Case Study: Bayesian Linear Regression and Sparse Bayesian Models

Piyush Rai

Dept. of CSE, IIT Kanpur

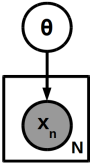
(Mini-course: lecture 2)

Nov 05, 2015

Recap

Maximum Likelihood Estimation (MLE)

- We wish to estimate parameters θ from observed data $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$



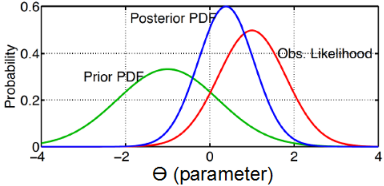
- MLE does this by finding θ that maximizes the (log)likelihood $p(\mathbf{X}|\theta)$

$$\hat{\theta} = \arg\max_{\theta} \log p(\mathbf{X}|\theta) = \arg\max_{\theta} \log \prod_{n=1}^{N} p(\mathbf{x}_n|\theta) = \arg\max_{\theta} \sum_{n=1}^{N} \log p(\mathbf{x}_n|\theta)$$

- MLE now reduces to solving an optimization problem w.r.t. θ

Maximum-a-Posteriori (MAP) Estimation

Incorporating **prior knowledge** $p(\theta)$ about the parameters



- MAP estimation finds θ that maximizes the posterior $p(\theta|\mathbf{X}) \propto p(\mathbf{X}|\theta)p(\theta)$

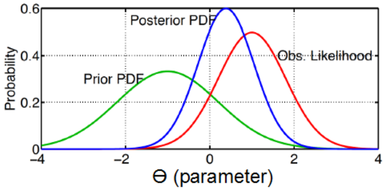
$$\hat{\theta} = \arg\max_{\theta} \log \prod_{n=1}^{N} p(\mathbf{x}_n|\theta)p(\theta) = \arg\max_{\theta} \sum_{n=1}^{N} \log p(\mathbf{x}_n|\theta) + \log p(\theta)$$

- MAP now reduces to solving an optimization problem w.r.t. θ
- Objective function very similar to MLE, except for the $\log p(\theta)$ term
- In some sense, MAP is just a "regularized" MLE

Bayesian Learning

- Both MLE and MAP only give a point estimate (single best answer) of θ
- How can we capture/quantify the uncertainty in θ?
- Need to infer the full posterior distribution

$$p(\theta|\mathbf{X}) = \frac{p(\mathbf{X}|\theta)p(\theta)}{p(\mathbf{X})} = \frac{p(\mathbf{X}|\theta)p(\theta)}{\int_\theta p(\mathbf{X}|\theta)p(\theta)d\theta} \propto \text{Likelihood} \times \text{Prior}$$



- Requires doing a "fully Bayesian" inference
- Inference sometimes a somewhat easy and sometimes a (very) hard problem
- **Conjugate priors** often make life easy when doing inference

Warm-up: Least Squares Regression

- Training data: $\{\mathbf{x}_n, y_n\}_{n=1}^N$. Response is a noisy function of the input

$$y_n = f(\mathbf{x}_n, \mathbf{w}) + \epsilon_n$$

Warm-up: Least Squares Regression

- Training data: $\{\mathbf{x}_n, y_n\}_{n=1}^N$. Response is a noisy function of the input

$$y_n = f(\mathbf{x}_n, \mathbf{w}) + \epsilon_n$$

- Assume a data representation $\phi(\mathbf{x}_n) = [\phi_1(\mathbf{x}_n), \ldots, \phi_M(\mathbf{x}_n)] \in \mathbb{R}^M$

Warm-up: Least Squares Regression

- Training data: $\{\mathbf{x}_n, y_n\}_{n=1}^N$. Response is a noisy function of the input

$$y_n = f(\mathbf{x}_n, \mathbf{w}) + \epsilon_n$$

- Assume a data representation $\phi(\mathbf{x}_n) = [\phi_1(\mathbf{x}_n), \ldots, \phi_M(\mathbf{x}_n)] \in \mathbb{R}^M$
- Denote $\mathbf{y} = [y_1, \ldots, y_N]^\top \in \mathbb{R}^N$, $\mathbf{\Phi} = [\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_N)]^\top \in \mathbb{R}^{N \times M}$

Warm-up: Least Squares Regression

- Training data: $\{\mathbf{x}_n, y_n\}_{n=1}^N$. Response is a noisy function of the input

$$y_n = f(\mathbf{x}_n, \mathbf{w}) + \epsilon_n$$

- Assume a data representation $\phi(\mathbf{x}_n) = [\phi_1(\mathbf{x}_n), \ldots, \phi_M(\mathbf{x}_n)] \in \mathbb{R}^M$
- Denote $\mathbf{y} = [y_1, \ldots, y_N]^\top \in \mathbb{R}^N$, $\mathbf{\Phi} = [\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_N)]^\top \in \mathbb{R}^{N \times M}$
- Assume linear (in the parameters) function: $f(\mathbf{x}_n, \mathbf{w}) = \mathbf{w}^\top \phi(\mathbf{x}_n)$

Warm-up: Least Squares Regression

- Training data: $\{\mathbf{x}_n, y_n\}_{n=1}^N$. Response is a noisy function of the input

$$y_n = f(\mathbf{x}_n, \mathbf{w}) + \epsilon_n$$

- Assume a data representation $\phi(\mathbf{x}_n) = [\phi_1(\mathbf{x}_n), \dots, \phi_M(\mathbf{x}_n)] \in \mathbb{R}^M$
- Denote $\mathbf{y} = [y_1, \dots, y_N]^\top \in \mathbb{R}^N$, $\mathbf{\Phi} = [\phi(\mathbf{x}_1), \dots, \phi(\mathbf{x}_N)]^\top \in \mathbb{R}^{N \times M}$
- Assume linear (in the parameters) function: $f(\mathbf{x}_n, \mathbf{w}) = \mathbf{w}^\top \phi(\mathbf{x}_n)$
- Sum of squared error function

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N |f(\mathbf{x}_n, \mathbf{w}) - y_n|^2$$

Warm-up: Least Squares Regression

- Training data: $\{\mathbf{x}_n, y_n\}_{n=1}^N$. Response is a noisy function of the input

$$y_n = f(\mathbf{x}_n, \mathbf{w}) + \epsilon_n$$

- Assume a data representation $\phi(\mathbf{x}_n) = [\phi_1(\mathbf{x}_n), \ldots, \phi_M(\mathbf{x}_n)] \in \mathbb{R}^M$
- Denote $\mathbf{y} = [y_1, \ldots, y_N]^\top \in \mathbb{R}^N$, $\mathbf{\Phi} = [\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_N)]^\top \in \mathbb{R}^{N \times M}$
- Assume linear (in the parameters) function: $f(\mathbf{x}_n, \mathbf{w}) = \mathbf{w}^\top \phi(\mathbf{x}_n)$
- Sum of squared error function

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} |f(\mathbf{x}_n, \mathbf{w}) - y_n|^2$$

- Classical solution: $\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} E(\mathbf{w}) = (\mathbf{\Phi}^\top \mathbf{\Phi})^{-1} \mathbf{\Phi}^\top \mathbf{y}$

Warm-up: Least Squares Regression

- Training data: $\{\mathbf{x}_n, y_n\}_{n=1}^N$. Response is a noisy function of the input

$$y_n = f(\mathbf{x}_n, \mathbf{w}) + \epsilon_n$$

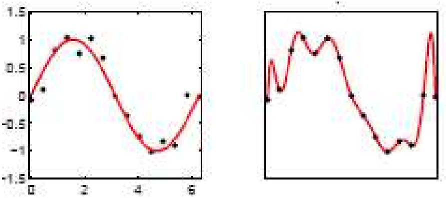
- Assume a data representation $\phi(\mathbf{x}_n) = [\phi_1(\mathbf{x}_n), \ldots, \phi_M(\mathbf{x}_n)] \in \mathbb{R}^M$
- Denote $\mathbf{y} = [y_1, \ldots, y_N]^\top \in \mathbb{R}^N$, $\mathbf{\Phi} = [\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_N)]^\top \in \mathbb{R}^{N\times M}$
- Assume linear (in the parameters) function: $f(\mathbf{x}_n, \mathbf{w}) = \mathbf{w}^\top \phi(\mathbf{x}_n)$
- Sum of squared error function

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N} |f(\mathbf{x}_n, \mathbf{w}) - y_n|^2$$

- Classical solution: $\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} E(\mathbf{w}) = (\mathbf{\Phi}^\top\mathbf{\Phi})^{-1}\mathbf{\Phi}^\top\mathbf{y}$
- Classification: replace the least squares by some other loss (e.g., logistic)

Regularization

- Want functions that are "simple" (and hence "generalize" to future data)



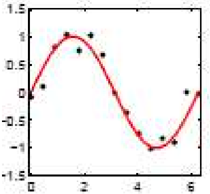
- How: penalize "complex" functions. Use a regularized loss function

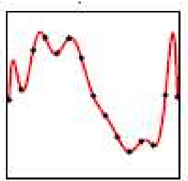
$$\tilde{E}(\mathbf{w}) = E(\mathbf{w}) + \lambda\Omega(\mathbf{w})$$

- $\Omega(\mathbf{w})$: a measure of how complex $\mathbf{w}$ is (want it small)

Regularization

- Want functions that are "simple" (and hence "generalize" to future data)





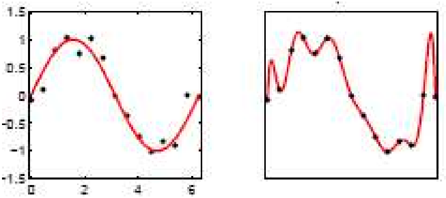
- How: penalize "complex" functions. Use a regularized loss function

$$\tilde{E}(\mathbf{w}) = E(\mathbf{w}) + \lambda\Omega(\mathbf{w})$$

- $\Omega(\mathbf{w})$: a measure of how complex $\mathbf{w}$ is (want it small)
- Regularization parameter λ trades off data fit vs model simplicity

Regularization

- Want functions that are "simple" (and hence "generalize" to future data)



- How: penalize "complex" functions. Use a regularized loss function

$$\tilde{E}(\mathbf{w}) = E(\mathbf{w}) + \lambda\Omega(\mathbf{w})$$

- $\Omega(\mathbf{w})$: a measure of how complex $\mathbf{w}$ is (want it small)
- Regularization parameter λ trades off data fit vs model simplicity
- For $\Omega(\mathbf{w}) = ||\mathbf{w}||^2$, the solution $\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \tilde{E}(\mathbf{w}) = (\mathbf{\Phi}^\top\mathbf{\Phi} + \lambda\mathbf{I})^{-1}\mathbf{\Phi}^\top\mathbf{y}$

A Probabilistic Framework for Regression

- Recall: $y_n = f(\mathbf{x}_n, \mathbf{w}) + \epsilon_n$
- Assume a zero-mean Gaussian error

$$p(\epsilon|\sigma^2) = \mathcal{N}(\epsilon|0, \sigma^2)$$

A Probabilistic Framework for Regression

- Recall: $y_n = f(\mathbf{x}_n, \mathbf{w}) + \epsilon_n$
- Assume a zero-mean Gaussian error

$$p(\epsilon|\sigma^2) = \mathcal{N}(\epsilon|0, \sigma^2)$$

- Leads to a Gaussian likelihood model $p(y_n|\mathbf{x}_n, \mathbf{w}) = \mathcal{N}(y_n|f(\mathbf{x}_n, \mathbf{w}), \sigma^2)$

$$p(y_n|\mathbf{x}_n, \mathbf{w}) = \left(\frac{1}{2\pi\sigma^2}\right)^{1/2} \exp\left\{-\frac{1}{2\sigma^2}(f(\mathbf{x}_n, \mathbf{w}) - y_n)^2\right\}$$

A Probabilistic Framework for Regression

- Recall: $y_n = f(\mathbf{x}_n, \mathbf{w}) + \epsilon_n$
- Assume a zero-mean Gaussian error

$$p(\epsilon|\sigma^2) = \mathcal{N}(\epsilon|0, \sigma^2)$$

- Leads to a Gaussian likelihood model $p(y_n|\mathbf{x}_n, \mathbf{w}) = \mathcal{N}(y_n|f(\mathbf{x}_n, \mathbf{w}), \sigma^2)$

$$p(y_n|\mathbf{x}_n, \mathbf{w}) = \left(\frac{1}{2\pi\sigma^2}\right)^{1/2} \exp\left\{-\frac{1}{2\sigma^2}(f(\mathbf{x}_n, \mathbf{w}) - y_n)^2\right\}$$

- Joint probability of the data (likelihood)

$$L(\mathbf{w}) = \prod_{n=1}^{N} p(y_n|\mathbf{x}_n, \mathbf{w}) = \left(\frac{1}{2\pi\sigma^2}\right)^{N/2} \exp\left\{-\frac{1}{2\sigma^2}\sum_{n=1}^{N}(f(\mathbf{x}_n, \mathbf{w}) - y_n)^2\right\}$$

A Probabilistic Framework for Regression

- Let's look at the negative log-likelihood

$$-\log L(\mathbf{w}) = \frac{N}{2}\log\sigma^2 + \frac{N}{2}\log 2\pi + \frac{1}{2\sigma^2}\sum_{n=1}^{N}(f(\mathbf{x}_n,\mathbf{w}) - y_n)^2$$

A Probabilistic Framework for Regression

- Let's look at the negative log-likelihood

$$-\log L(\mathbf{w}) = \frac{N}{2}\log\sigma^2 + \frac{N}{2}\log 2\pi + \frac{1}{2\sigma^2}\sum_{n=1}^{N}(f(\mathbf{x}_n, \mathbf{w}) - y_n)^2$$

- Minimizing w.r.t. **w** leads to the same answer as the unregularized case

$$\hat{\mathbf{w}} = (\mathbf{\Phi}^\top\mathbf{\Phi})^{-1}\mathbf{\Phi}^\top\mathbf{y}$$

A Probabilistic Framework for Regression

- Let's look at the negative log-likelihood

$$-\log L(\mathbf{w}) = \frac{N}{2}\log\sigma^2 + \frac{N}{2}\log 2\pi + \frac{1}{2\sigma^2}\sum_{n=1}^{N}(f(\mathbf{x}_n,\mathbf{w}) - y_n)^2$$

- Minimizing w.r.t. **w** leads to the same answer as the unregularized case

$$\hat{\mathbf{w}} = (\mathbf{\Phi}^\top\mathbf{\Phi})^{-1}\mathbf{\Phi}^\top\mathbf{y}$$

- Also get an estimate of error variance

$$\frac{1}{\hat{\sigma}^2} = \frac{1}{N}\sum_{n=1}^{N}(f(\mathbf{x}_n,\hat{\mathbf{w}}) - y_n)^2$$

Specifying a Prior and Computing the Posterior

- Let's assume a Gaussian prior on the weight vector $\mathbf{w} = [w_1, \ldots, w_M]$

$$p(\mathbf{w}|\alpha) = \prod_{m=1}^{M} p(w_m|\alpha) = \prod_{m=1}^{M} \left(\frac{\alpha}{2\pi}\right)^{1/2} \exp\left(-\frac{\alpha}{2} w_m^2\right)$$

Specifying a Prior and Computing the Posterior

- Let's assume a Gaussian prior on the weight vector $\mathbf{w} = [w_1, \ldots, w_M]$

$$p(\mathbf{w}|\alpha) = \prod_{m=1}^{M} p(w_m|\alpha) = \prod_{m=1}^{M} \left(\frac{\alpha}{2\pi}\right)^{1/2} \exp\left(-\frac{\alpha}{2} w_m^2\right)$$

- The posterior

$$p(\mathbf{w}|\mathbf{y}, \alpha, \sigma^2) = \frac{\text{likelihood} \times \text{prior}}{\text{normalizing factor}} = \frac{p(\mathbf{y}|\mathbf{w}, \sigma^2) \times p(\mathbf{w}|\alpha)}{p(\mathbf{y}|\alpha, \sigma^2)}$$

Specifying a Prior and Computing the Posterior

- Let's assume a Gaussian prior on the weight vector $\mathbf{w} = [w_1, \ldots, w_M]$

$$p(\mathbf{w}|\alpha) = \prod_{m=1}^{M} p(w_m|\alpha) = \prod_{m=1}^{M} \left(\frac{\alpha}{2\pi}\right)^{1/2} \exp\left(-\frac{\alpha}{2} w_m^2\right)$$

- The posterior

$$p(\mathbf{w}|\mathbf{y}, \alpha, \sigma^2) = \frac{\text{likelihood} \times \text{prior}}{\text{normalizing factor}} = \frac{p(\mathbf{y}|\mathbf{w}, \sigma^2) \times p(\mathbf{w}|\alpha)}{p(\mathbf{y}|\alpha, \sigma^2)}$$

- The posterior $p(\mathbf{w}|\mathbf{y}, \alpha, \sigma^2)$ will be Gaussian $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

$$\begin{aligned} \boldsymbol{\mu} &= (\boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \sigma^2 \alpha \mathbf{I})^{-1} \boldsymbol{\Phi}^\top \mathbf{y} \\ \boldsymbol{\Sigma} &= \sigma^2 (\boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \sigma^2 \alpha \mathbf{I})^{-1} \end{aligned}$$

- Instead of a single estimate, we now have a distribution over $\mathbf{w}$

Maximizing the Posterior

- Recall: Gaussian prior on the weight vector $\mathbf{w} = [w_1, \ldots, w_M]$

$$p(\mathbf{w}|\alpha) = \prod_{m=1}^{M} p(w_m|\alpha) = \prod_{m=1}^{M} \left(\frac{\alpha}{2\pi}\right)^{1/2} \exp\left(-\frac{\alpha}{2} w_m^2\right)$$

- The likelihood $p(\mathbf{y}_n|\mathbf{w}, \mathbf{x}_n, \sigma^2) \propto \exp\left\{-\frac{1}{2\sigma^2}(f(\mathbf{x}_n, \mathbf{w}) - y_n)^2\right\}$

Maximizing the Posterior

- Recall: Gaussian prior on the weight vector $\mathbf{w} = [w_1, \ldots, w_M]$

$$p(\mathbf{w}|\alpha) = \prod_{m=1}^{M} p(w_m|\alpha) = \prod_{m=1}^{M} \left(\frac{\alpha}{2\pi}\right)^{1/2} \exp\left(-\frac{\alpha}{2} w_m^2\right)$$

- The likelihood $p(\mathbf{y}_n|\mathbf{w}, \mathbf{x}_n, \sigma^2) \propto \exp\left\{-\frac{1}{2\sigma^2}(f(\mathbf{x}_n, \mathbf{w}) - y_n)^2\right\}$
- Maximizing the posterior $p(\mathbf{w}|\mathbf{y}, \alpha, \sigma^2) \propto p(\mathbf{y}|\mathbf{w}, \sigma^2) \times p(\mathbf{w}|\alpha)$ w.r.t $\mathbf{w}$ is equivalent to minimizing

$$E_{MAP}(\mathbf{w}) = \frac{1}{2\sigma^2} \sum_{n=1}^{N} \{f(\mathbf{x}_n, \mathbf{w}) - y_n\}^2 + \frac{\alpha}{2} \sum_{m=1}^{M} w_m^2$$

Maximizing the Posterior

- Recall: Gaussian prior on the weight vector $\mathbf{w} = [w_1, \ldots, w_M]$

$$p(\mathbf{w}|\alpha) = \prod_{m=1}^{M} p(w_m|\alpha) = \prod_{m=1}^{M} \left(\frac{\alpha}{2\pi}\right)^{1/2} \exp\left(-\frac{\alpha}{2} w_m^2\right)$$

- The likelihood $p(\mathbf{y}_n|\mathbf{w}, \mathbf{x}_n, \sigma^2) \propto \exp\left\{-\frac{1}{2\sigma^2}(f(\mathbf{x}_n, \mathbf{w}) - y_n)^2\right\}$
- Maximizing the posterior $p(\mathbf{w}|\mathbf{y}, \alpha, \sigma^2) \propto p(\mathbf{y}|\mathbf{w}, \sigma^2) \times p(\mathbf{w}|\alpha)$ w.r.t $\mathbf{w}$ is equivalent to minimizing

$$E_{MAP}(\mathbf{w}) = \frac{1}{2\sigma^2} \sum_{n=1}^{N} \{f(\mathbf{x}_n, \mathbf{w}) - y_n\}^2 + \frac{\alpha}{2} \sum_{m=1}^{M} w_m^2$$

- Will lead to an identical solution as ridge-regression with $\lambda = \sigma^2\alpha$

Evolution of the Posterior

- Posterior updates have a naturally online flavor..

$$p(\mathbf{w}|y_1, y_2, y_3) \quad \propto \quad p(y_1, y_2, y_3|\mathbf{w})p(\mathbf{w})$$

Evolution of the Posterior

- Posterior updates have a naturally online flavor..

$$
\begin{aligned}
p(\mathbf{w}|y_1, y_2, y_3) &\propto p(y_1, y_2, y_3|\mathbf{w})p(\mathbf{w}) \\
&= p(y_2, y_3|\mathbf{w})p(y_1|\mathbf{w})p(\mathbf{w})
\end{aligned}
$$

Evolution of the Posterior

- Posterior updates have a naturally online flavor..

$$\begin{aligned} p(\mathbf{w}|y_1, y_2, y_3) &\propto p(y_1, y_2, y_3|\mathbf{w})p(\mathbf{w}) \\ &= p(y_2, y_3|\mathbf{w})p(y_1|\mathbf{w})p(\mathbf{w}) \\ &= p(y_2, y_3|\mathbf{w})p(\mathbf{w}|y_1) \end{aligned}$$

Evolution of the Posterior

- Posterior updates have a naturally online flavor..

$$
\begin{aligned}
p(\mathbf{w}|y_1, y_2, y_3) &\propto p(y_1, y_2, y_3|\mathbf{w})p(\mathbf{w}) \\
&= p(y_2, y_3|\mathbf{w})p(y_1|\mathbf{w})p(\mathbf{w}) \\
&= p(y_2, y_3|\mathbf{w})p(\mathbf{w}|y_1) \\
&= \text{likelihood w.r.t. } y_2 \,\&\, y_3 \times \text{posterior after seeing } y_1
\end{aligned}
$$

Let's Compare Predictions

- Ridge regression

$$\text{prediction} = f(\hat{\mathbf{w}}, \mathbf{x}_*)$$

Let's Compare Predictions

- Ridge regression

$$\text{prediction} = f(\hat{\mathbf{w}}, \mathbf{x}_*)$$

- MAP estimation (or "Pseudo" Bayesian)

$$\text{prediction} = p(y_* | \mathbf{w}_{MAP}, \mathbf{x}_*, \sigma^2)$$

Let's Compare Predictions

- Ridge regression

$$\text{prediction} = f(\hat{\mathbf{w}}, \mathbf{x}_*)$$

- MAP estimation (or "Pseudo" Bayesian)

$$\text{prediction} = p(y_*|\mathbf{w}_{MAP}, \mathbf{x}_*, \sigma^2)$$

- True Bayesian

$$\text{prediction} = p(y_*|\mathbf{x}_*, \mathbf{y}, \mathbf{X}, \sigma^2, \alpha) = \int p(y_*|\mathbf{w}, \mathbf{x}_*, \sigma^2) p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \alpha, \sigma^2) d\mathbf{w}$$

- The true Bayesian way integrates out or marginalizes/averages over the uncertain variables ($\mathbf{w}$ in this case) to get a predictive distribution

Not Quite Done Yet..

- We haven't really averaged over all unknowns (which also include α, σ^2)
- Ideally, would like to get the posterior over all the unknowns

$$p(\mathbf{w}, \alpha, \sigma^2|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{w}, \sigma^2)p(\mathbf{w}|\alpha)p(\alpha)p(\sigma^2)}{p(\mathbf{y})}$$

where $p(\mathbf{y}) = \int p(\mathbf{y}|\mathbf{w}, \sigma^2)p(\mathbf{w}|\alpha)p(\alpha)p(\sigma^2)\ d\mathbf{w}\ d\alpha\ d\sigma^2$ (hard to compute)

Not Quite Done Yet..

- We haven't really averaged over all unknowns (which also include α, σ^2)
- Ideally, would like to get the posterior over all the unknowns

$$p(\mathbf{w}, \alpha, \sigma^2|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{w}, \sigma^2)p(\mathbf{w}|\alpha)p(\alpha)p(\sigma^2)}{p(\mathbf{y})}$$

where $p(\mathbf{y}) = \int p(\mathbf{y}|\mathbf{w}, \sigma^2)p(\mathbf{w}|\alpha)p(\alpha)p(\sigma^2)\ d\mathbf{w}\ d\alpha\ d\sigma^2$ (hard to compute)

- Making prediction for new data points. The predictive distribution:

$$p(y_*|\mathbf{y}) = \int p(y_*|\mathbf{w}, \sigma^2)p(\mathbf{w}, \alpha, \sigma^2|\mathbf{y})\ d\mathbf{w}\ d\alpha\ d\sigma^2$$

.. again, hard to compute

Not Quite Done Yet..

- We haven't really averaged over all unknowns (which also include α, σ^2)
- Ideally, would like to get the posterior over all the unknowns

$$p(\mathbf{w}, \alpha, \sigma^2|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{w}, \sigma^2)p(\mathbf{w}|\alpha)p(\alpha)p(\sigma^2)}{p(\mathbf{y})}$$

where $p(\mathbf{y}) = \int p(\mathbf{y}|\mathbf{w}, \sigma^2)p(\mathbf{w}|\alpha)p(\alpha)p(\sigma^2)\ d\mathbf{w}\ d\alpha\ d\sigma^2$ (hard to compute)

- Making prediction for new data points. The predictive distribution:

$$p(y_*|\mathbf{y}) = \int p(y_*|\mathbf{w}, \sigma^2)p(\mathbf{w}, \alpha, \sigma^2|\mathbf{y})\ d\mathbf{w}\ d\alpha\ d\sigma^2$$

.. again, hard to compute

- Approx. Bayesian inference (Type-II maximum likelihood, Laplace approximation, MCMC, variational Bayes, etc.) saves the day..

Approximating the Predictive Distribution

- Making prediction for new data points

$$p(y_*|\mathbf{y}) = \int p(y_*|\mathbf{w}, \sigma^2) p(\mathbf{w}, \alpha, \sigma^2|\mathbf{y}) \, d\mathbf{w} \, d\alpha \, d\sigma^2$$

Approximating the Predictive Distribution

- Making prediction for new data points

$$
\begin{aligned}
p(y_*|\mathbf{y}) &= \int p(y_*|\mathbf{w},\sigma^2)p(\mathbf{w},\alpha,\sigma^2|\mathbf{y})\, d\mathbf{w}\, d\alpha\, d\sigma^2 \\
&= \int p(y_*|\mathbf{w},\sigma^2)p(\mathbf{w}|\alpha,\sigma^2,\mathbf{y})p(\alpha,\sigma^2|\mathbf{y})\, d\mathbf{w}\, d\alpha\, d\sigma^2
\end{aligned}
$$

Approximating the Predictive Distribution

- Making prediction for new data points

$$
\begin{aligned}
p(y_*|\mathbf{y}) &= \int p(y_*|\mathbf{w},\sigma^2)p(\mathbf{w},\alpha,\sigma^2|\mathbf{y})\, d\mathbf{w}\, d\alpha\, d\sigma^2 \\
&= \int p(y_*|\mathbf{w},\sigma^2)p(\mathbf{w}|\alpha,\sigma^2,\mathbf{y})p(\alpha,\sigma^2|\mathbf{y})\, d\mathbf{w}\, d\alpha\, d\sigma^2 \\
&\approx \int p(y_*|\mathbf{w},\sigma^2)p(\mathbf{w}|\alpha,\sigma^2,\mathbf{y})\delta(\alpha_{MP},\sigma^2_{MP})\, d\mathbf{w}\, d\alpha\, d\sigma^2
\end{aligned}
$$

Approximating the Predictive Distribution

- Making prediction for new data points

$$
\begin{aligned}
p(y_*|\mathbf{y}) &= \int p(y_*|\mathbf{w},\sigma^2)p(\mathbf{w},\alpha,\sigma^2|\mathbf{y})\; d\mathbf{w}\; d\alpha\; d\sigma^2 \\
&= \int p(y_*|\mathbf{w},\sigma^2)p(\mathbf{w}|\alpha,\sigma^2,\mathbf{y})p(\alpha,\sigma^2|\mathbf{y})\; d\mathbf{w}\; d\alpha\; d\sigma^2 \\
&\approx \int p(y_*|\mathbf{w},\sigma^2)p(\mathbf{w}|\alpha,\sigma^2,\mathbf{y})\delta(\alpha_{MP},\sigma^2_{MP})\; d\mathbf{w}\; d\alpha\; d\sigma^2 \\
&= \int p(y_*|\mathbf{w},\sigma^2)p(\mathbf{w}|\alpha_{MP},\sigma^2_{MP},\mathbf{y})\; d\mathbf{w}
\end{aligned}
$$

Approximating the Predictive Distribution

- Making prediction for new data points

$$
\begin{aligned}
p(y_*|\mathbf{y}) &= \int p(y_*|\mathbf{w},\sigma^2)p(\mathbf{w},\alpha,\sigma^2|\mathbf{y})\; d\mathbf{w}\; d\alpha\; d\sigma^2 \\
&= \int p(y_*|\mathbf{w},\sigma^2)p(\mathbf{w}|\alpha,\sigma^2,\mathbf{y})p(\alpha,\sigma^2|\mathbf{y})\; d\mathbf{w}\; d\alpha\; d\sigma^2 \\
&\approx \int p(y_*|\mathbf{w},\sigma^2)p(\mathbf{w}|\alpha,\sigma^2,\mathbf{y})\delta(\alpha_{MP},\sigma^2_{MP})\; d\mathbf{w}\; d\alpha\; d\sigma^2 \\
&= \int p(y_*|\mathbf{w},\sigma^2)p(\mathbf{w}|\alpha_{MP},\sigma^2_{MP},\mathbf{y})\; d\mathbf{w}
\end{aligned}
$$

- Recall: $p(\mathbf{w}|\alpha_{MP},\sigma^2_{MP},\mathbf{y})$ is a Gaussian; so is $p(y_*|\mathbf{w},\sigma^2)$

Approximating the Predictive Distribution

- Making prediction for new data points

$$
\begin{aligned}
p(y_*|\mathbf{y}) &= \int p(y_*|\mathbf{w},\sigma^2)p(\mathbf{w},\alpha,\sigma^2|\mathbf{y})\ d\mathbf{w}\ d\alpha\ d\sigma^2 \\
&= \int p(y_*|\mathbf{w},\sigma^2)p(\mathbf{w}|\alpha,\sigma^2,\mathbf{y})p(\alpha,\sigma^2|\mathbf{y})\ d\mathbf{w}\ d\alpha\ d\sigma^2 \\
&\approx \int p(y_*|\mathbf{w},\sigma^2)p(\mathbf{w}|\alpha,\sigma^2,\mathbf{y})\delta(\alpha_{MP},\sigma^2_{MP})\ d\mathbf{w}\ d\alpha\ d\sigma^2 \\
&= \int p(y_*|\mathbf{w},\sigma^2)p(\mathbf{w}|\alpha_{MP},\sigma^2_{MP},\mathbf{y})\ d\mathbf{w}
\end{aligned}
$$

- Recall: $p(\mathbf{w}|\alpha_{MP},\sigma^2_{MP},\mathbf{y})$ is a Gaussian; so is $p(y_*|\mathbf{w},\sigma^2)$
- Can thus now compute $p(y_*|\mathbf{y}) = \int p(y_*|\mathbf{w},\sigma^2)p(\mathbf{w}|\alpha_{MP},\sigma^2_{MP},\mathbf{y})\ d\mathbf{w}$, which is again a Gaussian $\mathcal{N}(y_*|\mu_*,\sigma^2_*)$

$$
\begin{aligned}
\mu_* &= f(\mathbf{x}_*,\mathbf{w}) \\
{\sigma_*}^2 &= {\sigma_{MP}}^2 + \phi(\mathbf{x}_*)^\top \mathbf{\Sigma}\phi(\mathbf{x}_*)
\end{aligned}
$$

Marginal Likelihood

- Hyperparameters α, σ^2 are estimated by maximizing the marginal likelihood
- Marginal likelihood (averaged over the prior on $\mathbf{w}$) is

$$
\begin{aligned}
p(\mathbf{y}|\alpha,\sigma^2) &= \int p(\mathbf{y}|\mathbf{w},\sigma^2)p(\mathbf{w}|\alpha)d\alpha \\
&= \frac{1}{(2\pi)^{N/2}}|\sigma^2\mathbf{I} + \mathbf{\Phi}\mathbf{A}^{-1}\mathbf{\Phi}^\top|^{-1/2}\exp(-\frac{1}{2}\mathbf{y}^\top(\sigma^2\mathbf{I} + \mathbf{\Phi}\mathbf{A}^{-1}\mathbf{\Phi}^\top|^{-1}\mathbf{y})
\end{aligned}
$$

Marginal Likelihood

- Hyperparameters α, σ^2 are estimated by maximizing the marginal likelihood
- Marginal likelihood (averaged over the prior on **w**) is

$$
\begin{aligned}
p(\mathbf{y}|\alpha,\sigma^2) &= \int p(\mathbf{y}|\mathbf{w},\sigma^2)p(\mathbf{w}|\alpha)d\alpha \\
&= \frac{1}{(2\pi)^{N/2}}|\sigma^2\mathbf{I} + \boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^\top|^{-1/2}\exp(-\frac{1}{2}\mathbf{y}^\top(\sigma^2\mathbf{I} + \boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^\top|^{-1}\mathbf{y})
\end{aligned}
$$

- Maximizing $p(\mathbf{y}|\alpha,\sigma^2)$ w.r.t. α and σ^2 gives α_{MP} and σ^2_{MP}, respectively
- Maximization can be done using gradient-based methods

Marginal Likelihood

- Hyperparameters α, σ^2 are estimated by maximizing the marginal likelihood
- Marginal likelihood (averaged over the prior on **w**) is

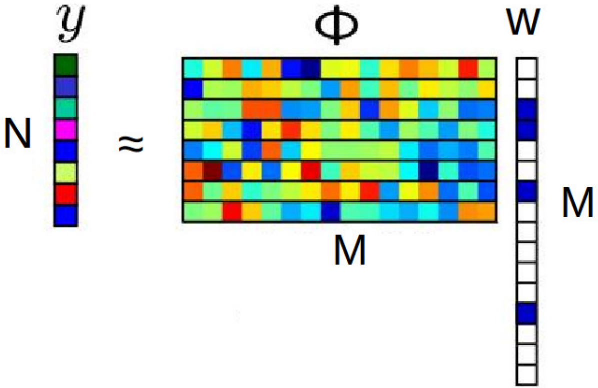
$$p(\mathbf{y}|\alpha,\sigma^2) = \int p(\mathbf{y}|\mathbf{w},\sigma^2)p(\mathbf{w}|\alpha)d\alpha$$
$$= \frac{1}{(2\pi)^{N/2}}|\sigma^2\mathbf{I} + \mathbf{\Phi}\mathbf{A}^{-1}\mathbf{\Phi}^\top|^{-1/2}\exp(-\frac{1}{2}\mathbf{y}^\top(\sigma^2\mathbf{I} + \mathbf{\Phi}\mathbf{A}^{-1}\mathbf{\Phi}^\top|^{-1}\mathbf{y})$$

- Maximizing $p(\mathbf{y}|\alpha,\sigma^2)$ w.r.t. α and σ^2 gives α_{MP} and σ^2_{MP}, respectively
- Maximization can be done using gradient-based methods
- Can assume uniform priors on α, σ^2 and compute marginal model probability

$$p(\mathbf{y}|\mathcal{M}) = \int p(\mathbf{y}|\alpha,\sigma^2)p(\alpha)p(\sigma^2)d\alpha d\sigma^2$$

$$p(\mathbf{y}|\mathcal{M}) \approx \frac{1}{S}\sum_{s=1}^{S} p(\mathbf{y}|\alpha_s,\sigma_s^2)$$ (useful for model-selection)

Sparse Modeling



- Want very few elements in **w** to be nonzero

Sparse Bayesian Regression

- Recall the Gaussian prior on $\mathbf{w}$

$$p(\mathbf{w}|\alpha) = \prod_{m=1}^{M} p(w_m|\alpha) = \prod_{m=1}^{M} \left(\frac{\alpha}{2\pi}\right)^{1/2} \exp\left(-\frac{\alpha}{2} w_m^2\right)$$

- Each component of $\mathbf{w}$ is a zero-mean Gaussian $p(w_m|\alpha) = \mathcal{N}(w_m|0, \alpha^{-1})$
- Same hyperparameter α on each entry of $\mathbf{w}$. Can't impose sparsity on $\mathbf{w}$

Sparse Bayesian Regression

- Recall the Gaussian prior on $\mathbf{w}$

$$p(\mathbf{w}|\alpha) = \prod_{m=1}^{M} p(w_m|\alpha) = \prod_{m=1}^{M} \left(\frac{\alpha}{2\pi}\right)^{1/2} \exp\left(-\frac{\alpha}{2} w_m^2\right)$$

- Each component of $\mathbf{w}$ is a zero-mean Gaussian $p(w_m|\alpha) = \mathcal{N}(w_m|0, \alpha^{-1})$
- Same hyperparameter α on each entry of $\mathbf{w}$. Can't impose sparsity on $\mathbf{w}$
- Let's have a separate inverse variance α_m for each component of $\mathbf{w}$

$$p(\mathbf{w}|\boldsymbol{\alpha}) = \prod_{m=1}^{M} p(w_m|\alpha_m) = \prod_{m=1}^{M} \left(\frac{\alpha_m}{2\pi}\right)^{1/2} \exp\left(-\frac{\alpha_m}{2} w_m^2\right)$$

- We now have M hyperparameters $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_M]$ individually controlling the variance of each component w_m of $\mathbf{w}$

A Hierarchical Prior

- Our new hierarchical prior on $\mathbf{w}$

$$p(\mathbf{w}|\boldsymbol{\alpha}) = \prod_{m=1}^{M} p(w_m|\alpha_m) = \prod_{m=1}^{M} \left(\frac{\alpha_m}{2\pi}\right)^{1/2} \exp\left(-\frac{\alpha_m}{2} w_m^2\right)$$

- We will assume a gamma prior on α_m: $p(\alpha_m) \propto \alpha_m^{a-1} \exp^{-\alpha_m/b}$

A Hierarchical Prior

- Our new hierarchical prior on $\mathbf{w}$

$$p(\mathbf{w}|\boldsymbol{\alpha}) = \prod_{m=1}^{M} p(w_m|\alpha_m) = \prod_{m=1}^{M} \left(\frac{\alpha_m}{2\pi}\right)^{1/2} \exp\left(-\frac{\alpha_m}{2} w_m^2\right)$$

- We will assume a gamma prior on α_m: $p(\alpha_m) \propto \alpha_m^{a-1} \exp^{-\alpha_m/b}$
- The marginal prior on each weight w_m after averaging over $p(\alpha_m)$

$$p(w_m) = \int p(w_m|\alpha_m)p(\alpha_m)d\alpha_m \qquad \text{(will be a Student-t distribution)}$$

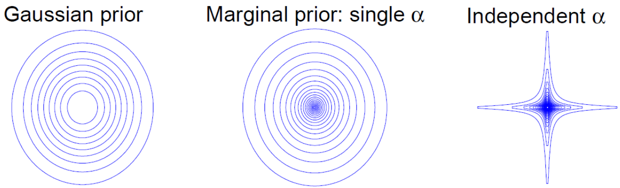
A Hierarchical Prior

- Our new hierarchical prior on $\mathbf{w}$

$$p(\mathbf{w}|\boldsymbol{\alpha}) = \prod_{m=1}^{M} p(w_m|\alpha_m) = \prod_{m=1}^{M} \left(\frac{\alpha_m}{2\pi}\right)^{1/2} \exp\left(-\frac{\alpha_m}{2} w_m^2\right)$$

- We will assume a gamma prior on α_m: $p(\alpha_m) \propto \alpha_m^{a-1} \exp^{-\alpha_m/b}$
- The marginal prior on each weight w_m after averaging over $p(\alpha_m)$

$$p(w_m) = \int p(w_m|\alpha_m)p(\alpha_m)d\alpha_m \qquad \text{(will be a Student-t distribution)}$$



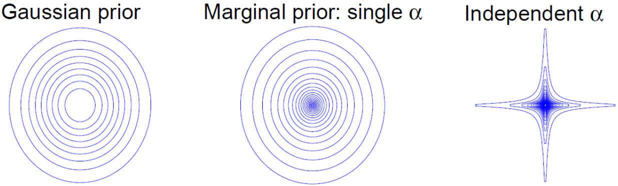
A Hierarchical Prior

- Our new hierarchical prior on $\mathbf{w}$

$$p(\mathbf{w}|\boldsymbol{\alpha}) = \prod_{m=1}^{M} p(w_m|\alpha_m) = \prod_{m=1}^{M} \left(\frac{\alpha_m}{2\pi}\right)^{1/2} \exp\left(-\frac{\alpha_m}{2} w_m^2\right)$$

- We will assume a gamma prior on α_m: $p(\alpha_m) \propto \alpha_m^{a-1} \exp^{-\alpha_m/b}$
- The marginal prior on each weight w_m after averaging over $p(\alpha_m)$

$$p(w_m) = \int p(w_m|\alpha_m)p(\alpha_m)d\alpha_m \qquad \text{(will be a Student-t distribution)}$$



- Akin to penalizing $\sum_{m=1}^{M} \log |w_m|$. Leads to sparse solutions for $\mathbf{w}$

Sparse Bayesian Regression

- Likelihood model

$$p(\mathbf{y}|\mathbf{w}, \sigma^2) = (2\pi\sigma^2)^{-N/2} \exp\left\{-\frac{1}{2\sigma^2}||\mathbf{y} - \mathbf{\Phi}\boldsymbol{\mu}||^2\right\}$$

- Prior on **w**: Gaussian-gamma (Student-t)
- Posterior

$$p(\mathbf{w}, \alpha, \sigma^2|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{w}, \alpha, \sigma^2)p(\mathbf{w}, \alpha, \sigma^2)}{p(\mathbf{y})}$$

Sparse Bayesian Regression

- Likelihood model

$$p(\mathbf{y}|\mathbf{w},\sigma^2) = (2\pi\sigma^2)^{-N/2} \exp\left\{-\frac{1}{2\sigma^2}||\mathbf{y} - \mathbf{\Phi}\boldsymbol{\mu}||^2\right\}$$

- Prior on $\mathbf{w}$: Gaussian-gamma (Student-t)
- Posterior

$$p(\mathbf{w},\alpha,\sigma^2|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{w},\alpha,\sigma^2)p(\mathbf{w},\alpha,\sigma^2)}{p(\mathbf{y})}$$

- Posterior $p(\mathbf{w},\alpha,\sigma^2|\mathbf{y})$ is further decomposed as

$$p(\mathbf{w},\alpha,\sigma^2|\mathbf{y}) = p(\mathbf{w}|\mathbf{y},\alpha,\sigma^2)p(\alpha,\sigma^2|\mathbf{y})$$

The Posterior

- Posterior over weights will be Gaussian

$$
\begin{aligned}
p(\mathbf{w}|\mathbf{y},\alpha,\sigma^2) &= \frac{p(\mathbf{y}|\mathbf{w},\sigma^2)p(\mathbf{w}|\alpha)}{p(\mathbf{y}|\alpha,\sigma^2)} \\
&= (2\pi)^{(N+1)/2}|\boldsymbol{\Sigma}|^{-1/2}\exp\left\{-\frac{1}{2}(\mathbf{w}-\boldsymbol{\mu})\boldsymbol{\Sigma}^{-1}(\mathbf{w}-\boldsymbol{\mu})\right\}
\end{aligned}
$$

where $\boldsymbol{\Sigma} = (\sigma^{-2}\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \mathbf{A})^{-1}$, $\boldsymbol{\mu} = \sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{\Phi}^\top\mathbf{y}$, $\mathbf{A} = \text{diag}(\alpha_1, \alpha_2, \ldots, \alpha_M)$

- Note: if $\alpha_m = \infty$ then $\mu_m = 0$

Hyperparameter Re-estimation

- Posterior over $\mathbf{w}$: $p(\mathbf{w}|\mathbf{y},\alpha,\sigma^2) = \mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})$
- Marginal likelihood (averaged over the prior on $\mathbf{w}$) is

$$
\begin{aligned}
p(\mathbf{y}|\alpha,\sigma^2) &= \int p(\mathbf{y}|\mathbf{w},\sigma^2)p(\mathbf{w}|\alpha)d\alpha \\
&= \frac{1}{(2\pi)^{N/2}}|\sigma^2\mathbf{I}+\boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^\top|^{-1/2}\exp(-\frac{1}{2}\mathbf{y}^\top(\sigma^2\mathbf{I}+\boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^\top|^{-1}\mathbf{y})
\end{aligned}
$$

Hyperparameter Re-estimation

- Posterior over $\mathbf{w}$: $p(\mathbf{w}|\mathbf{y}, \alpha, \sigma^2) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$
- Marginal likelihood (averaged over the prior on $\mathbf{w}$) is

$$\begin{aligned} p(\mathbf{y}|\alpha, \sigma^2) &= \int p(\mathbf{y}|\mathbf{w}, \sigma^2) p(\mathbf{w}|\alpha) d\alpha \\ &= \frac{1}{(2\pi)^{N/2}} |\sigma^2 \mathbf{I} + \boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^\top|^{-1/2} \exp(-\frac{1}{2}\mathbf{y}^\top (\sigma^2 \mathbf{I} + \boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^\top|^{-1}\mathbf{y}) \end{aligned}$$

- Maximize the marginal likelihood $p(\mathbf{y}|\alpha, \sigma^2)$ w.r.t. $\alpha = [\alpha_1, \ldots, \alpha_M]$ and σ^2

$$\begin{aligned} \alpha_m^{new} &= \frac{\gamma_m}{\mu_m^2} \\ (\sigma^2)^{new} &= \frac{||\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\mu}||^2}{N - \sum_{m=1}^{M} \gamma_m} \end{aligned}$$

where $\gamma_m = 1 - \alpha_m \boldsymbol{\Sigma}_{mm}$

- Alternate between estimating $\mathbf{w}$, α, and σ^2

Approximate Bayesian Inference

Bayesian learning routinely needs to deal with intractable integrals, e.g.,

- **Normalization:** when computing the posterior distribution

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})} = \frac{p(\mathcal{D}|\theta)p(\theta)}{\int p(\mathcal{D}|\theta)p(\theta)d\theta}$$

  where the denominator is rarely available in closed analytical form

Approximate Bayesian Inference

Bayesian learning routinely needs to deal with intractable integrals, e.g.,

- **Normalization:** when computing the posterior distribution

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})} = \frac{p(\mathcal{D}|\theta)p(\theta)}{\int p(\mathcal{D}|\theta)p(\theta)d\theta}$$

  where the denominator is rarely available in closed analytical form

- **Marginalization:**

$$p(\theta|\mathcal{D}) = \int p(\theta, \phi|\mathcal{D})p(\phi)d\phi$$

Approximate Bayesian Inference

Bayesian learning routinely needs to deal with intractable integrals, e.g.,

- **Normalization:** when computing the posterior distribution

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})} = \frac{p(\mathcal{D}|\theta)p(\theta)}{\int p(\mathcal{D}|\theta)p(\theta)d\theta}$$

  where the denominator is rarely available in closed analytical form

- **Marginalization:**

$$p(\theta|\mathcal{D}) = \int p(\theta, \phi|\mathcal{D})p(\phi)d\phi$$

- **Expectations:**

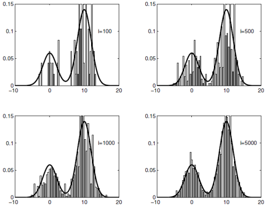
$$\mathbb{E}_{p(\theta|\mathcal{D})}[f(\mathbf{x})] = \int f(\mathbf{x})p(\theta|\mathcal{D})d\theta$$

Approximate Bayesian Inference

- Several ways to do approximate inference in Bayesian models

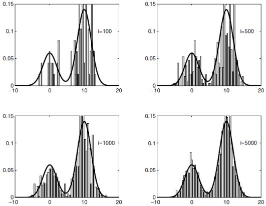
Approximate Bayesian Inference

- Several ways to do approximate inference in Bayesian models
  - **Sampling based approximations:** Monte Carlo methods, Markov-Chain Monte Carlo (MCMC) methods (e.g., Gibbs sampling)



Approximate Bayesian Inference

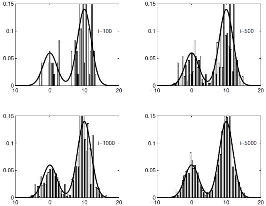
- Several ways to do approximate inference in Bayesian models
  - **Sampling based approximations:** Monte Carlo methods, Markov-Chain Monte Carlo (MCMC) methods (e.g., Gibbs sampling)



  - **Deterministic approximations:** Laplace approximation, Variational Bayes (VB), Expectation Propagation (EP). Treats inference as an optimization problem of finding the parameters of the closest distribution from a family.

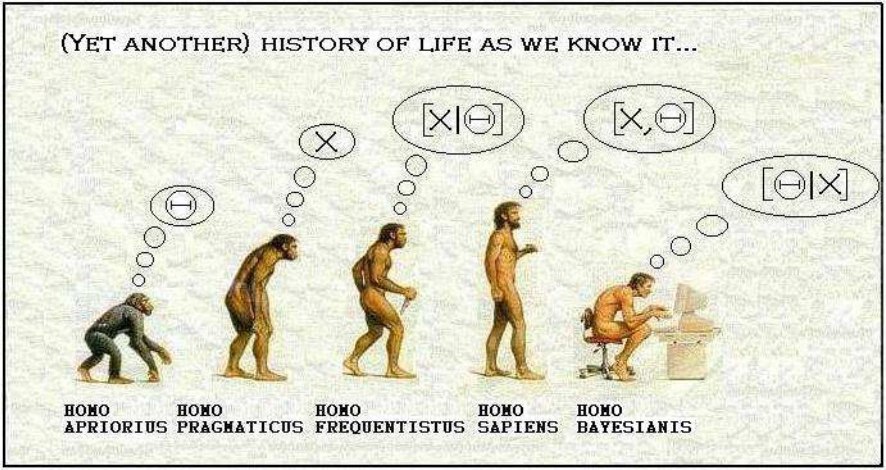
Approximate Bayesian Inference

- Several ways to do approximate inference in Bayesian models
  - **Sampling based approximations:** Monte Carlo methods, Markov-Chain Monte Carlo (MCMC) methods (e.g., Gibbs sampling)



  - **Deterministic approximations:** Laplace approximation, Variational Bayes (VB), Expectation Propagation (EP). Treats inference as an optimization problem of finding the parameters of the closest distribution from a family.
- A very active area of research, lot of recent work on scalable inference (online and distributed Bayesian inference)

Being Bayesian



Other Recent Advances in Bayesian Learning

- Bayesian Optimization
  - Used for optimization problems where the objective function is unknown and expensive to evaluate

Other Recent Advances in Bayesian Learning

- Bayesian Optimization
  - Used for optimization problems where the objective function is unknown and expensive to evaluate
- Closed connections to other "hot" areas in ML, e.g.,
  - Dropout in Deep Learning vs approximate Bayesian inference

Other Recent Advances in Bayesian Learning

- Bayesian Optimization
  - Used for optimization problems where the objective function is unknown and expensive to evaluate
- Closed connections to other "hot" areas in ML, e.g.,
  - Dropout in Deep Learning vs approximate Bayesian inference
- A lot of ongoing work to automate Bayesian inference
  - Probabilistic Programming: computer programs to express probabilistic models

Other Recent Advances in Bayesian Learning

- Bayesian Optimization
  - Used for optimization problems where the objective function is unknown and expensive to evaluate
- Closed connections to other "hot" areas in ML, e.g.,
  - Dropout in Deep Learning vs approximate Bayesian inference
- A lot of ongoing work to automate Bayesian inference
  - Probabilistic Programming: computer programs to express probabilistic models
- Nonparametric Bayesian modeling (or "letting the data speak for itself")

Next Talk

- Introduction to nonparametric Bayesian modeling
- Nonparametric Bayesian regression: Gaussian Process (GP) regression

Thanks! Questions?