture alignment of hypotheses with the data generating distribution: eg density of distribution close to hyperplane
- Similar to idea of compatibility of Blum and Balcan (2005)
- Related to the local PAC-Bayes analysis of Catoni

Example

- Motivating example was the case of large margin classifiers: luckiness measured margin on the sample
- Note that hyperplanes as classifiers cannot be ranked by margin until sample is seen
- Can overcome this difficulty if we consider real valued functions, require outputs to be ± 1 and measure complexity by the norm of the weight vector
- This obscures the role of luckiness to capture alignment of hypotheses with the data generating distribution: eg density of distribution close to hyperplane
- Similar to idea of compatibility of Blum and Balcan (2005)
- Related to the local PAC-Bayes analysis of Catoni

Example

- Motivating example was the case of large margin classifiers: luckiness measured margin on the sample
- Note that hyperplanes as classifiers cannot be ranked by margin until sample is seen
- Can overcome this difficulty if we consider real valued functions, require outputs to be ± 1 and measure complexity by the norm of the weight vector
- This obscures the role of luckiness to capture alignment of hypotheses with the data generating distribution: eg density of distribution close to hyperplane
- Similar to idea of compatibility of Blum and Balcan (2005)
- Related to the local PAC-Bayes analysis of Catoni

Defining priors from data distributions

- Can use part of the data to learn the prior: eg train svm on half the data and centre the prior gaussian on this weight vector (Emilio will give results for this)
- Can use some expectation over the true distribution to define the centre of the prior distribution such as

$$\mathbf{w} = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[y\phi(\mathbf{x})]$$

- or more sophisticated:

$$\mathbf{w} = \mathbb{E}_{S\sim\mathcal{D}^{m_0}}[\mathbf{w}_{\text{SVM}}(S)]$$

with $m_0 << m$

Defining priors from data distributions

- Can use part of the data to learn the prior: eg train svm on half the data and centre the prior gaussian on this weight vector (Emilio will give results for this)
- Can use some expectation over the true distribution to define the centre of the prior distribution such as

$$\mathbf{w} = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[y\phi(\mathbf{x})]$$

- or more sophisticated:

$$\mathbf{w} = \mathbb{E}_{S\sim\mathcal{D}^{m_0}}[\mathbf{w}_{\mathrm{SVM}}(S)]$$

with $m_0 << m$

Defining priors from data distributions

- Can use part of the data to learn the prior: eg train svm on half the data and centre the prior gaussian on this weight vector (Emilio will give results for this)
- Can use some expectation over the true distribution to define the centre of the prior distribution such as

$$\mathbf{w} = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[y\phi(\mathbf{x})]$$

- or more sophisticated:

$$\mathbf{w} = \mathbb{E}_{S\sim\mathcal{D}^{m_0}}[\mathbf{w}_{\mathrm{SVM}}(S)]$$

with $m_0 << m$

Defining priors from data distributions

- In the latter cases use empirical versions for actual prior
  - bound the difference between this and true (data distribution) prior
  - use this to upper bound KL between true prior and posterior
  - complexity term typically decays with increasing sample size
  - Shiliang will present theory and Emilio empirical results
- Can also define prior based on true risk (or expectation over SVM weight vectors on samples of training set size m
  - Even tighter bounds presented by Guy with extensions to manifold learning
  - Closer to Catoni's approach

Defining priors from data distributions

- In the latter cases use empirical versions for actual prior
  - bound the difference between this and true (data distribution) prior
  - use this to upper bound KL between true prior and posterior
  - complexity term typically decays with increasing sample size
  - Shiliang will present theory and Emilio empirical results
- Can also define prior based on true risk (or expectation over SVM weight vectors on samples of training set size m
  - Even tighter bounds presented by Guy with extensions to manifold learning
  - Closer to Catoni's approach

Defining priors from data distributions

- In the latter cases use empirical versions for actual prior
  - bound the difference between this and true (data distribution) prior
  - use this to upper bound KL between true prior and posterior
  - complexity term typically decays with increasing sample size
  - Shiliang will present theory and Emilio empirical results
- Can also define prior based on true risk (or expectation over SVM weight vectors on samples of training set size m
  - Even tighter bounds presented by Guy with extensions to manifold learning
  - Closer to Catoni's approach

Defining priors from data distributions

- In the latter cases use empirical versions for actual prior
  - bound the difference between this and true (data distribution) prior
  - use this to upper bound KL between true prior and posterior
  - complexity term typically decays with increasing sample size
  - Shiliang will present theory and Emilio empirical results
- Can also define prior based on true risk (or expectation over SVM weight vectors on samples of training set size m
  - Even tighter bounds presented by Guy with extensions to manifold learning
  - Closer to Catoni's approach

Defining priors from data distributions

- In the latter cases use empirical versions for actual prior
  - bound the difference between this and true (data distribution) prior
  - use this to upper bound KL between true prior and posterior
  - complexity term typically decays with increasing sample size
  - Shiliang will present theory and Emilio empirical results
- Can also define prior based on true risk (or expectation over SVM weight vectors on samples of training set size m
  - Even tighter bounds presented by Guy with extensions to manifold learning
  - Closer to Catoni's approach

Defining priors from data distributions

- In the latter cases use empirical versions for actual prior
  - bound the difference between this and true (data distribution) prior
  - use this to upper bound KL between true prior and posterior
  - complexity term typically decays with increasing sample size
  - Shiliang will present theory and Emilio empirical results
- Can also define prior based on true risk (or expectation over SVM weight vectors on samples of training set size m
  - Even tighter bounds presented by Guy with extensions to manifold learning
  - Closer to Catoni's approach

Defining priors from data distributions

- In the latter cases use empirical versions for actual prior
  - bound the difference between this and true (data distribution) prior
  - use this to upper bound KL between true prior and posterior
  - complexity term typically decays with increasing sample size
  - Shiliang will present theory and Emilio empirical results
- Can also define prior based on true risk (or expectation over SVM weight vectors on samples of training set size m
  - Even tighter bounds presented by Guy with extensions to manifold learning
  - Closer to Catoni's approach

Defining priors from data distributions

- In the latter cases use empirical versions for actual prior
  - bound the difference between this and true (data distribution) prior
  - use this to upper bound KL between true prior and posterior
  - complexity term typically decays with increasing sample size
  - Shiliang will present theory and Emilio empirical results
- Can also define prior based on true risk (or expectation over SVM weight vectors on samples of training set size m
  - Even tighter bounds presented by Guy with extensions to manifold learning
  - Closer to Catoni's approach

Maximum entropy learning

- consider function class for $\mathcal{X}$ is a subset of the ℓ_∞ unit ball

$$\mathcal{F} = \left\{ f_{\boldsymbol{w}} : \boldsymbol{x} \in \mathcal{X} \mapsto \operatorname{sgn}\left(\sum_{i=1}^{N} w_i x_i \right) : \|\boldsymbol{w}\|_1 \leq 1 \right\},$$

- want posterior distribution $Q(\boldsymbol{w})$ such that can bound

$$P_{(\boldsymbol{x},y)\sim\mathcal{D}}(f_{\boldsymbol{w}}(\boldsymbol{x}) \neq y) \leq 2e_{Q(\boldsymbol{w})}(= 2Q_{\mathcal{D}}(\boldsymbol{w})) = 2\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D},q\sim Q(\boldsymbol{w})}\left[I\left[q(\boldsymbol{x}) \neq y\right]\right]$$

- Given a training sample $S = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)\}$, we similarly define

$$\hat{e}_{Q(\boldsymbol{w})}(= \hat{Q}_S(\boldsymbol{w})) = \frac{1}{m}\sum_{i=1}^{m} \mathbb{E}_{q\sim Q(\boldsymbol{w})}\left[I\left[q(\boldsymbol{x}_i) \neq y_i\right]\right].$$

Maximum entropy learning

- consider function class for $\mathcal{X}$ is a subset of the ℓ_∞ unit ball

$$\mathcal{F} = \left\{ f_{\boldsymbol{w}} : \boldsymbol{x} \in \mathcal{X} \mapsto \operatorname{sgn}\left(\sum_{i=1}^{N} w_i x_i \right) : \|\boldsymbol{w}\|_1 \leq 1 \right\},$$

- want posterior distribution $Q(\boldsymbol{w})$ such that can bound

$$P_{(\boldsymbol{x},y)\sim\mathcal{D}}(f_{\boldsymbol{w}}(\boldsymbol{x}) \neq y) \leq 2e_{Q(\boldsymbol{w})}(= 2Q_{\mathcal{D}}(\boldsymbol{w})) = 2\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D},q\sim Q(\boldsymbol{w})}\left[I\left[q(\boldsymbol{x}) \neq y\right]\right]$$

- Given a training sample $S = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)\}$, we similarly define

$$\hat{e}_{Q(\boldsymbol{w})}(= \hat{Q}_S(\boldsymbol{w})) = \frac{1}{m}\sum_{i=1}^{m} \mathbb{E}_{q\sim Q(\boldsymbol{w})}\left[I\left[q(\boldsymbol{x}_i) \neq y_i\right]\right].$$

Maximum entropy learning

- consider function class for $\mathcal{X}$ is a subset of the ℓ_∞ unit ball

$$\mathcal{F} = \left\{ f_{\boldsymbol{w}} : \boldsymbol{x} \in \mathcal{X} \mapsto \operatorname{sgn}\left(\sum_{i=1}^{N} w_i x_i \right) : \|\boldsymbol{w}\|_1 \leq 1 \right\},$$

- want posterior distribution $Q(\boldsymbol{w})$ such that can bound

$$P_{(\boldsymbol{x},y)\sim\mathcal{D}}(f_{\boldsymbol{w}}(\boldsymbol{x}) \neq y) \leq 2e_{Q(\boldsymbol{w})}(= 2Q_{\mathcal{D}}(\boldsymbol{w})) = 2\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D},q\sim Q(\boldsymbol{w})}\left[I\left[q(\boldsymbol{x}) \neq y\right]\right]$$

- Given a training sample $S = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)\}$, we similarly define

$$\hat{e}_{Q(\boldsymbol{w})}(= \hat{Q}_S(\boldsymbol{w})) = \frac{1}{m}\sum_{i=1}^{m} \mathbb{E}_{q\sim Q(\boldsymbol{w})}\left[I\left[q(\boldsymbol{x}_i) \neq y_i\right]\right].$$

Posterior distribution $Q(\boldsymbol{w})$

- Classifier q involves random weight vector $W \in \mathbb{R}^N$ plus random threshold Θ

$$q_{W,\Theta}(\boldsymbol{x}) = \operatorname{sgn}\left(\langle W, \boldsymbol{x}\rangle - \Theta\right).$$

- The distribution $Q(\boldsymbol{w})$ of W will be discrete with

$$W = \operatorname{sgn}(w_i)\boldsymbol{e}_i; \text{ with probability } |w_i|, i = 1, \ldots, N,$$

where $\boldsymbol{e}_i$ is the unit vector. The distribution of Θ is uniform on the interval $[-1, 1]$.

Posterior distribution $Q(\boldsymbol{w})$

- Classifier q involves random weight vector $W \in \mathbb{R}^N$ plus random threshold Θ

$$q_{W,\Theta}(\boldsymbol{x}) = \operatorname{sgn}\left(\langle W, \boldsymbol{x}\rangle - \Theta\right).$$

- The distribution $Q(\boldsymbol{w})$ of W will be discrete with

$$W = \operatorname{sgn}(w_i)\boldsymbol{e}_i; \text{ with probability } |w_i|, i = 1, \ldots, N,$$

where $\boldsymbol{e}_i$ is the unit vector. The distribution of Θ is uniform on the interval $[-1, 1]$.

Error expression

Proposition

With the above definitions, we have for $\boldsymbol{w}$ *satisfying* $\|\boldsymbol{w}\|_1 = 1$, *that for any* $(\boldsymbol{x}, y) \in \mathcal{X} \times \{-1, +1\}$,

$$P_{q \sim Q(\boldsymbol{w})}(q(\boldsymbol{x}) \neq y) = 0.5(1 - y\langle \boldsymbol{w}, \boldsymbol{x} \rangle).$$

Error expression proof

Proof.

$$
\begin{aligned}
P_{q\sim Q(\boldsymbol{w})}(q(\boldsymbol{x}) \neq y) &= \sum_{i=1}^{N} |w_i| P_{\Theta}\left(\operatorname{sgn}\left(\operatorname{sgn}(w_i)\langle \boldsymbol{e}_i, \boldsymbol{x}\rangle - \Theta\right) \neq y\right) \\
&= \sum_{i=1}^{N} |w_i| P_{\Theta}\left(\operatorname{sgn}\left(\operatorname{sgn}(w_i) x_i - \Theta\right) \neq y\right) \\
&= 0.5 \sum_{i=1}^{N} |w_i| (1 - y \operatorname{sgn}(w_i) x_i) \\
&= 0.5(1 - y\langle \boldsymbol{w}, \boldsymbol{x}\rangle),
\end{aligned}
$$

□

Generalisation error

Corollary

$$P_{(\boldsymbol{x},y)\sim\mathcal{D}}\left(f_{\boldsymbol{w}}(\boldsymbol{x}) \neq y\right) \leq 2e_{Q(\boldsymbol{w})}.$$

Proof.

$$\begin{aligned} P_{q\sim Q(\boldsymbol{w})}(q(\boldsymbol{x}) \neq y) &\geq 0.5 \\ &\Leftrightarrow \\ f_{\boldsymbol{w}}(\boldsymbol{x}) &\neq y. \end{aligned}$$

□

Base result

Theorem

With probability at least $1-\delta$ *over the draw of training sets of size* m

$$\mathrm{KL}(\hat{e}_{Q(\boldsymbol{w})} \| e_{Q(\boldsymbol{w})}) \leq \frac{\sum_{i=1}^{N} |w_i| \ln |w_i| + \ln(2N) + \ln((m+1)/\delta)}{m}$$

Proof.

- Use prior P uniform on unit vectors $\pm \boldsymbol{e}_i$.
- Posterior described above so $\mathrm{KL}(P \| Q(\boldsymbol{w}))$ equals $\ln(2N) -$ entropy of $\boldsymbol{w}$.

Base result

Theorem

With probability at least $1-\delta$ *over the draw of training sets of size* m

$$\mathrm{KL}(\hat{e}_{Q(\boldsymbol{w})}\|e_{Q(\boldsymbol{w})}) \leq \frac{\sum_{i=1}^{N}|w_i|\ln|w_i| + \ln(2N) + \ln((m+1)/\delta)}{m}$$

Proof.

- Use prior P uniform on unit vectors $\pm\boldsymbol{e}_i$.
- Posterior described above so $\mathrm{KL}(P\|Q(\boldsymbol{w}))$ equals $\ln(2N)-$ entropy of $\boldsymbol{w}$.

Base result

Theorem

With probability at least $1-\delta$ *over the draw of training sets of size* m

$$\mathrm{KL}(\hat{e}_{Q(\boldsymbol{w})} \| e_{Q(\boldsymbol{w})}) \leq \frac{\sum_{i=1}^{N} |w_i| \ln |w_i| + \ln(2N) + \ln((m+1)/\delta)}{m}$$

Proof.

- Use prior P uniform on unit vectors $\pm \boldsymbol{e}_i$.
- Posterior described above so $\mathrm{KL}(P \| Q(\boldsymbol{w}))$ equals $\ln(2N) -$ entropy of $\boldsymbol{w}$.

Interpretation

- Suggests maximising the entropy as a means of minimising the bound.
- Problem that empirical error $\hat{e}_{Q(\boldsymbol{w})}$ is too large:

$$\hat{e}_{Q(\boldsymbol{w})} = \sum_{i=1}^{m} 0.5(1 - y_i \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle)$$

- Function of margin – but just linear function.

Interpretation

- Suggests maximising the entropy as a means of minimising the bound.
- Problem that empirical error $\hat{e}_{Q(\boldsymbol{w})}$ is too large:

$$\hat{e}_{Q(\boldsymbol{w})} = \sum_{i=1}^{m} 0.5(1 - y_i \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle)$$

- Function of margin – but just linear function.

Interpretation

- Suggests maximising the entropy as a means of minimising the bound.
- Problem that empirical error $\hat{e}_{Q(\boldsymbol{w})}$ is too large:

$$\hat{e}_{Q(\boldsymbol{w})} = \sum_{i=1}^{m} 0.5(1 - y_i \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle)$$

- Function of margin – but just linear function.

Boosting the bound

- Trick to boost the power of the bound is to take T independent samples of the distribution $Q(\boldsymbol{w})$ and vote for the classification:

$$q_{\boldsymbol{W},\Theta}(\boldsymbol{x}) = \operatorname{sgn}\left(\sum_{i=1}^{T} \operatorname{sgn}\left(\langle W^t, \boldsymbol{x}\rangle - \Theta^t\right)\right),$$

- Now empirical error becomes

$$\hat{e}_{Q(\boldsymbol{w})} = \frac{0.5^T}{m} \sum_{i=1}^{m} \sum_{t=0}^{\lfloor T/2 \rfloor} \binom{T}{t} (1 + y_i\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle)^t (1 - y_i\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle)^{T-t},$$

giving sigmoid like loss as function of the margin.

Boosting the bound

- Trick to boost the power of the bound is to take T independent samples of the distribution $Q(\boldsymbol{w})$ and vote for the classification:

$$q_{\boldsymbol{w},\Theta}(\boldsymbol{x}) = \operatorname{sgn}\left(\sum_{i=1}^{T} \operatorname{sgn}\left(\langle W^t, \boldsymbol{x}\rangle - \Theta^t\right)\right),$$

- Now empirical error becomes

$$\hat{e}_{Q(\boldsymbol{w})} = \frac{0.5^T}{m} \sum_{i=1}^{m} \sum_{t=0}^{\lfloor T/2 \rfloor} \binom{T}{t} \left(1 + y_i \langle \boldsymbol{w}, \boldsymbol{x}_i\rangle\right)^t \left(1 - y_i \langle \boldsymbol{w}, \boldsymbol{x}_i\rangle\right)^{T-t},$$

giving sigmoid like loss as function of the margin.

Full result

Theorem

With probability at least $1-\delta$ *over the draw of training sets of size* m

$$P_{(\boldsymbol{x},y)\sim\mathcal{D}}\left(f_{\boldsymbol{w}}(\boldsymbol{x})\neq y\right)\leq$$
$$2\mathrm{KL}^{-1}\left(\hat{e}_{Q^T(\boldsymbol{w})},\frac{T\sum_{i=1}^{N}|w_i|\ln(|w_i|)+T\ln(2N)+\ln((m+1)/\delta)}{m}\right),$$

- Note penalty factor of T applied to KL
- Behaves like the (inverse) margin in usual bounds

Full result

Theorem

With probability at least $1-\delta$ *over the draw of training sets of size* m

$$P_{(\boldsymbol{x},y)\sim\mathcal{D}}\left(f_{\boldsymbol{w}}(\boldsymbol{x})\neq y\right)\leq$$
$$2\mathrm{KL}^{-1}\left(\hat{e}_{Q^T(\boldsymbol{w})},\frac{T\sum_{i=1}^{N}|w_i|\ln(|w_i|)+T\ln(2N)+\ln((m+1)/\delta)}{m}\right),$$

- Note penalty factor of T applied to KL
- Behaves like the (inverse) margin in usual bounds

Full result

Theorem

With probability at least $1-\delta$ *over the draw of training sets of size* m

$$P_{(\boldsymbol{x},y)\sim\mathcal{D}}\left(f_{\boldsymbol{w}}(\boldsymbol{x})\neq y\right)\leq$$
$$2\mathrm{KL}^{-1}\left(\hat{e}_{Q^T(\boldsymbol{w})},\frac{T\sum_{i=1}^{N}|w_i|\ln(|w_i|)+T\ln(2N)+\ln((m+1)/\delta)}{m}\right),$$

- Note penalty factor of T applied to KL
- Behaves like the (inverse) margin in usual bounds

Algorithmics

- Bound motivates the optimisation:

$$\begin{aligned} \min_{\boldsymbol{w},\rho,\boldsymbol{\xi}} \quad & \sum_{j=1}^{N} |w_j| \ln |w_j| - C\rho + D\sum_{i=1}^{m} \xi_i \\ \text{subject to:} \quad & y_i \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle \geq \rho - \xi_i, 1 \leq i \leq m, \\ & \|\boldsymbol{w}\|_1 \leq 1, \xi_i \geq 0, 1 \leq i \leq m. \end{aligned}$$

- This follows the SVM route of approximating the sigmoid like loss by the (convex) hinge loss

Algorithmics

- Bound motivates the optimisation:

$$\begin{aligned} \min_{\boldsymbol{w},\rho,\boldsymbol{\xi}} \quad & \sum_{j=1}^{N} |w_j| \ln |w_j| - C\rho + D\sum_{i=1}^{m} \xi_i \\ \text{subject to:} \quad & y_i \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle \geq \rho - \xi_i, 1 \leq i \leq m, \\ & \|\boldsymbol{w}\|_1 \leq 1, \xi_i \geq 0, 1 \leq i \leq m. \end{aligned}$$

- This follows the SVM route of approximating the sigmoid like loss by the (convex) hinge loss

Dual optimisation

$$\max_{\alpha} \quad L = -\sum_{j=1}^{N} \exp\left(\left|\sum_{i=1}^{m} \alpha_i y_i x_{ij}\right| - 1 - \lambda\right) - \lambda$$

$$\text{subject to:} \quad \sum_{i=1}^{m} \alpha_i = C \quad 0 \leq \alpha_i \leq D, 1 \leq i \leq m.$$

- Similar to SVM but with exponential function
- Surprisingly also gives dual sparsity
- Coordinate wise descent works very well (cf SMO algorithm)

Dual optimisation

$$\max_{\alpha} \quad L = -\sum_{j=1}^{N} \exp\left(\left|\sum_{i=1}^{m} \alpha_i y_i x_{ij}\right| - 1 - \lambda\right) - \lambda$$

$$\text{subject to:} \quad \sum_{i=1}^{m} \alpha_i = C \quad 0 \leq \alpha_i \leq D, 1 \leq i \leq m.$$

- Similar to SVM but with exponential function
- Surprisingly also gives dual sparsity
- Coordinate wise descent works very well (cf SMO algorithm)

Dual optimisation

$$\max_{\alpha} \quad L = -\sum_{j=1}^{N} \exp\left(\left|\sum_{i=1}^{m} \alpha_i y_i x_{ij}\right| - 1 - \lambda\right) - \lambda$$

$$\text{subject to:} \quad \sum_{i=1}^{m} \alpha_i = C \quad 0 \leq \alpha_i \leq D, 1 \leq i \leq m.$$

- Similar to SVM but with exponential function
- Surprisingly also gives dual sparsity
- Coordinate wise descent works very well (cf SMO algorithm)

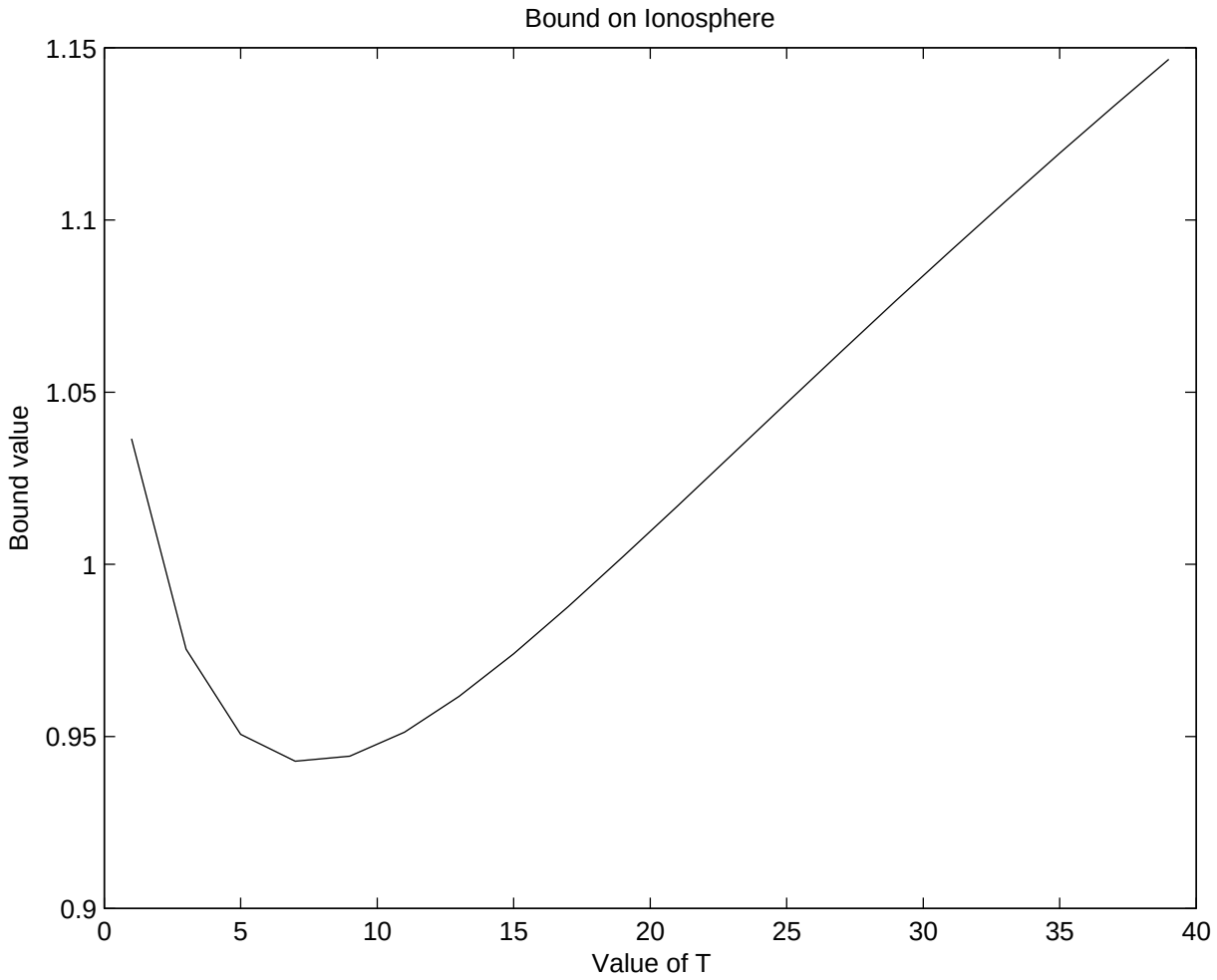
Dual optimisation

$$\max_{\alpha} \quad L = -\sum_{j=1}^{N} \exp\left(\left|\sum_{i=1}^{m} \alpha_i y_i x_{ij}\right| - 1 - \lambda\right) - \lambda$$

$$\text{subject to:} \quad \sum_{i=1}^{m} \alpha_i = C \quad 0 \leq \alpha_i \leq D, 1 \leq i \leq m.$$

- Similar to SVM but with exponential function
- Surprisingly also gives dual sparsity
- Coordinate wise descent works very well (cf SMO algorithm)

Results: effect of varying T



Results

Bound and test errors:

Data	Bound	Error	SVM error
Ionosphere	0.63	0.28	0.24
Votes	0.78	0.35	0.35
Glass	0.69	0.46	0.47
Haberman	0.64	0.25	0.26
Credit	0.60	0.25	0.28

Gaussian Process Regression

- GP is distribution over real valued functions that is multivariate Gaussian when restricted to any finite subset of inputs
- Characterised by a kernel that specifies the covariance function when marginalising on any finite subset
- If have finite set of input/output observations generated with additive Gaussian noise on the outputs, posterior is also Gaussian process
- KL divergence between prior and posterior can be computed as ($K = RR'$ is a Cholesky decomposition of K):

$$2\mathrm{KL}(Q\|P) = \log\det\left(I + \frac{1}{\sigma^2}K\right) - \mathrm{tr}\left(\left(\sigma^2 I + K\right)^{-1} K\right) + \left\|R(K + \sigma^2 I)^{-1}\boldsymbol{y}\right\|^2$$

Gaussian Process Regression

- GP is distribution over real valued functions that is multivariate Gaussian when restricted to any finite subset of inputs
- Characterised by a kernel that specifies the covariance function when marginalising on any finite subset
- If have finite set of input/output observations generated with additive Gaussian noise on the outputs, posterior is also Gaussian process
- KL divergence between prior and posterior can be computed as ($K = RR'$ is a Cholesky decomposition of K):

$$2\mathrm{KL}(Q\|P) = \log\det\left(I + \frac{1}{\sigma^2}K\right) - \mathrm{tr}\left(\left(\sigma^2 I + K\right)^{-1} K\right) + \left\|R(K + \sigma^2 I)^{-1}\boldsymbol{y}\right\|^2$$

Gaussian Process Regression

- GP is distribution over real valued functions that is multivariate Gaussian when restricted to any finite subset of inputs
- Characterised by a kernel that specifies the covariance function when marginalising on any finite subset
- If have finite set of input/output observations generated with additive Gaussian noise on the outputs, posterior is also Gaussian process
- KL divergence between prior and posterior can be computed as ($K = RR'$ is a Cholesky decomposition of K):

$$2\mathrm{KL}(Q\|P) = \log\det\left(I + \frac{1}{\sigma^2}K\right) - \mathrm{tr}\left(\left(\sigma^2 I + K\right)^{-1} K\right) + \left\|R(K + \sigma^2 I)^{-1}\boldsymbol{y}\right\|^2$$

Gaussian Process Regression

- GP is distribution over real valued functions that is multivariate Gaussian when restricted to any finite subset of inputs
- Characterised by a kernel that specifies the covariance function when marginalising on any finite subset
- If have finite set of input/output observations generated with additive Gaussian noise on the outputs, posterior is also Gaussian process
- KL divergence between prior and posterior can be computed as ($K = RR'$ is a Cholesky decomposition of K):

$$2\mathrm{KL}(Q\|P) = \log\det\left(I + \frac{1}{\sigma^2}K\right) - \mathrm{tr}\left(\left(\sigma^2 I + K\right)^{-1} K\right) + \left\|R(K + \sigma^2 I)^{-1}\boldsymbol{y}\right\|^2$$

Applying PAC-Bayes theorem

- Suggests can use the PB theorem if can create appropriate classifiers indexed by real value functions
- Consider for some $\epsilon > 0$ classifiers:

$$h_f^\epsilon(\boldsymbol{x}, y) = \begin{cases} 1; & \text{if } |y - f(\boldsymbol{x})| \leq \epsilon; \\ 0; & \text{otherwise.} \end{cases}$$

- Can compute expected value of h_f^ϵ under posterior function:

$$\mathbb{E}_{f\sim Q}\left[h_f^\epsilon(\boldsymbol{x}, y)\right] = \frac{1}{2}\text{erf}\left(\frac{y + \epsilon - m(\boldsymbol{x})}{\sqrt{2v(\boldsymbol{x})}}\right) - \frac{1}{2}\text{erf}\left(\frac{y - \epsilon - m(\boldsymbol{x})}{\sqrt{2v(\boldsymbol{x})}}\right)$$

Applying PAC-Bayes theorem

- Suggests can use the PB theorem if can create appropriate classifiers indexed by real value functions
- Consider for some $\epsilon > 0$ classifiers:

$$h_f^\epsilon(\boldsymbol{x}, y) = \begin{cases} 1; & \text{if } |y - f(\boldsymbol{x})| \leq \epsilon; \\ 0; & \text{otherwise.} \end{cases}$$

- Can compute expected value of h_f^ϵ under posterior function:

$$\mathbb{E}_{f\sim Q}\left[h_f^\epsilon(\boldsymbol{x}, y)\right] = \frac{1}{2}\text{erf}\left(\frac{y + \epsilon - m(\boldsymbol{x})}{\sqrt{2v(\boldsymbol{x})}}\right) - \frac{1}{2}\text{erf}\left(\frac{y - \epsilon - m(\boldsymbol{x})}{\sqrt{2v(\boldsymbol{x})}}\right)$$

Applying PAC-Bayes theorem

- Suggests can use the PB theorem if can create appropriate classifiers indexed by real value functions
- Consider for some $\epsilon > 0$ classifiers:

$$h_f^\epsilon(\boldsymbol{x}, y) = \begin{cases} 1; & \text{if } |y - f(\boldsymbol{x})| \leq \epsilon; \\ 0; & \text{otherwise.} \end{cases}$$

- Can compute expected value of h_f^ϵ under posterior function:

$$\mathbb{E}_{f\sim Q}\left[h_f^\epsilon(\boldsymbol{x}, y)\right] = \frac{1}{2}\text{erf}\left(\frac{y + \epsilon - m(\boldsymbol{x})}{\sqrt{2v(\boldsymbol{x})}}\right) - \frac{1}{2}\text{erf}\left(\frac{y - \epsilon - m(\boldsymbol{x})}{\sqrt{2v(\boldsymbol{x})}}\right)$$

GP Result

- Furthermore can lower bound expected value of point $(\boldsymbol{x}, y)$ in the posterior distribution by

$$2\epsilon\mathcal{N}(y|m(\boldsymbol{x}), v(\boldsymbol{x})) \geq \mathbb{E}_{f\sim Q}\left[h_f^\epsilon(\boldsymbol{x}, y)\right] - \sup_{\tau\in[\epsilon,\epsilon]} \frac{\epsilon^2}{2}\frac{2}{v(\boldsymbol{x})\sqrt{2e\pi}}.$$

enabling an application of the PB Theorem to give:

$$\mathbb{E}\left[\mathcal{N}(y|m(\boldsymbol{x}), v(\boldsymbol{x})) + \frac{\epsilon}{2v(\boldsymbol{x})\sqrt{2e\pi}}\right] \geq \frac{1}{2\epsilon}\mathrm{KL}^{-1}\left(E(\epsilon), \frac{D + \ln((m+1)/\delta)}{m}\right)$$

where $E(\epsilon)$ is the empirical average of $\mathbb{E}_{f\sim Q}\left[h_f^\epsilon(\boldsymbol{x}, y)\right]$ and D is the KL between prior and posterior.

GP Experimental Results

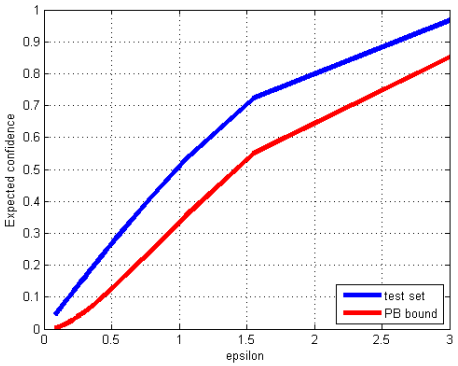
- The robot arm problem (R), 150 training points and 51 test points.
- The Boston housing problem (H), 455 training points and 51 test points.
- The forest fire problem (F), 450 training points 67 test points.

Dat	σ	$\hat{e}$	KL^{-1}	e_{test}	KL^{-1}	varGP e_{test}
R	0.0494	0.8903	0.4782	0.8419		
H	0.1924	0.8699	0.4645	0.7155	0.8401	0.9416
F	1.0129	0.5694	0.4557	0.5533		

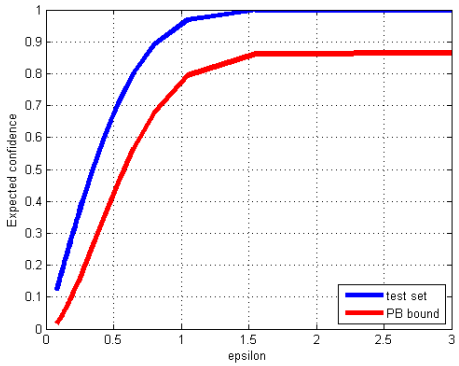
GP Experimental Results

- We can also plot the test accuracy and bound as a function of ϵ:

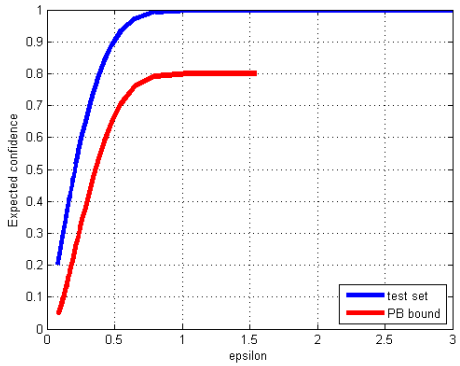
Figure: Gaussian noise: Plot of $\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[1-\alpha(\mathbf{x})]$ against ϵ with for varying noise level η.



(a) $\eta = 1$



(b) $\eta = 3$

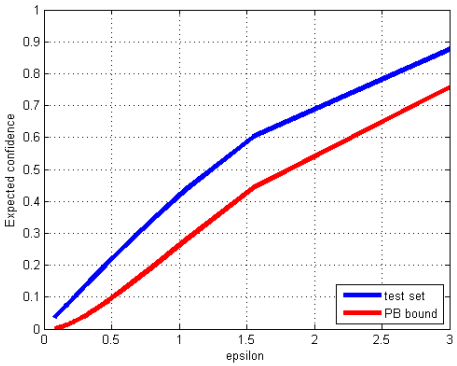


(c) $\eta = 5$

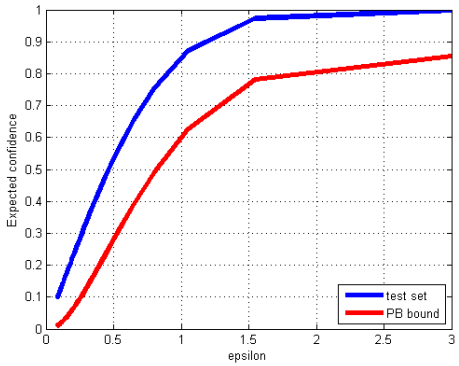
GP Experimental Results

- With Laplace noise:

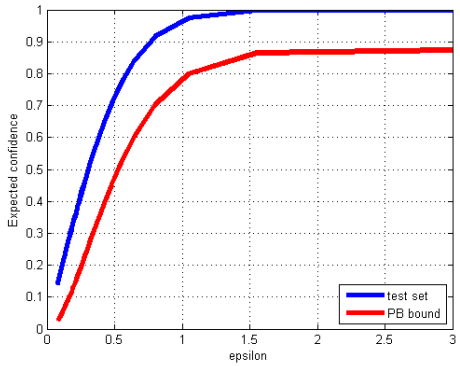
Figure: Laplace noise: Plot of $\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[1-\alpha(\mathbf{x})]$ against ϵ with for varying η.



(a) $\eta = 1$



(b) $\eta = 3$

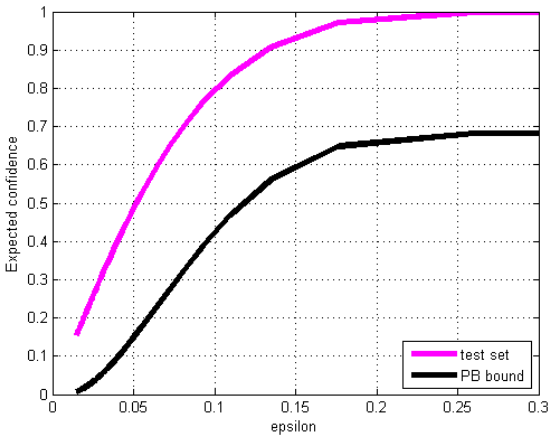


(c) $\eta = 5$

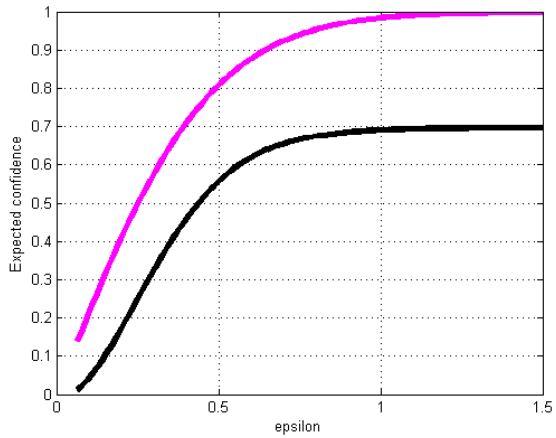
GP Experimental Results

- Robot arm problem and Boston Housing:

Figure: Confidence levels for Robot arm problem



(a) Robot arm



(b) Boston housing

Stochastic Differential Equation Models

- Consider modelling a time varying process with a (non-linear) stochastic differential equation:

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + \sqrt{\mathbf{\Sigma}}\, d\mathbf{W}$$

- $\mathbf{f}(\mathbf{x}, t)$ is a non-linear drift term and $d\mathbf{W}$ is a Wiener process
- This is the limit of the discrete time equation:

$$\Delta\mathbf{x}_k \equiv \mathbf{x}_{k+1} - \mathbf{x}_k = \mathbf{f}(\mathbf{x}_k)\Delta t + \sqrt{\Delta t\, \mathbf{\Sigma}}\, \epsilon_k \ .$$

where ϵ_k is zero mean, unit variance Gaussian noise.

Stochastic Differential Equation Models

- Consider modelling a time varying process with a (non-linear) stochastic differential equation:

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + \sqrt{\mathbf{\Sigma}}\, d\mathbf{W}$$

- $\mathbf{f}(\mathbf{x}, t)$ is a non-linear drift term and $d\mathbf{W}$ is a Wiener process
- This is the limit of the discrete time equation:

$$\Delta\mathbf{x}_k \equiv \mathbf{x}_{k+1} - \mathbf{x}_k = \mathbf{f}(\mathbf{x}_k)\Delta t + \sqrt{\Delta t\, \mathbf{\Sigma}}\, \epsilon_k \ .$$

where ϵ_k is zero mean, unit variance Gaussian noise.

Stochastic Differential Equation Models

- Consider modelling a time varying process with a (non-linear) stochastic differential equation:

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + \sqrt{\mathbf{\Sigma}}\, d\mathbf{W}$$

- $\mathbf{f}(\mathbf{x}, t)$ is a non-linear drift term and $d\mathbf{W}$ is a Wiener process
- This is the limit of the discrete time equation:

$$\Delta\mathbf{x}_k \equiv \mathbf{x}_{k+1} - \mathbf{x}_k = \mathbf{f}(\mathbf{x}_k)\Delta t + \sqrt{\Delta t\, \mathbf{\Sigma}}\, \epsilon_k \ .$$

where ϵ_k is zero mean, unit variance Gaussian noise.

Variational approximation

- We use the Bayesian approach to data modelling with a noise model given by:

$$p(\mathbf{y}_n|\mathbf{x}(t_n)) = \mathcal{N}(\mathbf{y}_n|\mathbf{H}\mathbf{x}(t_n), \mathbf{R}),$$

- We consider a variational approximation of the posterior using a time-varying linear SDE:

$$d\mathbf{x} = \mathbf{f}_L(\mathbf{x}, t)dt + \sqrt{\mathbf{\Sigma}}\, d\mathbf{W},$$

where

$$\mathbf{f}_L(\mathbf{x}, t) = -\mathbf{A}(t)\mathbf{x} + \mathbf{b}(t).$$

Variational approximation

- We use the Bayesian approach to data modelling with a noise model given by:

$$p(\mathbf{y}_n|\mathbf{x}(t_n)) = \mathcal{N}(\mathbf{y}_n|\mathbf{H}\mathbf{x}(t_n), \mathbf{R}),$$

- We consider a variational approximation of the posterior using a time-varying linear SDE:

$$d\mathbf{x} = \mathbf{f}_L(\mathbf{x}, t)dt + \sqrt{\mathbf{\Sigma}}\, d\mathbf{W},$$

where

$$\mathbf{f}_L(\mathbf{x}, t) = -\mathbf{A}(t)\mathbf{x} + \mathbf{b}(t).$$

Girsanov change of measure

- Measure for the drift $\mathbf{f}$ denoted by P and the one for drift $\mathbf{f}_L$ by Q.
- The KL divergence in the infinite dimensional setting is given by Radon-Nikodym derivative of Q with respect to P:

$$\mathrm{KL}[Q\|P] = \int dQ \ln \frac{dQ}{dP} = E_Q \ln \frac{dQ}{dP},$$

which can be computed as

$$\frac{dQ}{dP} = \exp\left\{ -\int_{t_0}^{t_f} (\mathbf{f} - \mathbf{f}_L)^\top \boldsymbol{\Sigma}^{-1/2} \, d\widehat{W}_t + \frac{1}{2} \int_{t_0}^{t_f} (\mathbf{f} - \mathbf{f}_L)^\top \boldsymbol{\Sigma}^{-1} (\mathbf{f} - \mathbf{f}_L) \, dt \right\},$$

where $\widehat{W}$ is a Wiener process with respect to Q.

Girsanov change of measure

- Measure for the drift $\mathbf{f}$ denoted by P and the one for drift $\mathbf{f}_L$ by Q.
- The KL divergence in the infinite dimensional setting is given by Radon-Nikodym derivative of Q with respect to P:

$$\mathrm{KL}[Q\|P] \quad = \int dQ \ln \frac{dQ}{dP} = E_Q \ln \frac{dQ}{dP} \ ,$$

which can be computed as

$$\frac{dQ}{dP} \quad = \exp\left\{ - \int_{t_0}^{t_f} (\mathbf{f} - \mathbf{f}_L)^\top \boldsymbol{\Sigma}^{-1/2} \, d\widehat{W}_t + \frac{1}{2} \int_{t_0}^{t_f} (\mathbf{f} - \mathbf{f}_L)^\top \boldsymbol{\Sigma}^{-1} (\mathbf{f} - \mathbf{f}_L) \, dt \right\},$$

where $\widehat{W}$ is a Wiener process with respect to Q.

KL divergence

- Hence, KL divergence is

$$\mathrm{KL}[Q\|P] = \frac{1}{2}\int_{t_0}^{t_f}\left\langle (\mathbf{f}(\mathbf{x}(t),t)-\mathbf{f}_L(\mathbf{x}(t),t))^\top \boldsymbol{\Sigma}^{-1}(\mathbf{f}(\mathbf{x}(t),t)-\mathbf{f}_L(\mathbf{x}(t),t))\right\rangle_{q_t}$$

where $\langle\cdot\rangle_{q_t}$ denotes the expectation with respect to the marginal density at time t of the measure Q.

Variational approximation

- As approximating SDE is linear, marginal distribution q_t is Gaussian

$$q_t(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\mathbf{m}(t), \mathbf{S}(t)).$$

- with the mean $\mathbf{m}(t)$ and covariance $\mathbf{S}(t)$ described by ordinary differential equations (ODEs):

$$\frac{d\mathbf{m}}{dt} = -\mathbf{A}\mathbf{m} + \mathbf{b},$$
$$\frac{d\mathbf{S}}{dt} = -\mathbf{A}\mathbf{S} - \mathbf{S}\mathbf{A}^{\mathrm{T}} + \mathbf{\Sigma}.$$

Variational approximation

- As approximating SDE is linear, marginal distribution q_t is Gaussian

$$q_t(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\mathbf{m}(t), \mathbf{S}(t)).$$

- with the mean $\mathbf{m}(t)$ and covariance $\mathbf{S}(t)$ described by ordinary differential equations (ODEs):

$$\begin{aligned} \frac{d\mathbf{m}}{dt} &= -\mathbf{A}\mathbf{m} + \mathbf{b}, \\ \frac{d\mathbf{S}}{dt} &= -\mathbf{A}\mathbf{S} - \mathbf{S}\mathbf{A}^{\mathrm{T}} + \mathbf{\Sigma}. \end{aligned}$$

Algorithmics

- Using Lagrangian methods can derive algorithm that finds the variational approximation by minimising the KL divergence between posterior and approximating distribution.
- But KL also appears in the PAC-Bayes bound – is it possible to define appropriate loss over paths ω that captures the properties of interest?

Algorithmics

- Using Lagrangian methods can derive algorithm that finds the variational approximation by minimising the KL divergence between posterior and approximating distribution.
- But KL also appears in the PAC-Bayes bound – is it possible to define appropriate loss over paths ω that captures the properties of interest?

Error estimation

- For $\omega : [0, T] \longrightarrow \mathbb{R}^D$ defining a trajectory $\omega(t) \in \mathbb{R}^D$, we define the classifier h_ω by

$$h_\omega(\mathbf{y}, t) = \begin{cases} 1; & \text{if } \|\mathbf{y} - \mathbf{H}\omega(t)\| \leq \epsilon; \\ 0; & \text{otherwise.} \end{cases}$$

- where the actual observations are linear functions of the state variable given by the operator $\mathbf{H}$.
- Prior and posterior distribution over functions are inherited from distributions P and Q over paths ω.
- Hence, $P = p_{\text{sde}}$ and $Q = q$ defined by linear approximating sde.

Error estimation

- For $\omega : [0, T] \longrightarrow \mathbb{R}^D$ defining a trajectory $\omega(t) \in \mathbb{R}^D$, we define the classifier h_ω by

$$h_\omega(\mathbf{y}, t) = \begin{cases} 1; & \text{if } \|\mathbf{y} - \mathbf{H}\omega(t)\| \leq \epsilon; \\ 0; & \text{otherwise.} \end{cases}$$

- where the actual observations are linear functions of the state variable given by the operator $\mathbf{H}$.
- Prior and posterior distribution over functions are inherited from distributions P and Q over paths ω.
- Hence, $P = p_{\text{sde}}$ and $Q = q$ defined by linear approximating sde.

Error estimation

- For $\omega : [0, T] \longrightarrow \mathbb{R}^D$ defining a trajectory $\omega(t) \in \mathbb{R}^D$, we define the classifier h_ω by

$$h_\omega(\mathbf{y}, t) = \begin{cases} 1; & \text{if } \|\mathbf{y} - \mathbf{H}\omega(t)\| \leq \epsilon; \\ 0; & \text{otherwise.} \end{cases}$$

- where the actual observations are linear functions of the state variable given by the operator $\mathbf{H}$.
- Prior and posterior distribution over functions are inherited from distributions P and Q over paths ω.
- Hence, $P = p_{\text{sde}}$ and $Q = q$ defined by linear approximating sde.

Error estimation

- For $\omega : [0, T] \longrightarrow \mathbb{R}^D$ defining a trajectory $\omega(t) \in \mathbb{R}^D$, we define the classifier h_ω by

$$h_\omega(\mathbf{y}, t) = \begin{cases} 1; & \text{if } \|\mathbf{y} - \mathbf{H}\omega(t)\| \leq \epsilon; \\ 0; & \text{otherwise.} \end{cases}$$

- where the actual observations are linear functions of the state variable given by the operator $\mathbf{H}$.
- Prior and posterior distribution over functions are inherited from distributions P and Q over paths ω.
- Hence, $P = p_{\text{sde}}$ and $Q = q$ defined by linear approximating sde.

Generalisation analysis

- For the PAC-Bayes analysis we must compute: $\mathrm{KL}(Q\|P)$, e_Q, $\hat{e}_Q$. We have as above

$$\mathrm{KL}(Q\|P) = \int dq \ln \frac{dq}{dp_{\mathrm{sde}}}.$$

- If we now consider a fixed sample $(\mathbf{y}, t)$ we can estimate

$$\mathbb{E}_{\omega\sim Q}\left[h_\omega(\mathbf{y}, t)\right] = \int I\left[\|\mathbf{Hx} - \mathbf{y}\| \leq \epsilon\right] dq_t(\mathbf{x}),$$

- For sufficiently small values of ϵ we can approximate by

$$\approx \frac{V_d \epsilon^d}{(2\pi|\mathbf{HS}(t)\mathbf{H}^T|)^{d/2}} \exp\left(-(\mathbf{y} - \mathbf{Hm}(t))^T(\mathbf{HS}(t)\mathbf{H}^T)^{-1}(\mathbf{y} - \mathbf{Hm}(t))\right),$$

where V_d is the volume of a unit ball in $\mathbb{R}^d$.

Generalisation analysis

- For the PAC-Bayes analysis we must compute: $\mathrm{KL}(Q\|P)$, e_Q, $\hat{e}_Q$. We have as above

$$\mathrm{KL}(Q\|P) = \int dq \ln \frac{dq}{dp_{\mathrm{sde}}}.$$

- If we now consider a fixed sample $(\mathbf{y}, t)$ we can estimate

$$\mathbb{E}_{\omega\sim Q}\left[h_\omega(\mathbf{y}, t)\right] = \int I\left[\|\mathbf{Hx} - \mathbf{y}\| \leq \epsilon\right] dq_t(\mathbf{x}),$$

- For sufficiently small values of ϵ we can approximate by

$$\approx \frac{V_d \epsilon^d}{(2\pi|\mathbf{HS}(t)\mathbf{H}^T|)^{d/2}} \exp\left(-(\mathbf{y} - \mathbf{Hm}(t))^T(\mathbf{HS}(t)\mathbf{H}^T)^{-1}(\mathbf{y} - \mathbf{Hm}(t))\right),$$

where V_d is the volume of a unit ball in $\mathbb{R}^d$.

Generalisation analysis

- For the PAC-Bayes analysis we must compute: $\mathrm{KL}(Q\|P)$, e_Q, $\hat{e}_Q$. We have as above

$$\mathrm{KL}(Q\|P) = \int dq \ln \frac{dq}{dp_{\mathrm{sde}}}.$$

- If we now consider a fixed sample $(\mathbf{y}, t)$ we can estimate

$$\mathbb{E}_{\omega\sim Q}\left[h_\omega(\mathbf{y}, t)\right] = \int I\left[\|\mathbf{Hx} - \mathbf{y}\| \leq \epsilon\right] dq_t(\mathbf{x}),$$

- For sufficiently small values of ϵ we can approximate by

$$\approx \frac{V_d \epsilon^d}{(2\pi|\mathbf{HS}(t)\mathbf{H}^T|)^{d/2}} \exp\left(-(\mathbf{y} - \mathbf{Hm}(t))^T(\mathbf{HS}(t)\mathbf{H}^T)^{-1}(\mathbf{y} - \mathbf{Hm}(t))\right),$$

where V_d is the volume of a unit ball in $\mathbb{R}^d$.

Error estimates

- Note that e_Q is simply

$$e_Q = \mathbb{E}_{(\mathbf{y},t)\sim\mu}\mathbb{E}_{\omega\sim Q}\left[h_\omega(\mathbf{y},t)\right] \propto \int \mathcal{N}(\mathbf{y}|\mathbf{Hm}(t), \mathbf{HS}(t)\mathbf{H}^T) d\mu(\mathbf{y},t),$$

  while $\hat{e}_Q$ is the empirical average of this quantity.
- A tension arises in setting ϵ – if large approximation inaccurate.
- If e_Q and $\hat{e}_Q$ both small, the bound implied by the $\mathrm{KL}(e_Q||\hat{e}_Q) \leq C$ becomes weak.

Error estimates

- Note that e_Q is simply

$$e_Q = \mathbb{E}_{(\mathbf{y},t)\sim\mu}\mathbb{E}_{\omega\sim Q}\left[h_\omega(\mathbf{y},t)\right] \propto \int \mathcal{N}(\mathbf{y}|\mathbf{Hm}(t), \mathbf{HS}(t)\mathbf{H}^T)d\mu(\mathbf{y},t),$$

  while $\hat{e}_Q$ is the empirical average of this quantity.
- A tension arises in setting ϵ – if large approximation inaccurate.
- If e_Q and $\hat{e}_Q$ both small, the bound implied by the $\mathrm{KL}(e_Q||\hat{e}_Q) \leq C$ becomes weak.

Error estimates

- Note that e_Q is simply

$$e_Q = \mathbb{E}_{(\mathbf{y},t)\sim\mu}\mathbb{E}_{\omega\sim Q}\left[h_\omega(\mathbf{y},t)\right] \propto \int \mathcal{N}(\mathbf{y}|\mathbf{Hm}(t), \mathbf{HS}(t)\mathbf{H}^T) d\mu(\mathbf{y},t),$$

  while $\hat{e}_Q$ is the empirical average of this quantity.
- A tension arises in setting ϵ – if large approximation inaccurate.
- If e_Q and $\hat{e}_Q$ both small, the bound implied by the $\mathrm{KL}(e_Q\|\hat{e}_Q) \leq C$ becomes weak.

Refining the distributions

- Overcome this weakness by taking K-fold product distributions and defining $h_{(\omega_1,\ldots,\omega_K)}$ as

$$h_{(\omega_1,\ldots,\omega_K)}(\mathbf{y},t) = \begin{cases} 1; & \text{if there exists } 1 \leq i \leq K \text{ such that} \|\mathbf{y} - \mathbf{H}\omega_i(t)\| \leq \epsilon; \\ 0; & \text{otherwise.} \end{cases}$$

- We now have

$$\mathbb{E}_{(\omega_1,\ldots,\omega_K)\sim Q^K}\left[h_{(\omega_1,\ldots,\omega_K)}(\mathbf{y},t)\right] \approx 1 - \left(1 - \int I\left[\|\mathbf{H}\mathbf{x} - \mathbf{y}\| \leq \epsilon\right] dq_t(\mathbf{x})\right)^K$$
$$\approx K V_d \epsilon^d \mathcal{N}(\mathbf{y}|\mathbf{H}\mathbf{m}(t), \mathbf{H}\mathbf{S}(t)\mathbf{H}^T),$$

Refining the distributions

- Overcome this weakness by taking K-fold product distributions and defining $h_{(\omega_1,\ldots,\omega_K)}$ as

$$h_{(\omega_1,\ldots,\omega_K)}(\mathbf{y},t) = \begin{cases} 1; & \text{if there exists } 1 \leq i \leq K \text{ such that} \|\mathbf{y} - \mathbf{H}\omega_i(t)\| \leq \epsilon; \\ 0; & \text{otherwise.} \end{cases}$$

- We now have

$$\begin{aligned} \mathbb{E}_{(\omega_1,\ldots,\omega_K)\sim Q^K}\left[h_{(\omega_1,\ldots,\omega_K)}(\mathbf{y},t)\right] &\approx 1 - \left(1 - \int I\left[\|\mathbf{H}\mathbf{x} - \mathbf{y}\| \leq \epsilon\right] dq_t(\mathbf{x})\right)^K \\ &\approx K V_d \epsilon^d \mathcal{N}(\mathbf{y}|\mathbf{H}\mathbf{m}(t), \mathbf{H}\mathbf{S}(t)\mathbf{H}^T), \end{aligned}$$

Final result

- Putting all together gives final bound:

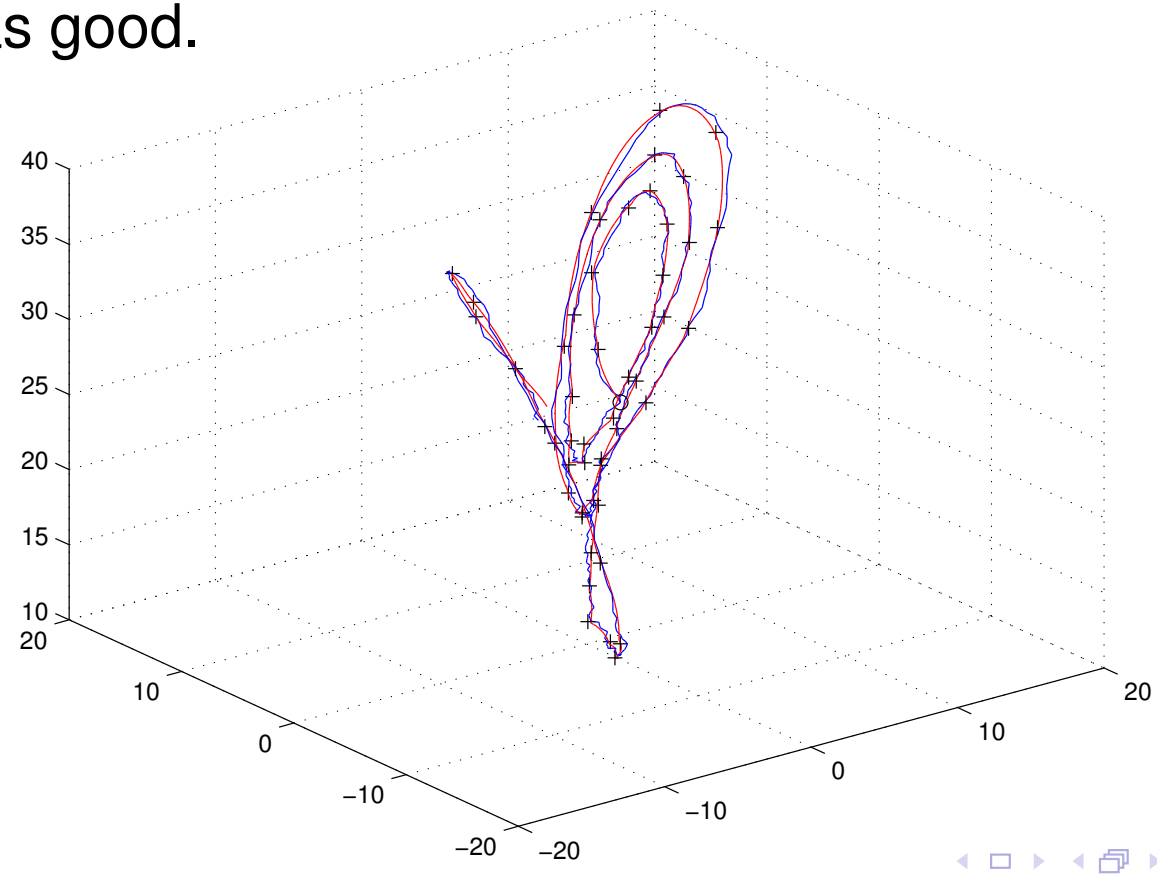
$$\mathbb{E}_{(\mathbf{y},t)\sim\mu}\left[\mathcal{N}(\mathbf{y}|\mathbf{H}\mathbf{m}(t),\mathbf{H}\mathbf{S}(t)\mathbf{H}^T)\right] \geq \frac{1}{V_d\epsilon^d K}\mathrm{KL}^{-1}\left(KV_d\epsilon^d\hat{\mathbb{E}}\left[\mathcal{N}(\mathbf{y}|\mathbf{H}\mathbf{m}(t),\mathbf{H}\mathbf{S}(t)\mathbf{H}^T)\right], \frac{K\int_0^T E_{\mathrm{sde}}(t)dt + \ln((m+1)/\delta)}{m}\right).$$

where

$$E_{sde}(t) = \frac{1}{2}\left\langle(\mathbf{f}(\mathbf{x})-\mathbf{f}_L(\mathbf{x},t))^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{f}(\mathbf{x})-\mathbf{f}_L(\mathbf{x},t))\right\rangle_{q_t},$$

Small scale experiment

- We applied the analysis to the results of performing a variational Bayesian approximation to the Lorentz attractor in three dimension. The quality of the fit with 49 examples was good.



Small scale experiment

- chose $V_d \epsilon^d$ to optimise the bound – fairly small ball implying that our approximation should be reasonable.
- compared the bound with the left hand side estimated on a random draw of 99 test points. The corresponding values are

m	dt	$\hat{e}_Q$	A	e_Q	$\mathrm{KL}^{-1}(\cdot,\cdot)/V$
49	0.005	0.137	3.536	0.128	0.004

Small scale experiment

- chose $V_d \epsilon^d$ to optimise the bound – fairly small ball implying that our approximation should be reasonable.
- compared the bound with the left hand side estimated on a random draw of 99 test points. The corresponding values are

m	dt	$\hat{e}_Q$	A	e_Q	$\mathrm{KL}^{-1}(\cdot,\cdot)/V$
49	0.005	0.137	3.536	0.128	0.004

Conclusions

- Links between luckiness and choosing the prior based on the data distribution
- Applications to maximum entropy classification
- Also consider lower bounding the accuracy of a posterior distribution for Gaussian processes (GP)
- Applied the theory to bound the performance of estimations made using approximate Bayesian inference for dynamical systems:
  - Prior determined by a non-linear stochastic differential equation (SDE)
  - Variational approximation results in posterior given by an approximating linear SDE – hence Gaussian process posterior.

Conclusions

- Links between luckiness and choosing the prior based on the data distribution
- Applications to maximum entropy classification
- Also consider lower bounding the accuracy of a posterior distribution for Gaussian processes (GP)
- Applied the theory to bound the performance of estimations made using approximate Bayesian inference for dynamical systems:
  - Prior determined by a non-linear stochastic differential equation (SDE)
  - Variational approximation results in posterior given by an approximating linear SDE – hence Gaussian process posterior.

Conclusions

- Links between luckiness and choosing the prior based on the data distribution
- Applications to maximum entropy classification
- Also consider lower bounding the accuracy of a posterior distribution for Gaussian processes (GP)
- Applied the theory to bound the performance of estimations made using approximate Bayesian inference for dynamical systems:
  - Prior determined by a non-linear stochastic differential equation (SDE)
  - Variational approximation results in posterior given by an approximating linear SDE – hence Gaussian process posterior.

Conclusions

- Links between luckiness and choosing the prior based on the data distribution
- Applications to maximum entropy classification
- Also consider lower bounding the accuracy of a posterior distribution for Gaussian processes (GP)
- Applied the theory to bound the performance of estimations made using approximate Bayesian inference for dynamical systems:
  - Prior determined by a non-linear stochastic differential equation (SDE)
  - Variational approximation results in posterior given by an approximating linear SDE – hence Gaussian process posterior.

Conclusions

- Links between luckiness and choosing the prior based on the data distribution
- Applications to maximum entropy classification
- Also consider lower bounding the accuracy of a posterior distribution for Gaussian processes (GP)
- Applied the theory to bound the performance of estimations made using approximate Bayesian inference for dynamical systems:
  - Prior determined by a non-linear stochastic differential equation (SDE)
  - Variational approximation results in posterior given by an approximating linear SDE – hence Gaussian process posterior.

Conclusions

- Links between luckiness and choosing the prior based on the data distribution
- Applications to maximum entropy classification
- Also consider lower bounding the accuracy of a posterior distribution for Gaussian processes (GP)
- Applied the theory to bound the performance of estimations made using approximate Bayesian inference for dynamical systems:
  - Prior determined by a non-linear stochastic differential equation (SDE)
  - Variational approximation results in posterior given by an approximating linear SDE – hence Gaussian process posterior.