Hyperparameter Estimation in Probabilistic Models

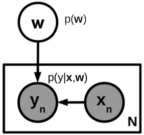
Piyush Rai

Probabilistic Machine Learning (CS772A)

Aug 24, 2017

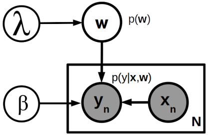
Hyperparameters

- An example: Recall the linear regression model with Gaussian likelihood and Gaussian prior



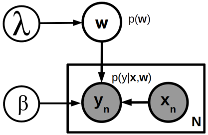
Hyperparameters

- An example: Recall the linear regression model with Gaussian likelihood and Gaussian prior



Hyperparameters

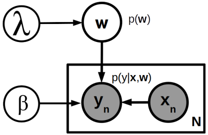
- An example: Recall the linear regression model with Gaussian likelihood and Gaussian prior



- Prior $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\mathbf{0}, \lambda^{-1}\mathbf{I}_D)$ and likelihood $p(y|\boldsymbol{x}, \boldsymbol{w}) = \mathcal{N}(y|\boldsymbol{w}^\top \boldsymbol{x}, \beta^{-1})$ have hyperparameters

Hyperparameters

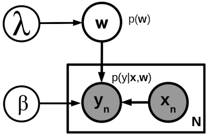
- An example: Recall the linear regression model with Gaussian likelihood and Gaussian prior



- Prior $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\mathbf{0}, \lambda^{-1}\mathbf{I}_D)$ and likelihood $p(y|\boldsymbol{x}, \boldsymbol{w}) = \mathcal{N}(y|\boldsymbol{w}^\top \boldsymbol{x}, \beta^{-1})$ have hyperparameters
- We've seen how to learn $\boldsymbol{w}$ (with λ, β known) in this model using MLE/MAP/Bayesian inference

Hyperparameters

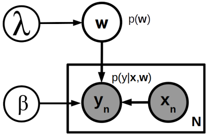
- An example: Recall the linear regression model with Gaussian likelihood and Gaussian prior



- Prior $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\mathbf{0}, \lambda^{-1}\mathbf{I}_D)$ and likelihood $p(y|\boldsymbol{x}, \boldsymbol{w}) = \mathcal{N}(y|\boldsymbol{w}^\top \boldsymbol{x}, \beta^{-1})$ have hyperparameters
- We've seen how to learn $\boldsymbol{w}$ (with λ, β known) in this model using MLE/MAP/Bayesian inference
- Hyperparameters are crucial for any model. E.g., for the above model, they help to regularize

Hyperparameters

- An example: Recall the linear regression model with Gaussian likelihood and Gaussian prior

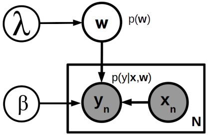


- Prior $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\mathbf{0}, \lambda^{-1}\mathbf{I}_D)$ and likelihood $p(y|\boldsymbol{x}, \boldsymbol{w}) = \mathcal{N}(y|\boldsymbol{w}^\top\boldsymbol{x}, \beta^{-1})$ have hyperparameters
- We've seen how to learn $\boldsymbol{w}$ (with λ, β known) in this model using MLE/MAP/Bayesian inference
- Hyperparameters are crucial for any model. E.g., for the above model, they help to regularize

$$L(\boldsymbol{w}, \beta, \lambda) = \log p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}, \beta, \lambda) = \left\{ \sum_{n=1}^{N} (y_n - \boldsymbol{w}^\top \boldsymbol{x}_n)^2 + \frac{\lambda}{\beta} \boldsymbol{w}^\top \boldsymbol{w} \right\}$$

Hyperparameters

- An example: Recall the linear regression model with Gaussian likelihood and Gaussian prior



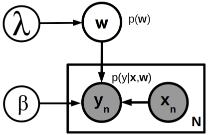
- Prior $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{0}, \lambda^{-1}\mathbf{I}_D)$ and likelihood $p(y|\boldsymbol{x}, \boldsymbol{w}) = \mathcal{N}(y|\boldsymbol{w}^\top \boldsymbol{x}, \beta^{-1})$ have hyperparameters
- We've seen how to learn $\boldsymbol{w}$ (with λ, β known) in this model using MLE/MAP/Bayesian inference
- Hyperparameters are crucial for any model. E.g., for the above model, they help to regularize

$$L(\boldsymbol{w}, \beta, \lambda) = \log p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}, \beta, \lambda) = \left\{ \sum_{n=1}^{N} (y_n - \boldsymbol{w}^\top \boldsymbol{x}_n)^2 + \frac{\lambda}{\beta} \boldsymbol{w}^\top \boldsymbol{w} \right\}$$

- Need a good way of setting the hyperparameters λ and β to their "optimal" values

Hyperparameters

- An example: Recall the linear regression model with Gaussian likelihood and Gaussian prior



- Prior $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\mathbf{0}, \lambda^{-1}\mathbf{I}_D)$ and likelihood $p(y|\boldsymbol{x}, \boldsymbol{w}) = \mathcal{N}(y|\boldsymbol{w}^\top \boldsymbol{x}, \beta^{-1})$ have hyperparameters
- We've seen how to learn $\boldsymbol{w}$ (with λ, β known) in this model using MLE/MAP/Bayesian inference
- Hyperparameters are crucial for any model. E.g., for the above model, they help to regularize

$$L(\boldsymbol{w}, \beta, \lambda) = \log p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}, \beta, \lambda) = \left\{ \sum_{n=1}^{N} (y_n - \boldsymbol{w}^\top \boldsymbol{x}_n)^2 + \frac{\lambda}{\beta} \boldsymbol{w}^\top \boldsymbol{w} \right\}$$

- Need a good way of setting the hyperparameters λ and β to their "optimal" values
- Can we learn these hyperparameters from data?

Why Learn Hyperparameters?

- Can avoid cross-validation (CV). CV wastes training data and is time-consuming

Why Learn Hyperparameters?

- Can avoid cross-validation (CV). CV wastes training data and is time-consuming
- CV can become especially difficult when the number of hyperparameters is very large

Why Learn Hyperparameters?

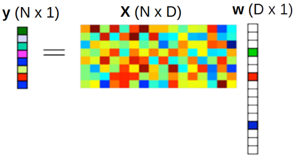
- Can avoid cross-validation (CV). CV wastes training data and is time-consuming
- CV can become especially difficult when the number of hyperparameters is very large
  - Example: Sparse Bayesian regression has component-specific regularization on $\boldsymbol{w}$

$$p(\boldsymbol{w}) = \prod_{d=1}^{D} \mathcal{N}(w_d|0, \lambda_d^{-1}) \qquad (D \text{ unknown hyperparams for } p(\boldsymbol{w}))$$

Why Learn Hyperparameters?

- Can avoid cross-validation (CV). CV wastes training data and is time-consuming
- CV can become especially difficult when the number of hyperparameters is very large
  - Example: Sparse Bayesian regression has component-specific regularization on $\boldsymbol{w}$

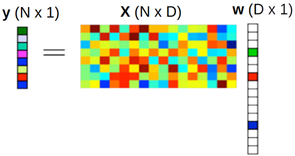
$$p(\boldsymbol{w}) = \prod_{d=1}^{D} \mathcal{N}(w_d | 0, \lambda_d^{-1}) \qquad (D \text{ unknown hyperparams for } p(\boldsymbol{w}))$$



Why Learn Hyperparameters?

- Can avoid cross-validation (CV). CV wastes training data and is time-consuming
- CV can become especially difficult when the number of hyperparameters is very large
  - Example: Sparse Bayesian regression has component-specific regularization on $\boldsymbol{w}$

$$p(\boldsymbol{w}) = \prod_{d=1}^{D} \mathcal{N}(w_d|0, \lambda_d^{-1}) \qquad (D \text{ unknown hyperparams for } p(\boldsymbol{w}))$$



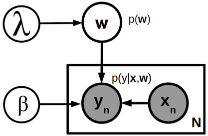
- Tuning is usually impossible to do in unsupervised learning problems (e.g., clustering)

Learning Hyperparameters in Probabilistic Models

- Can treat hyperparams as just a bunch of additional parameters
- Can be learned using a suitable inference algorithm (point estimation or fully Bayesian)

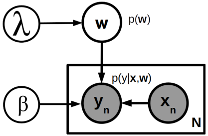
Learning Hyperparameters in Probabilistic Models

- Can treat hyperparams as just a bunch of additional parameters
- Can be learned using a suitable inference algorithm (point estimation or fully Bayesian)
- Example: For the linear regression model, the full set of parameters would be $(\boldsymbol{w}, \lambda, \beta)$



Learning Hyperparameters in Probabilistic Models

- Can treat hyperparams as just a bunch of additional parameters
- Can be learned using a suitable inference algorithm (point estimation or fully Bayesian)
- Example: For the linear regression model, the full set of parameters would be $(\boldsymbol{w}, \lambda, \beta)$

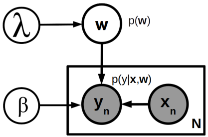


- Can assume priors on all these parameters and infer their "joint" posterior distribution

$$p(\boldsymbol{w}, \beta, \lambda | \mathbf{X}, \boldsymbol{y}) = \frac{p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}, \beta, \lambda) p(\boldsymbol{w}, \lambda, \beta)}{p(\boldsymbol{y}|\mathbf{X})} = \frac{p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}, \beta, \lambda) p(\boldsymbol{w}|\lambda) p(\beta) p(\lambda)}{\int p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}, \beta) p(\boldsymbol{w}|\lambda) p(\beta) p(\lambda) \, d\boldsymbol{w} \, d\lambda d\beta}$$

Learning Hyperparameters in Probabilistic Models

- Can treat hyperparams as just a bunch of additional parameters
- Can be learned using a suitable inference algorithm (point estimation or fully Bayesian)
- Example: For the linear regression model, the full set of parameters would be $(\boldsymbol{w}, \lambda, \beta)$



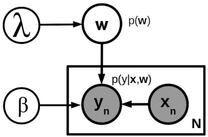
- Can assume priors on all these parameters and infer their "joint" posterior distribution

$$p(\boldsymbol{w}, \beta, \lambda | \mathbf{X}, \boldsymbol{y}) = \frac{p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}, \beta, \lambda) p(\boldsymbol{w}, \lambda, \beta)}{p(\boldsymbol{y}|\mathbf{X})} = \frac{p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}, \beta, \lambda) p(\boldsymbol{w}|\lambda) p(\beta) p(\lambda)}{\int p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}, \beta) p(\boldsymbol{w}|\lambda) p(\beta) p(\lambda) \; d\boldsymbol{w} \; d\lambda d\beta}$$

- Infering the above is usually intractable (rare to have conjugacy).

Learning Hyperparameters in Probabilistic Models

- Can treat hyperparams as just a bunch of additional parameters
- Can be learned using a suitable inference algorithm (point estimation or fully Bayesian)
- Example: For the linear regression model, the full set of parameters would be $(\boldsymbol{w}, \lambda, \beta)$



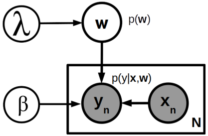
- Can assume priors on all these parameters and infer their "joint" posterior distribution

$$p(\boldsymbol{w}, \beta, \lambda | \mathbf{X}, \boldsymbol{y}) = \frac{p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}, \beta, \lambda)p(\boldsymbol{w}, \lambda, \beta)}{p(\boldsymbol{y}|\mathbf{X})} = \frac{p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}, \beta, \lambda)p(\boldsymbol{w}|\lambda)p(\beta)p(\lambda)}{\int p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}, \beta)p(\boldsymbol{w}|\lambda)p(\beta)p(\lambda)\ d\boldsymbol{w}\ d\lambda d\beta}$$

- Infering the above is usually intractable (rare to have conjugacy). Requires approximations.

Learning Hyperparameters in Probabilistic Models

- Can treat hyperparams as just a bunch of additional parameters
- Can be learned using a suitable inference algorithm (point estimation or fully Bayesian)
- Example: For the linear regression model, the full set of parameters would be $(\boldsymbol{w}, \lambda, \beta)$



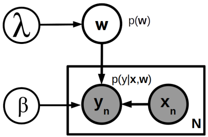
- Can assume priors on all these parameters and infer their "joint" posterior distribution

$$p(\boldsymbol{w}, \beta, \lambda | \mathbf{X}, \boldsymbol{y}) = \frac{p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}, \beta, \lambda) p(\boldsymbol{w}, \lambda, \beta)}{p(\boldsymbol{y}|\mathbf{X})} = \frac{p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}, \beta, \lambda) p(\boldsymbol{w}|\lambda) p(\beta) p(\lambda)}{\int p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}, \beta) p(\boldsymbol{w}|\lambda) p(\beta) p(\lambda) \, d\boldsymbol{w} \, d\lambda d\beta}$$

- Infering the above is usually intractable (rare to have conjugacy). Requires approximations. Also,

Learning Hyperparameters in Probabilistic Models

- Can treat hyperparams as just a bunch of additional parameters
- Can be learned using a suitable inference algorithm (point estimation or fully Bayesian)
- Example: For the linear regression model, the full set of parameters would be $(\boldsymbol{w}, \lambda, \beta)$



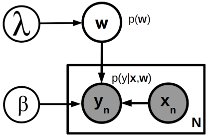
- Can assume priors on all these parameters and infer their "joint" posterior distribution

$$p(\boldsymbol{w}, \beta, \lambda|\mathbf{X}, \boldsymbol{y}) = \frac{p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}, \beta, \lambda)p(\boldsymbol{w}, \lambda, \beta)}{p(\boldsymbol{y}|\mathbf{X})} = \frac{p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}, \beta, \lambda)p(\boldsymbol{w}|\lambda)p(\beta)p(\lambda)}{\int p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}, \beta)p(\boldsymbol{w}|\lambda)p(\beta)p(\lambda)\ d\boldsymbol{w}\ d\lambda d\beta}$$

- Infering the above is usually intractable (rare to have conjugacy). Requires approximations. Also,
  - What priors (or "hyperpriors") to choose for β and λ?

Learning Hyperparameters in Probabilistic Models

- Can treat hyperparams as just a bunch of additional parameters
- Can be learned using a suitable inference algorithm (point estimation or fully Bayesian)
- Example: For the linear regression model, the full set of parameters would be $(\boldsymbol{w}, \lambda, \beta)$



- Can assume priors on all these parameters and infer their "joint" posterior distribution

$$p(\boldsymbol{w}, \beta, \lambda | \mathbf{X}, \boldsymbol{y}) = \frac{p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}, \beta, \lambda) p(\boldsymbol{w}, \lambda, \beta)}{p(\boldsymbol{y}|\mathbf{X})} = \frac{p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}, \beta, \lambda) p(\boldsymbol{w}|\lambda) p(\beta) p(\lambda)}{\int p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}, \beta) p(\boldsymbol{w}|\lambda) p(\beta) p(\lambda) \ d\boldsymbol{w} \ d\lambda d\beta}$$

- Infering the above is usually intractable (rare to have conjugacy). Requires approximations. Also,
  - What priors (or "hyperpriors") to choose for β and λ?
  - What about the hyperparameters of those priors?

Learning Hyperparameters via Point Estimation

- One popular way to estimate hyperparameters is by maximizing the marginal likelihood

Learning Hyperparameters via Point Estimation

- One popular way to estimate hyperparameters is by maximizing the marginal likelihood
- For our linear regression model, this quantity (a function of the hyperparams) will be

$$p(\boldsymbol{y}|\mathbf{X}, \beta, \lambda) = \int p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}, \beta)p(\boldsymbol{w}|\lambda)d\boldsymbol{w}$$

Learning Hyperparameters via Point Estimation

- One popular way to estimate hyperparameters is by maximizing the marginal likelihood
- For our linear regression model, this quantity (a function of the hyperparams) will be

$$p(\boldsymbol{y}|\mathbf{X}, \beta, \lambda) = \int p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}, \beta)p(\boldsymbol{w}|\lambda)d\boldsymbol{w}$$

- The "optimal" hyperparameters in this case can be then found by

$$\hat{\beta}, \hat{\lambda} = \arg\max_{\beta,\lambda} \log p(\boldsymbol{y}|\mathbf{X}, \beta, \lambda)$$

Learning Hyperparameters via Point Estimation

- One popular way to estimate hyperparameters is by maximizing the marginal likelihood
- For our linear regression model, this quantity (a function of the hyperparams) will be

$$p(\boldsymbol{y}|\mathbf{X}, \beta, \lambda) = \int p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}, \beta)p(\boldsymbol{w}|\lambda)d\boldsymbol{w}$$

- The "optimal" hyperparameters in this case can be then found by

$$\hat{\beta}, \hat{\lambda} = \arg\max_{\beta,\lambda} \log p(\boldsymbol{y}|\mathbf{X}, \beta, \lambda)$$

- This is called MLE-II or (log) evidence maximization

Learning Hyperparameters via Point Estimation

- One popular way to estimate hyperparameters is by maximizing the marginal likelihood
- For our linear regression model, this quantity (a function of the hyperparams) will be

$$p(\boldsymbol{y}|\mathbf{X},\beta,\lambda) = \int p(\boldsymbol{y}|\mathbf{X},\boldsymbol{w},\beta)p(\boldsymbol{w}|\lambda)d\boldsymbol{w}$$

- The "optimal" hyperparameters in this case can be then found by

$$\hat{\beta},\hat{\lambda} = \arg\max_{\beta,\lambda} \log p(\boldsymbol{y}|\mathbf{X},\beta,\lambda)$$

- This is called MLE-II or (log) evidence maximization
  - Akin to doing MLE to estimate the hyperparameters where the "main" parameter (in this case $\boldsymbol{w}$) has been integrated out from the model's likelihood function

Learning Hyperparameters via Point Estimation

- One popular way to estimate hyperparameters is by maximizing the marginal likelihood
- For our linear regression model, this quantity (a function of the hyperparams) will be

$$p(\boldsymbol{y}|\mathbf{X},\beta,\lambda) = \int p(\boldsymbol{y}|\mathbf{X},\boldsymbol{w},\beta)p(\boldsymbol{w}|\lambda)d\boldsymbol{w}$$

- The "optimal" hyperparameters in this case can be then found by

$$\hat{\beta},\hat{\lambda} = \arg\max_{\beta,\lambda} \log p(\boldsymbol{y}|\mathbf{X},\beta,\lambda)$$

- This is called MLE-II or (log) evidence maximization
  - Akin to doing MLE to estimate the hyperparameters where the "main" parameter (in this case $\boldsymbol{w}$) has been integrated out from the model's likelihood function
- Note: If the likelihood and prior are conjugate then marginal likelihood is available in closed form

What is MLE-II Doing?

- For linear regression case, would ideally like the posterior over all unknowns, i.e., $p(\boldsymbol{w}, \lambda, \beta | \mathbf{X}, \boldsymbol{y})$

$$p(\boldsymbol{w}, \beta, \lambda | \mathbf{X}, \boldsymbol{y}) = p(\boldsymbol{w} | \mathbf{X}, \boldsymbol{y}, \beta, \lambda) p(\beta, \lambda | \mathbf{X}, \boldsymbol{y}) \qquad \text{(from product rule)}$$

What is MLE-II Doing?

- For linear regression case, would ideally like the posterior over all unknowns, i.e., $p(\boldsymbol{w}, \lambda, \beta | \mathbf{X}, \boldsymbol{y})$

$$p(\boldsymbol{w}, \beta, \lambda | \mathbf{X}, \boldsymbol{y}) = p(\boldsymbol{w} | \mathbf{X}, \boldsymbol{y}, \beta, \lambda) p(\beta, \lambda | \mathbf{X}, \boldsymbol{y}) \qquad \text{(from product rule)}$$

- Note that $p(\boldsymbol{w} | \mathbf{X}, \boldsymbol{y}, \beta, \lambda)$ is easy if λ, β are known

What is MLE-II Doing?

- For linear regression case, would ideally like the posterior over all unknowns, i.e., $p(\boldsymbol{w}, \lambda, \beta | \mathbf{X}, \boldsymbol{y})$

$$p(\boldsymbol{w}, \beta, \lambda | \mathbf{X}, \boldsymbol{y}) = p(\boldsymbol{w} | \mathbf{X}, \boldsymbol{y}, \beta, \lambda) p(\beta, \lambda | \mathbf{X}, \boldsymbol{y}) \qquad \text{(from product rule)}$$

- Note that $p(\boldsymbol{w} | \mathbf{X}, \boldsymbol{y}, \beta, \lambda)$ is easy if λ, β are known
- However $p(\beta, \lambda | \mathbf{X}, \boldsymbol{y}) = \frac{p(\boldsymbol{y} | \mathbf{X}, \beta, \alpha) p(\beta) p(\lambda)}{p(\boldsymbol{y} | \mathbf{X})}$ is hard

What is MLE-II Doing?

- For linear regression case, would ideally like the posterior over all unknowns, i.e., $p(\boldsymbol{w}, \lambda, \beta|\mathbf{X}, \boldsymbol{y})$

$$p(\boldsymbol{w}, \beta, \lambda|\mathbf{X}, \boldsymbol{y}) = p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}, \beta, \lambda)p(\beta, \lambda|\mathbf{X}, \boldsymbol{y}) \qquad \text{(from product rule)}$$

- Note that $p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}, \beta, \lambda)$ is easy if λ, β are known
- However $p(\beta, \lambda|\mathbf{X}, \boldsymbol{y}) = \frac{p(\boldsymbol{y}|\mathbf{X}, \beta, \alpha)p(\beta)p(\lambda)}{p(\boldsymbol{y}|\mathbf{X})}$ is hard (lack of conjugacy, intractable denominator)

What is MLE-II Doing?

- For linear regression case, would ideally like the posterior over all unknowns, i.e., $p(\boldsymbol{w}, \lambda, \beta | \mathbf{X}, \boldsymbol{y})$

$$p(\boldsymbol{w}, \beta, \lambda | \mathbf{X}, \boldsymbol{y}) = p(\boldsymbol{w} | \mathbf{X}, \boldsymbol{y}, \beta, \lambda) p(\beta, \lambda | \mathbf{X}, \boldsymbol{y}) \qquad \text{(from product rule)}$$

- Note that $p(\boldsymbol{w} | \mathbf{X}, \boldsymbol{y}, \beta, \lambda)$ is easy if λ, β are known
- However $p(\beta, \lambda | \mathbf{X}, \boldsymbol{y}) = \frac{p(\boldsymbol{y} | \mathbf{X}, \beta, \alpha) p(\beta) p(\lambda)}{p(\boldsymbol{y} | \mathbf{X})}$ is hard (lack of conjugacy, intractable denominator)
- Let's approximate it by a point function δ at the mode of $p(\beta, \lambda | \mathbf{X}, \boldsymbol{y})$

$$p(\beta, \lambda | \mathbf{X}, \boldsymbol{y}) \approx \delta(\hat{\beta}, \hat{\lambda})$$

What is MLE-II Doing?

- For linear regression case, would ideally like the posterior over all unknowns, i.e., $p(\boldsymbol{w}, \lambda, \beta | \mathbf{X}, \boldsymbol{y})$

$$p(\boldsymbol{w}, \beta, \lambda | \mathbf{X}, \boldsymbol{y}) = p(\boldsymbol{w} | \mathbf{X}, \boldsymbol{y}, \beta, \lambda) p(\beta, \lambda | \mathbf{X}, \boldsymbol{y}) \qquad \text{(from product rule)}$$

- Note that $p(\boldsymbol{w} | \mathbf{X}, \boldsymbol{y}, \beta, \lambda)$ is easy if λ, β are known
- However $p(\beta, \lambda | \mathbf{X}, \boldsymbol{y}) = \frac{p(\boldsymbol{y} | \mathbf{X}, \beta, \alpha) p(\beta) p(\lambda)}{p(\boldsymbol{y} | \mathbf{X})}$ is hard (lack of conjugacy, intractable denominator)
- Let's approximate it by a point function δ at the mode of $p(\beta, \lambda | \mathbf{X}, \boldsymbol{y})$

$$p(\beta, \lambda | \mathbf{X}, \boldsymbol{y}) \approx \delta(\hat{\beta}, \hat{\lambda}) \quad \text{where} \quad \hat{\beta}, \hat{\lambda} = \arg\max_{\beta, \lambda} p(\beta, \lambda | \mathbf{X}, \boldsymbol{y}) = \arg\max_{\beta, \lambda} p(\boldsymbol{y} | \mathbf{X}, \beta, \lambda) p(\lambda) p(\beta)$$

What is MLE-II Doing?

- For linear regression case, would ideally like the posterior over all unknowns, i.e., $p(\boldsymbol{w}, \lambda, \beta | \mathbf{X}, \boldsymbol{y})$

$$p(\boldsymbol{w}, \beta, \lambda | \mathbf{X}, \boldsymbol{y}) = p(\boldsymbol{w} | \mathbf{X}, \boldsymbol{y}, \beta, \lambda) p(\beta, \lambda | \mathbf{X}, \boldsymbol{y}) \qquad \text{(from product rule)}$$

- Note that $p(\boldsymbol{w} | \mathbf{X}, \boldsymbol{y}, \beta, \lambda)$ is easy if λ, β are known
- However $p(\beta, \lambda | \mathbf{X}, \boldsymbol{y}) = \frac{p(\boldsymbol{y}|\mathbf{X}, \beta, \alpha) p(\beta) p(\lambda)}{p(\boldsymbol{y}|\mathbf{X})}$ is hard (lack of conjugacy, intractable denominator)
- Let's approximate it by a point function δ at the mode of $p(\beta, \lambda | \mathbf{X}, \boldsymbol{y})$

$$p(\beta, \lambda | \mathbf{X}, \boldsymbol{y}) \approx \delta(\hat{\beta}, \hat{\lambda}) \quad \text{where} \quad \hat{\beta}, \hat{\lambda} = \arg\max_{\beta, \lambda} p(\beta, \lambda | \mathbf{X}, \boldsymbol{y}) = \arg\max_{\beta, \lambda} p(\boldsymbol{y} | \mathbf{X}, \beta, \lambda) p(\lambda) p(\beta)$$

- Moreover, if $p(\beta)$, $p(\lambda)$ are uniform/uninformative priors then

$$\hat{\beta}, \hat{\lambda} = \arg\max_{\beta, \lambda} p(\boldsymbol{y} | \mathbf{X}, \beta, \lambda)$$

What is MLE-II Doing?

- For linear regression case, would ideally like the posterior over all unknowns, i.e., $p(\boldsymbol{w}, \lambda, \beta | \mathbf{X}, \boldsymbol{y})$

$$p(\boldsymbol{w}, \beta, \lambda | \mathbf{X}, \boldsymbol{y}) = p(\boldsymbol{w} | \mathbf{X}, \boldsymbol{y}, \beta, \lambda) p(\beta, \lambda | \mathbf{X}, \boldsymbol{y}) \qquad \text{(from product rule)}$$

- Note that $p(\boldsymbol{w} | \mathbf{X}, \boldsymbol{y}, \beta, \lambda)$ is easy if λ, β are known
- However $p(\beta, \lambda | \mathbf{X}, \boldsymbol{y}) = \frac{p(\boldsymbol{y} | \mathbf{X}, \beta, \alpha) p(\beta) p(\lambda)}{p(\boldsymbol{y} | \mathbf{X})}$ is hard (lack of conjugacy, intractable denominator)
- Let's approximate it by a point function δ at the mode of $p(\beta, \lambda | \mathbf{X}, \boldsymbol{y})$

$$p(\beta, \lambda | \mathbf{X}, \boldsymbol{y}) \approx \delta(\hat{\beta}, \hat{\lambda}) \quad \text{where} \quad \hat{\beta}, \hat{\lambda} = \arg\max_{\beta, \lambda} p(\beta, \lambda | \mathbf{X}, \boldsymbol{y}) = \arg\max_{\beta, \lambda} p(\boldsymbol{y} | \mathbf{X}, \beta, \lambda) p(\lambda) p(\beta)$$

- Moreover, if $p(\beta)$, $p(\lambda)$ are uniform/uninformative priors then

$$\hat{\beta}, \hat{\lambda} = \arg\max_{\beta, \lambda} p(\boldsymbol{y} | \mathbf{X}, \beta, \lambda)$$

- Thus MLE-II is approximating the posterior of hyperparams by their point estimate assuming uniform priors (therefore we don't need to worry about a prior over the hyperparams)

MLE-II for Linear Regression

- For the linear regression case, the marginal likelihood is defined as

$$p(\boldsymbol{y}|\mathbf{X},\beta,\lambda) = \int p(\boldsymbol{y}|\mathbf{X},\boldsymbol{w},\beta)p(\boldsymbol{w}|\lambda)d\boldsymbol{w}$$

where $p(\boldsymbol{y}|\mathbf{X},\boldsymbol{w},\beta) = \mathcal{N}(\boldsymbol{y}|\mathbf{X}\boldsymbol{w},\beta^{-1}\mathbf{I}_N)$ and $p(\boldsymbol{w}|\lambda) = \mathcal{N}(\boldsymbol{w}|0,\lambda^{-1}\mathbf{I}_D)$

MLE-II for Linear Regression

- For the linear regression case, the marginal likelihood is defined as

$$p(\boldsymbol{y}|\mathbf{X}, \beta, \lambda) = \int p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}, \beta) p(\boldsymbol{w}|\lambda) d\boldsymbol{w}$$

where $p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}, \beta) = \mathcal{N}(\boldsymbol{y}|\mathbf{X}\boldsymbol{w}, \beta^{-1}\mathbf{I}_N)$ and $p(\boldsymbol{w}|\lambda) = \mathcal{N}(\boldsymbol{w}|0, \lambda^{-1}\mathbf{I}_D)$

- The above can be computed in closed form, using properties of Gaussians (see below)

MLE-II for Linear Regression

- For the linear regression case, the marginal likelihood is defined as

$$p(\boldsymbol{y}|\mathbf{X},\beta,\lambda) = \int p(\boldsymbol{y}|\mathbf{X},\boldsymbol{w},\beta)p(\boldsymbol{w}|\lambda)d\boldsymbol{w}$$

  where $p(\boldsymbol{y}|\mathbf{X},\boldsymbol{w},\beta) = \mathcal{N}(\boldsymbol{y}|\mathbf{X}\boldsymbol{w},\beta^{-1}\mathbf{I}_N)$ and $p(\boldsymbol{w}|\lambda) = \mathcal{N}(\boldsymbol{w}|0,\lambda^{-1}\mathbf{I}_D)$

- The above can be computed in closed form, using properties of Gaussians (see below)

$$p(\mathbf{t}|\boldsymbol{z},\mathbf{A},\mathbf{b},\mathbf{L}) = \mathcal{N}(\mathbf{A}\boldsymbol{z}+\mathbf{b},\mathbf{L}^{-1})$$

MLE-II for Linear Regression

- For the linear regression case, the marginal likelihood is defined as

$$p(\boldsymbol{y}|\mathbf{X},\beta,\lambda) = \int p(\boldsymbol{y}|\mathbf{X},\boldsymbol{w},\beta)p(\boldsymbol{w}|\lambda)d\boldsymbol{w}$$

where $p(\boldsymbol{y}|\mathbf{X},\boldsymbol{w},\beta) = \mathcal{N}(\boldsymbol{y}|\mathbf{X}\boldsymbol{w},\beta^{-1}\mathbf{I}_N)$ and $p(\boldsymbol{w}|\lambda) = \mathcal{N}(\boldsymbol{w}|0,\lambda^{-1}\mathbf{I}_D)$

- The above can be computed in closed form, using properties of Gaussians (see below)

$$\begin{aligned} p(\mathbf{t}|\boldsymbol{z},\mathbf{A},\mathbf{b},\mathbf{L}) &= \mathcal{N}(\mathbf{A}\boldsymbol{z}+\mathbf{b},\mathbf{L}^{-1}) \\ p(\boldsymbol{z}|\boldsymbol{\mu},\boldsymbol{\Lambda}) &= \mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu},\boldsymbol{\Lambda}^{-1}) \end{aligned}$$

MLE-II for Linear Regression

- For the linear regression case, the marginal likelihood is defined as

$$p(\boldsymbol{y}|\mathbf{X},\beta,\lambda) = \int p(\boldsymbol{y}|\mathbf{X},\boldsymbol{w},\beta)p(\boldsymbol{w}|\lambda)d\boldsymbol{w}$$

where $p(\boldsymbol{y}|\mathbf{X},\boldsymbol{w},\beta) = \mathcal{N}(\boldsymbol{y}|\mathbf{X}\boldsymbol{w},\beta^{-1}\mathbf{I}_N)$ and $p(\boldsymbol{w}|\lambda) = \mathcal{N}(\boldsymbol{w}|0,\lambda^{-1}\mathbf{I}_D)$

- The above can be computed in closed form, using properties of Gaussians (see below)

$$\begin{aligned}
p(\mathbf{t}|\boldsymbol{z},\mathbf{A},\mathbf{b},\mathbf{L}) &= \mathcal{N}(\mathbf{A}\boldsymbol{z}+\mathbf{b},\mathbf{L}^{-1}) \\
p(\boldsymbol{z}|\boldsymbol{\mu},\boldsymbol{\Lambda}) &= \mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu},\boldsymbol{\Lambda}^{-1}) \\
\Rightarrow p(\mathbf{t}|\mathbf{A},\mathbf{b},\mathbf{L},\boldsymbol{\mu},\boldsymbol{\Lambda}) &= \int p(\mathbf{t}|\boldsymbol{z},\mathbf{A},\mathbf{b},\mathbf{L})p(\boldsymbol{z}|\boldsymbol{\mu},\boldsymbol{\Lambda})d\boldsymbol{z}
\end{aligned}$$

MLE-II for Linear Regression

- For the linear regression case, the marginal likelihood is defined as

$$p(\boldsymbol{y}|\mathbf{X}, \beta, \lambda) = \int p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}, \beta)p(\boldsymbol{w}|\lambda)d\boldsymbol{w}$$

where $p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}, \beta) = \mathcal{N}(\boldsymbol{y}|\mathbf{X}\boldsymbol{w}, \beta^{-1}\mathbf{I}_N)$ and $p(\boldsymbol{w}|\lambda) = \mathcal{N}(\boldsymbol{w}|0, \lambda^{-1}\mathbf{I}_D)$

- The above can be computed in closed form, using properties of Gaussians (see below)

$$\begin{aligned}
p(\mathbf{t}|\boldsymbol{z}, \mathbf{A}, \mathbf{b}, \mathbf{L}) &= \mathcal{N}(\mathbf{A}\boldsymbol{z} + \mathbf{b}, \mathbf{L}^{-1}) \\
p(\boldsymbol{z}|\boldsymbol{\mu}, \boldsymbol{\Lambda}) &= \mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}) \\
\Rightarrow p(\mathbf{t}|\mathbf{A}, \mathbf{b}, \mathbf{L}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) &= \int p(\mathbf{t}|\boldsymbol{z}, \mathbf{A}, \mathbf{b}, \mathbf{L})p(\boldsymbol{z}|\boldsymbol{\mu}, \boldsymbol{\Lambda})d\boldsymbol{z} = \mathcal{N}(\mathbf{A}\boldsymbol{\mu} + \mathbf{b}, \mathbf{L}^{-1} + \mathbf{A}\boldsymbol{\Lambda}^{-1}\mathbf{A}^\top)
\end{aligned}$$

MLE-II for Linear Regression

- For the linear regression case, the marginal likelihood is defined as

$$p(\boldsymbol{y}|\mathbf{X},\beta,\lambda) = \int p(\boldsymbol{y}|\mathbf{X},\boldsymbol{w},\beta)p(\boldsymbol{w}|\lambda)d\boldsymbol{w}$$

where $p(\boldsymbol{y}|\mathbf{X},\boldsymbol{w},\beta) = \mathcal{N}(\boldsymbol{y}|\mathbf{X}\boldsymbol{w},\beta^{-1}\mathbf{I}_N)$ and $p(\boldsymbol{w}|\lambda) = \mathcal{N}(\boldsymbol{w}|0,\lambda^{-1}\mathbf{I}_D)$

- The above can be computed in closed form, using properties of Gaussians (see below)

$$\begin{aligned}
p(\mathbf{t}|\boldsymbol{z},\mathbf{A},\mathbf{b},\mathbf{L}) &= \mathcal{N}(\mathbf{A}\boldsymbol{z}+\mathbf{b},\mathbf{L}^{-1})\\
p(\boldsymbol{z}|\boldsymbol{\mu},\boldsymbol{\Lambda}) &= \mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu},\boldsymbol{\Lambda}^{-1})\\
\Rightarrow p(\mathbf{t}|\mathbf{A},\mathbf{b},\mathbf{L},\boldsymbol{\mu},\boldsymbol{\Lambda}) &= \int p(\mathbf{t}|\boldsymbol{z},\mathbf{A},\mathbf{b},\mathbf{L})p(\boldsymbol{z}|\boldsymbol{\mu},\boldsymbol{\Lambda})d\boldsymbol{z} \quad = \quad \mathcal{N}(\mathbf{A}\boldsymbol{\mu}+\mathbf{b},\mathbf{L}^{-1}+\mathbf{A}\boldsymbol{\Lambda}^{-1}\mathbf{A}^\top)
\end{aligned}$$

- Using the above result the marginal likelihood will be

$$p(\boldsymbol{y}|\mathbf{X},\beta,\lambda) = \mathcal{N}(\boldsymbol{y}|\mathbf{0},\beta^{-1}\mathbf{I}+\lambda^{-1}\mathbf{X}\mathbf{X}^\top)$$

MLE-II for Linear Regression

- For the linear regression case, the marginal likelihood is defined as

$$p(\boldsymbol{y}|\mathbf{X},\beta,\lambda) = \int p(\boldsymbol{y}|\mathbf{X},\boldsymbol{w},\beta)p(\boldsymbol{w}|\lambda)d\boldsymbol{w}$$

  where $p(\boldsymbol{y}|\mathbf{X},\boldsymbol{w},\beta) = \mathcal{N}(\boldsymbol{y}|\mathbf{X}\boldsymbol{w},\beta^{-1}\mathbf{I}_N)$ and $p(\boldsymbol{w}|\lambda) = \mathcal{N}(\boldsymbol{w}|0,\lambda^{-1}\mathbf{I}_D)$

- The above can be computed in closed form, using properties of Gaussians (see below)

$$\begin{aligned} p(\mathbf{t}|\boldsymbol{z},\mathbf{A},\mathbf{b},\mathbf{L}) &= \mathcal{N}(\mathbf{A}\boldsymbol{z}+\mathbf{b},\mathbf{L}^{-1}) \\ p(\boldsymbol{z}|\boldsymbol{\mu},\boldsymbol{\Lambda}) &= \mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu},\boldsymbol{\Lambda}^{-1}) \\ \Rightarrow p(\mathbf{t}|\mathbf{A},\mathbf{b},\mathbf{L},\boldsymbol{\mu},\boldsymbol{\Lambda}) &= \int p(\mathbf{t}|\boldsymbol{z},\mathbf{A},\mathbf{b},\mathbf{L})p(\boldsymbol{z}|\boldsymbol{\mu},\boldsymbol{\Lambda})d\boldsymbol{z} \quad = \quad \mathcal{N}(\mathbf{A}\boldsymbol{\mu}+\mathbf{b},\mathbf{L}^{-1}+\mathbf{A}\boldsymbol{\Lambda}^{-1}\mathbf{A}^\top) \end{aligned}$$

- Using the above result the marginal likelihood will be

$$p(\boldsymbol{y}|\mathbf{X},\beta,\lambda) = \mathcal{N}(\boldsymbol{y}|\mathbf{0},\beta^{-1}\mathbf{I}+\lambda^{-1}\mathbf{X}\mathbf{X}^\top) = \frac{1}{(2\pi)^{N/2}}|\beta^{-1}\mathbf{I}+\lambda^{-1}\mathbf{X}\mathbf{X}^\top|^{-1/2}\exp(-\frac{1}{2}\boldsymbol{y}^\top(\beta^{-1}\mathbf{I}+\lambda^{-1}\mathbf{X}\mathbf{X}^\top)^{-1}\boldsymbol{y})$$

MLE-II for Linear Regression

- For the linear regression case, the marginal likelihood is defined as

$$p(\boldsymbol{y}|\mathbf{X},\beta,\lambda) = \int p(\boldsymbol{y}|\mathbf{X},\boldsymbol{w},\beta)p(\boldsymbol{w}|\lambda)d\boldsymbol{w}$$

where $p(\boldsymbol{y}|\mathbf{X},\boldsymbol{w},\beta) = \mathcal{N}(\boldsymbol{y}|\mathbf{X}\boldsymbol{w},\beta^{-1}\mathbf{I}_N)$ and $p(\boldsymbol{w}|\lambda) = \mathcal{N}(\boldsymbol{w}|0,\lambda^{-1}\mathbf{I}_D)$

- The above can be computed in closed form, using properties of Gaussians (see below)

$$\begin{aligned} p(\mathbf{t}|\boldsymbol{z},\mathbf{A},\mathbf{b},\mathbf{L}) &= \mathcal{N}(\mathbf{A}\boldsymbol{z}+\mathbf{b},\mathbf{L}^{-1}) \\ p(\boldsymbol{z}|\boldsymbol{\mu},\boldsymbol{\Lambda}) &= \mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu},\boldsymbol{\Lambda}^{-1}) \\ \Rightarrow p(\mathbf{t}|\mathbf{A},\mathbf{b},\mathbf{L},\boldsymbol{\mu},\boldsymbol{\Lambda}) &= \int p(\mathbf{t}|\boldsymbol{z},\mathbf{A},\mathbf{b},\mathbf{L})p(\boldsymbol{z}|\boldsymbol{\mu},\boldsymbol{\Lambda})d\boldsymbol{z} \quad = \quad \mathcal{N}(\mathbf{A}\boldsymbol{\mu}+\mathbf{b},\mathbf{L}^{-1}+\mathbf{A}\boldsymbol{\Lambda}^{-1}\mathbf{A}^\top) \end{aligned}$$

- Using the above result the marginal likelihood will be

$$p(\boldsymbol{y}|\mathbf{X},\beta,\lambda) = \mathcal{N}(\boldsymbol{y}|\mathbf{0},\beta^{-1}\mathbf{I}+\lambda^{-1}\mathbf{X}\mathbf{X}^\top) = \frac{1}{(2\pi)^{N/2}}|\beta^{-1}\mathbf{I}+\lambda^{-1}\mathbf{X}\mathbf{X}^\top|^{-1/2}\exp(-\frac{1}{2}\boldsymbol{y}^\top(\beta^{-1}\mathbf{I}+\lambda^{-1}\mathbf{X}\mathbf{X}^\top)^{-1}\boldsymbol{y})$$

- MLE-II maximizes $\log p(\boldsymbol{y}|\mathbf{X},\beta,\lambda)$ w.r.t. β and λ to estimate these hyperparams

MLE-II for Linear Regression

- For the linear regression case, the marginal likelihood is defined as

$$p(\boldsymbol{y}|\mathbf{X},\beta,\lambda) = \int p(\boldsymbol{y}|\mathbf{X},\boldsymbol{w},\beta)p(\boldsymbol{w}|\lambda)d\boldsymbol{w}$$

where $p(\boldsymbol{y}|\mathbf{X},\boldsymbol{w},\beta) = \mathcal{N}(\boldsymbol{y}|\mathbf{X}\boldsymbol{w},\beta^{-1}\mathbf{I}_N)$ and $p(\boldsymbol{w}|\lambda) = \mathcal{N}(\boldsymbol{w}|0,\lambda^{-1}\mathbf{I}_D)$

- The above can be computed in closed form, using properties of Gaussians (see below)

$$\begin{aligned}
p(\mathbf{t}|\boldsymbol{z},\mathbf{A},\mathbf{b},\mathbf{L}) &= \mathcal{N}(\mathbf{A}\boldsymbol{z}+\mathbf{b},\mathbf{L}^{-1}) \\
p(\boldsymbol{z}|\boldsymbol{\mu},\boldsymbol{\Lambda}) &= \mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu},\boldsymbol{\Lambda}^{-1}) \\
\Rightarrow p(\mathbf{t}|\mathbf{A},\mathbf{b},\mathbf{L},\boldsymbol{\mu},\boldsymbol{\Lambda}) &= \int p(\mathbf{t}|\boldsymbol{z},\mathbf{A},\mathbf{b},\mathbf{L})p(\boldsymbol{z}|\boldsymbol{\mu},\boldsymbol{\Lambda})d\boldsymbol{z} \quad = \quad \mathcal{N}(\mathbf{A}\boldsymbol{\mu}+\mathbf{b},\mathbf{L}^{-1}+\mathbf{A}\boldsymbol{\Lambda}^{-1}\mathbf{A}^\top)
\end{aligned}$$

- Using the above result the marginal likelihood will be

$$p(\boldsymbol{y}|\mathbf{X},\beta,\lambda) = \mathcal{N}(\boldsymbol{y}|\mathbf{0},\beta^{-1}\mathbf{I}+\lambda^{-1}\mathbf{X}\mathbf{X}^\top) = \frac{1}{(2\pi)^{N/2}}|\beta^{-1}\mathbf{I}+\lambda^{-1}\mathbf{X}\mathbf{X}^\top|^{-1/2}\exp(-\frac{1}{2}\boldsymbol{y}^\top(\beta^{-1}\mathbf{I}+\lambda^{-1}\mathbf{X}\mathbf{X}^\top)^{-1}\boldsymbol{y})$$

- MLE-II maximizes $\log p(\boldsymbol{y}|\mathbf{X},\beta,\lambda)$ w.r.t. β and λ to estimate these hyperparams
    - Usually there is no closed form solution. Solved using iterative/alternating optimization

An Iterative Scheme

We can follow an iterative scheme for estimating $\boldsymbol{w}$, λ, and β

An Iterative Scheme

We can follow an iterative scheme for estimating $\boldsymbol{w}$, λ, and β

- Initialize $\hat{\lambda}$ and $\hat{\beta}$ (can be randomly)

An Iterative Scheme

We can follow an iterative scheme for estimating $\boldsymbol{w}$, λ, and β

- Initialize $\hat{\lambda}$ and $\hat{\beta}$ (can be randomly)
- Repeat until convergence

An Iterative Scheme

We can follow an iterative scheme for estimating $\boldsymbol{w}$, λ, and β

- Initialize $\hat{\lambda}$ and $\hat{\beta}$ (can be randomly)
- Repeat until convergence
  - Estimate the posterior over $\boldsymbol{w}$ as

$$p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}, \hat{\lambda}, \hat{\beta}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

where $\boldsymbol{\Sigma} = (\hat{\beta}\sum_{n=1}^{N} \boldsymbol{x}_n \boldsymbol{x}_n^\top + \hat{\lambda}\mathbf{I}_D)^{-1}$ and $\boldsymbol{\mu} = \boldsymbol{\Sigma}(\hat{\beta}\sum_{n=1}^{N} y_n \boldsymbol{x}_n)$

An Iterative Scheme

We can follow an iterative scheme for estimating $\boldsymbol{w}$, λ, and β

- Initialize $\hat{\lambda}$ and $\hat{\beta}$ (can be randomly)
- Repeat until convergence
  - Estimate the posterior over $\boldsymbol{w}$ as

$$p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}, \hat{\lambda}, \hat{\beta}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

    where $\boldsymbol{\Sigma} = (\hat{\beta}\sum_{n=1}^{N} \boldsymbol{x}_n \boldsymbol{x}_n^\top + \hat{\lambda}\mathbf{I}_D)^{-1}$ and $\boldsymbol{\mu} = \boldsymbol{\Sigma}(\hat{\beta}\sum_{n=1}^{N} y_n \boldsymbol{x}_n)$
  - Re-estimate $\hat{\lambda}$ as

$$\hat{\lambda} = \frac{\gamma}{\boldsymbol{\mu}^\top \boldsymbol{\mu}} \qquad \text{(note: } \gamma \text{ depends on eigenvalues of } \boldsymbol{\Sigma}\text{)}$$

An Iterative Scheme

We can follow an iterative scheme for estimating $\boldsymbol{w}$, λ, and β

- Initialize $\hat{\lambda}$ and $\hat{\beta}$ (can be randomly)
- Repeat until convergence
  - Estimate the posterior over $\boldsymbol{w}$ as

$$p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}, \hat{\lambda}, \hat{\beta}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

  where $\boldsymbol{\Sigma} = (\hat{\beta}\sum_{n=1}^{N} \boldsymbol{x}_n \boldsymbol{x}_n^\top + \hat{\lambda}\mathbf{I}_D)^{-1}$ and $\boldsymbol{\mu} = \boldsymbol{\Sigma}(\hat{\beta}\sum_{n=1}^{N} y_n \boldsymbol{x}_n)$

  - Re-estimate $\hat{\lambda}$ as

$$\hat{\lambda} = \frac{\gamma}{\boldsymbol{\mu}^\top \boldsymbol{\mu}} \qquad \text{(note: } \gamma \text{ depends on eigenvalues of } \boldsymbol{\Sigma}\text{)}$$

  - Re-estimate $\hat{\beta}$ as

$$\hat{\beta} = \frac{N - \gamma}{\sum_{n=1}^{N}(y_n - \boldsymbol{\mu}^\top \boldsymbol{x}_n)^2}$$

An Iterative Scheme

We can follow an iterative scheme for estimating $\boldsymbol{w}$, λ, and β

- Initialize $\hat{\lambda}$ and $\hat{\beta}$ (can be randomly)
- Repeat until convergence
    - Estimate the posterior over $\boldsymbol{w}$ as

$$p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}, \hat{\lambda}, \hat{\beta}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

    where $\boldsymbol{\Sigma} = (\hat{\beta} \sum_{n=1}^{N} \boldsymbol{x}_n \boldsymbol{x}_n^\top + \hat{\lambda} \mathbf{I}_D)^{-1}$ and $\boldsymbol{\mu} = \boldsymbol{\Sigma}(\hat{\beta} \sum_{n=1}^{N} y_n \boldsymbol{x}_n)$

    - Re-estimate $\hat{\lambda}$ as

$$\hat{\lambda} = \frac{\gamma}{\boldsymbol{\mu}^\top \boldsymbol{\mu}} \qquad \text{(note: } \gamma \text{ depends on eigenvalues of } \boldsymbol{\Sigma})$$

    - Re-estimate $\hat{\beta}$ as

$$\hat{\beta} = \frac{N - \gamma}{\sum_{n=1}^{N} (y_n - \boldsymbol{\mu}^\top \boldsymbol{x}_n)^2}$$

Note that the update of one parameter depends on the current estimate of other parameters

Using MLE-II Estimates for Making Prediction

- With the MLE-II approximation $p(\beta,\lambda|\mathbf{X},\boldsymbol{y}) \approx \delta(\hat{\beta},\hat{\lambda})$, the posterior over unknowns

$$p(\boldsymbol{w},\beta,\lambda|\mathbf{X},\boldsymbol{y}) = p(\boldsymbol{w}|\mathbf{X},\boldsymbol{y},\beta,\lambda)p(\beta,\lambda|\mathbf{X},\boldsymbol{y})$$

Using MLE-II Estimates for Making Prediction

- With the MLE-II approximation $p(\beta, \lambda|\mathbf{X}, \boldsymbol{y}) \approx \delta(\hat{\beta}, \hat{\lambda})$, the posterior over unknowns

$$p(\boldsymbol{w}, \beta, \lambda|\mathbf{X}, \boldsymbol{y}) = p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}, \beta, \lambda)p(\beta, \lambda|\mathbf{X}, \boldsymbol{y}) \approx p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}, \hat{\beta}, \hat{\lambda})$$

Using MLE-II Estimates for Making Prediction

- With the MLE-II approximation $p(\beta, \lambda|\mathbf{X}, \boldsymbol{y}) \approx \delta(\hat{\beta}, \hat{\lambda})$, the posterior over unknowns

$$p(\boldsymbol{w}, \beta, \lambda|\mathbf{X}, \boldsymbol{y}) = p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}, \beta, \lambda)p(\beta, \lambda|\mathbf{X}, \boldsymbol{y}) \approx p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}, \hat{\beta}, \hat{\lambda})$$

- The posterior predictive distribution can also be approximated as

$$p(y_*|\boldsymbol{x}_*, \mathbf{X}, \boldsymbol{y}) = \int p(y_*|\boldsymbol{x}_*, \boldsymbol{w}, \beta)p(\boldsymbol{w}, \beta, \lambda|\mathbf{X}, \boldsymbol{y})\, d\boldsymbol{w}\, d\beta\, d\lambda$$

Using MLE-II Estimates for Making Prediction

- With the MLE-II approximation $p(\beta, \lambda|\mathbf{X}, \boldsymbol{y}) \approx \delta(\hat{\beta}, \hat{\lambda})$, the posterior over unknowns

$$p(\boldsymbol{w}, \beta, \lambda|\mathbf{X}, \boldsymbol{y}) = p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}, \beta, \lambda)p(\beta, \lambda|\mathbf{X}, \boldsymbol{y}) \approx p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}, \hat{\beta}, \hat{\lambda})$$

- The posterior predictive distribution can also be approximated as

$$\begin{aligned} p(y_*|\boldsymbol{x}_*, \mathbf{X}, \boldsymbol{y}) &= \int p(y_*|\boldsymbol{x}_*, \boldsymbol{w}, \beta)p(\boldsymbol{w}, \beta, \lambda|\mathbf{X}, \boldsymbol{y})\ d\boldsymbol{w}\ d\beta\ d\lambda \\ &= \int p(y_*|\boldsymbol{x}_*, \boldsymbol{w}, \beta)p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}, \beta, \lambda)p(\beta, \lambda|\mathbf{X}, \boldsymbol{y})d\beta\ d\lambda\ d\boldsymbol{w} \end{aligned}$$

Using MLE-II Estimates for Making Prediction

- With the MLE-II approximation $p(\beta, \lambda|\mathbf{X}, \boldsymbol{y}) \approx \delta(\hat{\beta}, \hat{\lambda})$, the posterior over unknowns

$$p(\boldsymbol{w}, \beta, \lambda|\mathbf{X}, \boldsymbol{y}) = p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}, \beta, \lambda)p(\beta, \lambda|\mathbf{X}, \boldsymbol{y}) \approx p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}, \hat{\beta}, \hat{\lambda})$$

- The posterior predictive distribution can also be approximated as

$$\begin{aligned} p(y_*|\boldsymbol{x}_*, \mathbf{X}, \boldsymbol{y}) &= \int p(y_*|\boldsymbol{x}_*, \boldsymbol{w}, \beta)p(\boldsymbol{w}, \beta, \lambda|\mathbf{X}, \boldsymbol{y})\; d\boldsymbol{w}\; d\beta\; d\lambda \\ &= \int p(y_*|\boldsymbol{x}_*, \boldsymbol{w}, \beta)p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}, \beta, \lambda)p(\beta, \lambda|\mathbf{X}, \boldsymbol{y}) d\beta\; d\lambda\; d\boldsymbol{w} \\ &\approx \int p(y_*|\boldsymbol{x}_*, \boldsymbol{w}, \beta)p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}, \hat{\beta}, \hat{\lambda})\; d\boldsymbol{w} \end{aligned}$$

Using MLE-II Estimates for Making Prediction

- With the MLE-II approximation $p(\beta, \lambda|\mathbf{X}, \boldsymbol{y}) \approx \delta(\hat{\beta}, \hat{\lambda})$, the posterior over unknowns

$$p(\boldsymbol{w}, \beta, \lambda|\mathbf{X}, \boldsymbol{y}) = p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}, \beta, \lambda)p(\beta, \lambda|\mathbf{X}, \boldsymbol{y}) \approx p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}, \hat{\beta}, \hat{\lambda})$$

- The posterior predictive distribution can also be approximated as

$$\begin{aligned} p(y_*|\boldsymbol{x}_*, \mathbf{X}, \boldsymbol{y}) &= \int p(y_*|\boldsymbol{x}_*, \boldsymbol{w}, \beta)p(\boldsymbol{w}, \beta, \lambda|\mathbf{X}, \boldsymbol{y}) \, d\boldsymbol{w} \; d\beta \; d\lambda \\ &= \int p(y_*|\boldsymbol{x}_*, \boldsymbol{w}, \beta)p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}, \beta, \lambda)p(\beta, \lambda|\mathbf{X}, \boldsymbol{y}) d\beta \; d\lambda \; d\boldsymbol{w} \\ &\approx \int p(y_*|\boldsymbol{x}_*, \boldsymbol{w}, \beta)p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}, \hat{\beta}, \hat{\lambda}) \, d\boldsymbol{w} \end{aligned}$$

- This is also the same as the usual posterior predictive distribution we have seen earlier, except we are treating the hyperparams $\hat{\beta}, \hat{\lambda}$ fixed at their MLE-II based estimates

Illustration of Hyperparameter Estimation via MLE-II

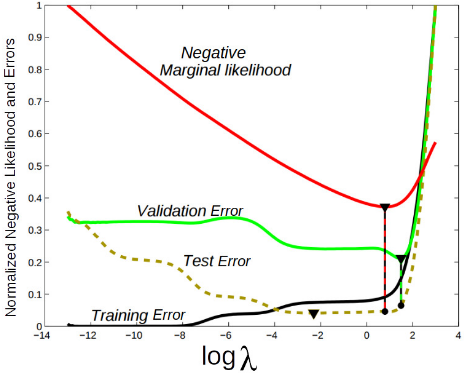
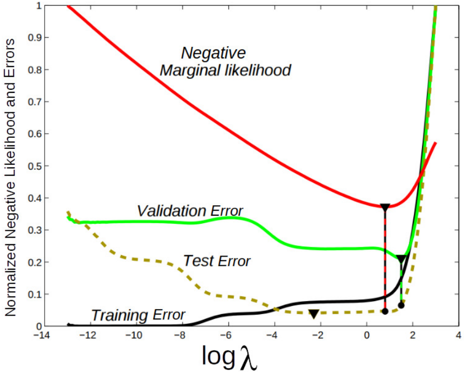


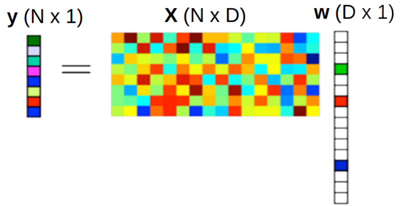
Illustration of Hyperparameter Estimation via MLE-II



Important: Unlike cross-validation, the marginal lik. based MLE-II approach doesn't need a separate held-out data set (MLE-II uses all the training data for estimating the hyperparameters)

Another Example: Sparse Modeling

- In high-dim regression problems, we want very few elements in $\boldsymbol{w}$ to be nonzero

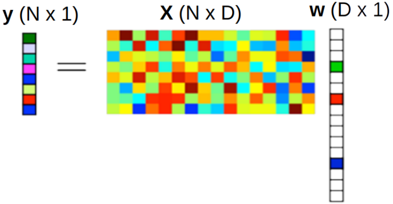


- We can impose component-wise regularization on $\boldsymbol{w}$ via the following prior

$$p(\boldsymbol{w}) = \prod_{d=1}^{D} \mathcal{N}(w_d|0, \lambda_d^{-1}) = \prod_{d=1}^{D} \left(\frac{\lambda_d}{2\pi}\right)^{1/2} \exp\left(-\frac{\lambda_d}{2} w_d^2\right)$$

Another Example: Sparse Modeling

- In high-dim regression problems, we want very few elements in $\boldsymbol{w}$ to be nonzero



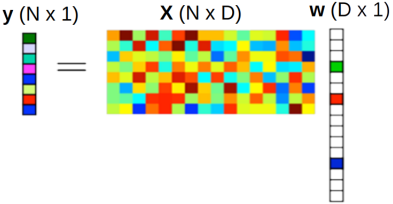
- We can impose component-wise regularization on $\boldsymbol{w}$ via the following prior

$$p(\boldsymbol{w}) = \prod_{d=1}^{D} \mathcal{N}(w_d|0, \lambda_d^{-1}) = \prod_{d=1}^{D} \left(\frac{\lambda_d}{2\pi}\right)^{1/2} \exp\left(-\frac{\lambda_d}{2} w_d^2\right)$$

- With a Gaussian likelihood for $\boldsymbol{y}$, this is called Sparse Bayesian Linear Regression

Another Example: Sparse Modeling

- In high-dim regression problems, we want very few elements in $\boldsymbol{w}$ to be nonzero



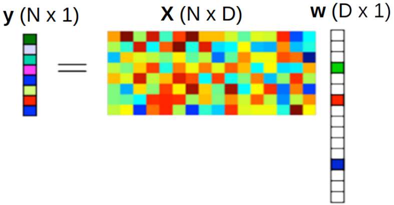
- We can impose component-wise regularization on $\boldsymbol{w}$ via the following prior

$$p(\boldsymbol{w}) = \prod_{d=1}^{D} \mathcal{N}(w_d|0, \lambda_d^{-1}) = \prod_{d=1}^{D} \left(\frac{\lambda_d}{2\pi}\right)^{1/2} \exp\left(-\frac{\lambda_d}{2} w_d^2\right)$$

- With a Gaussian likelihood for $\boldsymbol{y}$, this is called Sparse Bayesian Linear Regression
- Now there are D hyperparameters $\{\lambda_d\}_{d=1}^{D}$ to estimate (plus the noise variance β)

Another Example: Sparse Modeling

- In high-dim regression problems, we want very few elements in $\boldsymbol{w}$ to be nonzero



- We can impose component-wise regularization on $\boldsymbol{w}$ via the following prior

$$p(\boldsymbol{w}) = \prod_{d=1}^{D} \mathcal{N}(w_d|0, \lambda_d^{-1}) = \prod_{d=1}^{D} \left(\frac{\lambda_d}{2\pi}\right)^{1/2} \exp\left(-\frac{\lambda_d}{2} w_d^2\right)$$

- With a Gaussian likelihood for $\boldsymbol{y}$, this is called Sparse Bayesian Linear Regression
- Now there are D hyperparameters $\{\lambda_d\}_{d=1}^{D}$ to estimate (plus the noise variance β)
- We can again MLE-II to estimate these hyperparameters

MLE-II for Sparse Bayesian Linear Regression

- Posterior over $\boldsymbol{w}$: $p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}, \boldsymbol{\lambda}, \beta) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Again, easy to compute in closed form.

MLE-II for Sparse Bayesian Linear Regression

- Posterior over $\boldsymbol{w}$: $p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}, \boldsymbol{\lambda}, \beta) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Again, easy to compute in closed form.
- Marginal likelihood (averaged over the prior on $\boldsymbol{w}$) is

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

where $\mathbf{A} = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_D)$

MLE-II for Sparse Bayesian Linear Regression

- Posterior over $\boldsymbol{w}$: $p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}, \boldsymbol{\lambda}, \beta) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Again, easy to compute in closed form.
- Marginal likelihood (averaged over the prior on $\boldsymbol{w}$) is

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

where $\mathbf{A} = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_D)$

- Maximizing the marginal likelihood $p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{\lambda}, \beta)$ w.r.t. $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_D]$ and β gives

$$\hat{\lambda}_d = \frac{\gamma_d}{\mu_d^2} \qquad \text{where} \quad \gamma_d = 1 - \lambda_d \boldsymbol{\Sigma}_{dd}$$

$$\hat{\beta} = \frac{(N - \sum_{d=1}^{D} \gamma_d)}{||\boldsymbol{y} - \mathbf{X}\boldsymbol{\mu}||^2}$$

MLE-II for Sparse Bayesian Linear Regression

- Posterior over $\boldsymbol{w}$: $p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}, \boldsymbol{\lambda}, \beta) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Again, easy to compute in closed form.
- Marginal likelihood (averaged over the prior on $\boldsymbol{w}$) is

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

where $\mathbf{A} = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_D)$

- Maximizing the marginal likelihood $p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{\lambda}, \beta)$ w.r.t. $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_D]$ and β gives

$$\hat{\lambda}_d = \frac{\gamma_d}{\mu_d^2} \qquad \text{where} \quad \gamma_d = 1 - \lambda_d \boldsymbol{\Sigma}_{dd}$$

$$\hat{\beta} = \frac{(N - \sum_{d=1}^D \gamma_d)}{||\boldsymbol{y} - \mathbf{X}\boldsymbol{\mu}||^2}$$

- We alternate between estimating $\boldsymbol{w}$, $\boldsymbol{\lambda}$, and β, until we converge

Always Estimate Hyperparameters?

Always Estimate Hyperparameters?

- In general yes, but can sometimes give misleading results

Always Estimate Hyperparameters?

- In general yes, but can sometimes give misleading results
- Especially true if your model is grossly mis-specified (very different from the data distribution)

Always Estimate Hyperparameters?

- In general yes, but can sometimes give misleading results
- Especially true if your model is grossly mis-specified (very different from the data distribution)
    - In such a case, cross-validation although more expensive can be more robust

Always Estimate Hyperparameters?

- In general yes, but can sometimes give misleading results
- Especially true if your model is grossly mis-specified (very different from the data distribution)
  - In such a case, cross-validation although more expensive can be more robust
- Therefore use hyperparameter estimation methods with caution :-)

Always Estimate Hyperparameters?

- In general yes, but can sometimes give misleading results
- Especially true if your model is grossly mis-specified (very different from the data distribution)
  - In such a case, cross-validation although more expensive can be more robust
- Therefore use hyperparameter estimation methods with caution :-)
- Nevertheless, hyperparameter estimation is general is a powerful idea

Always Estimate Hyperparameters?

- In general yes, but can sometimes give misleading results
- Especially true if your model is grossly mis-specified (very different from the data distribution)
  - In such a case, cross-validation although more expensive can be more robust
- Therefore use hyperparameter estimation methods with caution :-)
- Nevertheless, hyperparameter estimation is general is a powerful idea
  - Really shines in unsupervised learning problems where cross-validation is not possible

Other Approaches for Hyperparameter Estimation

We mainly looked at point estimates for finding the best hyperparameters. However, there are other approach to hyperparameter estimation in probabilistic models

Other Approaches for Hyperparameter Estimation

We mainly looked at point estimates for finding the best hyperparameters. However, there are other approach to hyperparameter estimation in probabilistic models

- Infer the full posterior over all the parameters and the hyperparameters

Other Approaches for Hyperparameter Estimation

We mainly looked at point estimates for finding the best hyperparameters. However, there are other approach to hyperparameter estimation in probabilistic models

- Infer the full posterior over all the parameters and the hyperparameters
  - No real distinction anymore as to what's a parameter and what's a hyperparameter

Other Approaches for Hyperparameter Estimation

We mainly looked at point estimates for finding the best hyperparameters. However, there are other approach to hyperparameter estimation in probabilistic models

- Infer the full posterior over all the parameters and the hyperparameters
  - No real distinction anymore as to what's a parameter and what's a hyperparameter
  - Usually done via iterative methods.

Other Approaches for Hyperparameter Estimation

We mainly looked at point estimates for finding the best hyperparameters. However, there are other approach to hyperparameter estimation in probabilistic models

- Infer the full posterior over all the parameters and the hyperparameters
  - No real distinction anymore as to what's a parameter and what's a hyperparameter
  - Usually done via iterative methods. Inference boils down to estimating the posterior of one parameter conditioned on the current values of the other parameters, in an alternating fashion

Other Approaches for Hyperparameter Estimation

We mainly looked at point estimates for finding the best hyperparameters. However, there are other approach to hyperparameter estimation in probabilistic models

- Infer the full posterior over all the parameters and the hyperparameters
  - No real distinction anymore as to what's a parameter and what's a hyperparameter
  - Usually done via iterative methods. Inference boils down to estimating the posterior of one parameter conditioned on the current values of the other parameters, in an alternating fashion
  - More we discuss approximate inference methods such as MCMC (e.g., Gibbs sampling) and VB

Other Approaches for Hyperparameter Estimation

We mainly looked at point estimates for finding the best hyperparameters. However, there are other approach to hyperparameter estimation in probabilistic models

- Infer the full posterior over all the parameters and the hyperparameters
  - No real distinction anymore as to what's a parameter and what's a hyperparameter
  - Usually done via iterative methods. Inference boils down to estimating the posterior of one parameter conditioned on the current values of the other parameters, in an alternating fashion
  - More we discuss approximate inference methods such as MCMC (e.g., Gibbs sampling) and VB
- Using Bayesian Optimization (BO): Learn the cross-validation error function and find its minima

Other Approaches for Hyperparameter Estimation

We mainly looked at point estimates for finding the best hyperparameters. However, there are other approach to hyperparameter estimation in probabilistic models

- Infer the full posterior over all the parameters and the hyperparameters
  - No real distinction anymore as to what's a parameter and what's a hyperparameter
  - Usually done via iterative methods. Inference boils down to estimating the posterior of one parameter conditioned on the current values of the other parameters, in an alternating fashion
  - More we discuss approximate inference methods such as MCMC (e.g., Gibbs sampling) and VB
- Using Bayesian Optimization (BO): Learn the cross-validation error function and find its minima
  - The function depends on the hyperparameters but we don't know what it is

Other Approaches for Hyperparameter Estimation

We mainly looked at point estimates for finding the best hyperparameters. However, there are other approach to hyperparameter estimation in probabilistic models

- Infer the full posterior over all the parameters and the hyperparameters
    - No real distinction anymore as to what's a parameter and what's a hyperparameter
    - Usually done via iterative methods. Inference boils down to estimating the posterior of one parameter conditioned on the current values of the other parameters, in an alternating fashion
    - More we discuss approximate inference methods such as MCMC (e.g., Gibbs sampling) and VB
- Using Bayesian Optimization (BO): Learn the cross-validation error function and find its minima
    - The function depends on the hyperparameters but we don't know what it is
    - Can learn the function using its values at some choices of the hyperparameters

Other Approaches for Hyperparameter Estimation

We mainly looked at point estimates for finding the best hyperparameters. However, there are other approach to hyperparameter estimation in probabilistic models

- Infer the full posterior over all the parameters and the hyperparameters
  - No real distinction anymore as to what's a parameter and what's a hyperparameter
  - Usually done via iterative methods. Inference boils down to estimating the posterior of one parameter conditioned on the current values of the other parameters, in an alternating fashion
  - More we discuss approximate inference methods such as MCMC (e.g., Gibbs sampling) and VB
- Using Bayesian Optimization (BO): Learn the cross-validation error function and find its minima
  - The function depends on the hyperparameters but we don't know what it is
  - Can learn the function using its values at some choices of the hyperparameters
  - BO can perform the function learning and optima finding simultaneously and efficiently

Other Approaches for Hyperparameter Estimation

We mainly looked at point estimates for finding the best hyperparameters. However, there are other approach to hyperparameter estimation in probabilistic models

- Infer the full posterior over all the parameters and the hyperparameters
  - No real distinction anymore as to what's a parameter and what's a hyperparameter
  - Usually done via iterative methods. Inference boils down to estimating the posterior of one parameter conditioned on the current values of the other parameters, in an alternating fashion
  - More we discuss approximate inference methods such as MCMC (e.g., Gibbs sampling) and VB
- Using Bayesian Optimization (BO): Learn the cross-validation error function and find its minima
  - The function depends on the hyperparameters but we don't know what it is
  - Can learn the function using its values at some choices of the hyperparameters
  - BO can perform the function learning and optima finding simultaneously and efficiently
  - A lot of recent interest in BO for hyperparameter estimation

Other Approaches for Hyperparameter Estimation

We mainly looked at point estimates for finding the best hyperparameters. However, there are other approach to hyperparameter estimation in probabilistic models

- Infer the full posterior over all the parameters and the hyperparameters
  - No real distinction anymore as to what's a parameter and what's a hyperparameter
  - Usually done via iterative methods. Inference boils down to estimating the posterior of one parameter conditioned on the current values of the other parameters, in an alternating fashion
  - More we discuss approximate inference methods such as MCMC (e.g., Gibbs sampling) and VB
- Using Bayesian Optimization (BO): Learn the cross-validation error function and find its minima
  - The function depends on the hyperparameters but we don't know what it is
  - Can learn the function using its values at some choices of the hyperparameters
  - BO can perform the function learning and optima finding simultaneously and efficiently
  - A lot of recent interest in BO for hyperparameter estimation
  - We will discuss this later during the semester