Probabilistic Linear Regression

CS698X: Topics in Probabilistic Modeling and Inference

Piyush Rai

Probabilistic Linear Regression

Note: Only $\boldsymbol{y_n}$ being modeled, not $\boldsymbol{x_n}$ (discriminative model). A conditional model where $\boldsymbol{y_n}$ being modeled, conditioned on $\boldsymbol{x_n}$

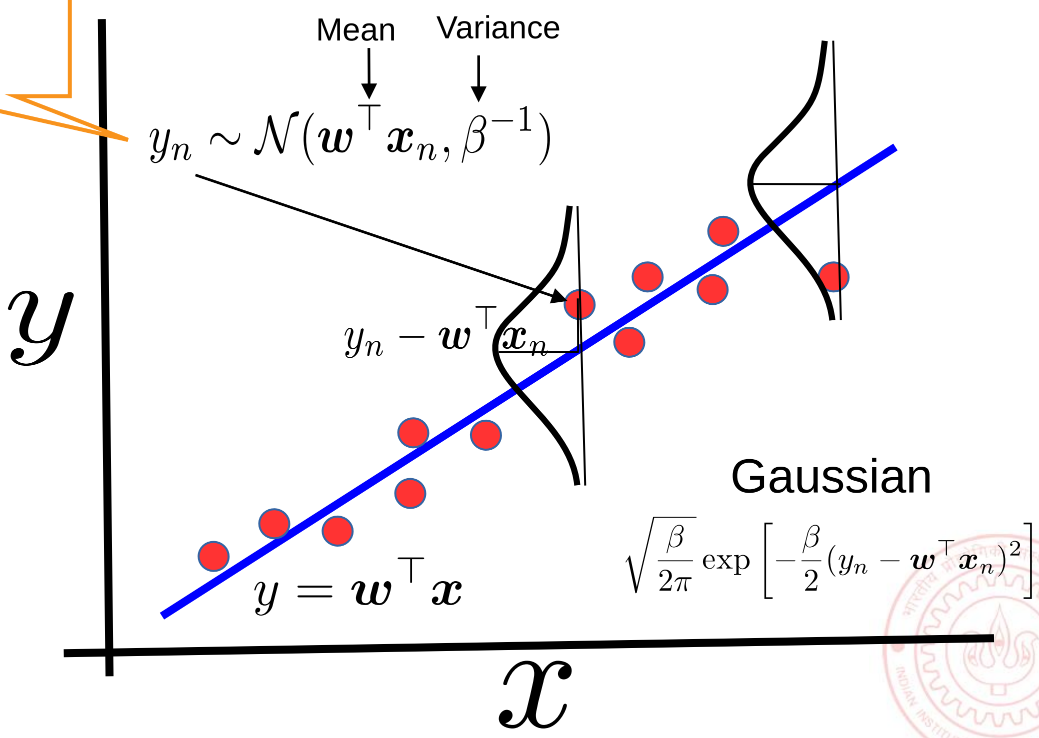
- Assume training data $\{\boldsymbol{x}_n, y_n\}_{n=1}^{N}$, with features $\boldsymbol{x}_n \in \mathbb{R}^D$ and responses $y_n \in \mathbb{R}$
- Assume each y_n generated by a noisy linear model with wts $\mathbf{w} = [w_1, \dots, w_D]$

Each weight assumed real-valued

$$y_n = \boldsymbol{w}^\top \boldsymbol{x}_n + \epsilon_n$$

$$\text{where } \epsilon_n \sim \mathcal{N}(0, \beta^{-1})$$

Output $\boldsymbol{y_n}$ assumed generated from a Gaussian with mean $\boldsymbol{w}^\top \boldsymbol{x}_n$



- Precision ($\boldsymbol{\beta}$) variance of the Gaussian noise tells is how noisy the outputs are (i.e., how far from the mean they are)
- Other noise models also possible (e.g., Laplace distribution for noise)

Probabilistic Linear Regression

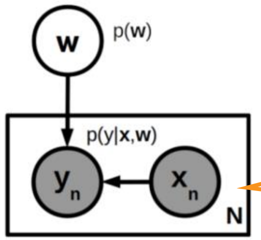


Plate diagram. Hyperparams (λ, β) are fixed and not shown for brevity

- The linear model with Gaussian noise corresponds to a Gaussian likelihood

$$p(y_n|\boldsymbol{x}_n, \boldsymbol{w}, \beta) = \mathcal{N}(y_n|\boldsymbol{w}^\top \boldsymbol{x}_n, \beta^{-1})$$

NLL corresponds to squared loss prop. to $(y_n - \boldsymbol{w}^\top \boldsymbol{x}_n)^2$

- Assuming responses to be i.i.d. given features and weights

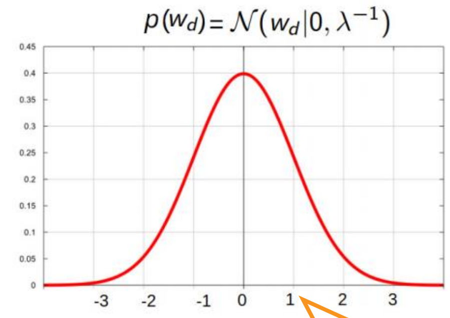
$N \times D$ feature matrix

$$p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(y_n|\boldsymbol{w}^\top \boldsymbol{x}_n, \beta^{-1}) = \mathcal{N}(\boldsymbol{y}|\mathbf{X}\boldsymbol{w}, \beta^{-1}\mathbf{I}_N)$$

$N \times 1$ response vector

- The above is equivalent to the following

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w} + \boldsymbol{\epsilon} \quad \text{where } \boldsymbol{\epsilon} \sim \mathcal{N}(0, \beta^{-1}\mathbf{I_N})$$



- Assume the following Gaussian prior on $\mathbf{w}$,

Neg. log-prior corresponds to ℓ_2 regularizer with λ being the reg. constant

$$p(\boldsymbol{w}) = \prod_{d=1}^{D} p(w_d) = \prod_{d=1}^{D} \mathcal{N}(w_d|0, \lambda^{-1}) = \mathcal{N}(\boldsymbol{w}|\mathbf{0}, \lambda^{-1}\mathbf{I}_D) = \left(\frac{\lambda}{2\pi}\right)^{\frac{D}{2}} \exp\left[-\frac{\lambda}{2}\boldsymbol{w}^\top \boldsymbol{w}\right]$$

Can even use different λ's for different w_d's. Useful in sparse modeling (later)

The precision λ of the Gaussian prior controls how aggressively the prior pushes the elements towards mean (0)

- Then $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w} + \boldsymbol{\epsilon}$ is simply a linear Gaussian model
  - Can use all the rules of linear Gaussian models to perform inference/predictions ☺

The Posterior

Will only look at fully Bayesian inference. For MLE/MAP, refer to CS771 slides or book

- The posterior over $\boldsymbol{w}$ (for now, assume hyperparams $\boldsymbol{\beta}$ and $\boldsymbol{\lambda}$ to be known)

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

Marginal likelihood for this regression model. Note that it is conditioned on $\mathbf{X}$ too which is assumed given and not being modeled

Must be a Gaussian due to conjugacy

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

- Using the "completing the squares" trick (or linear Gaussian model results)

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

Note that $\boldsymbol{\lambda}$ and $\boldsymbol{\beta}$ can be learned under the probabilistic set-up (though assumed fixed as of now)

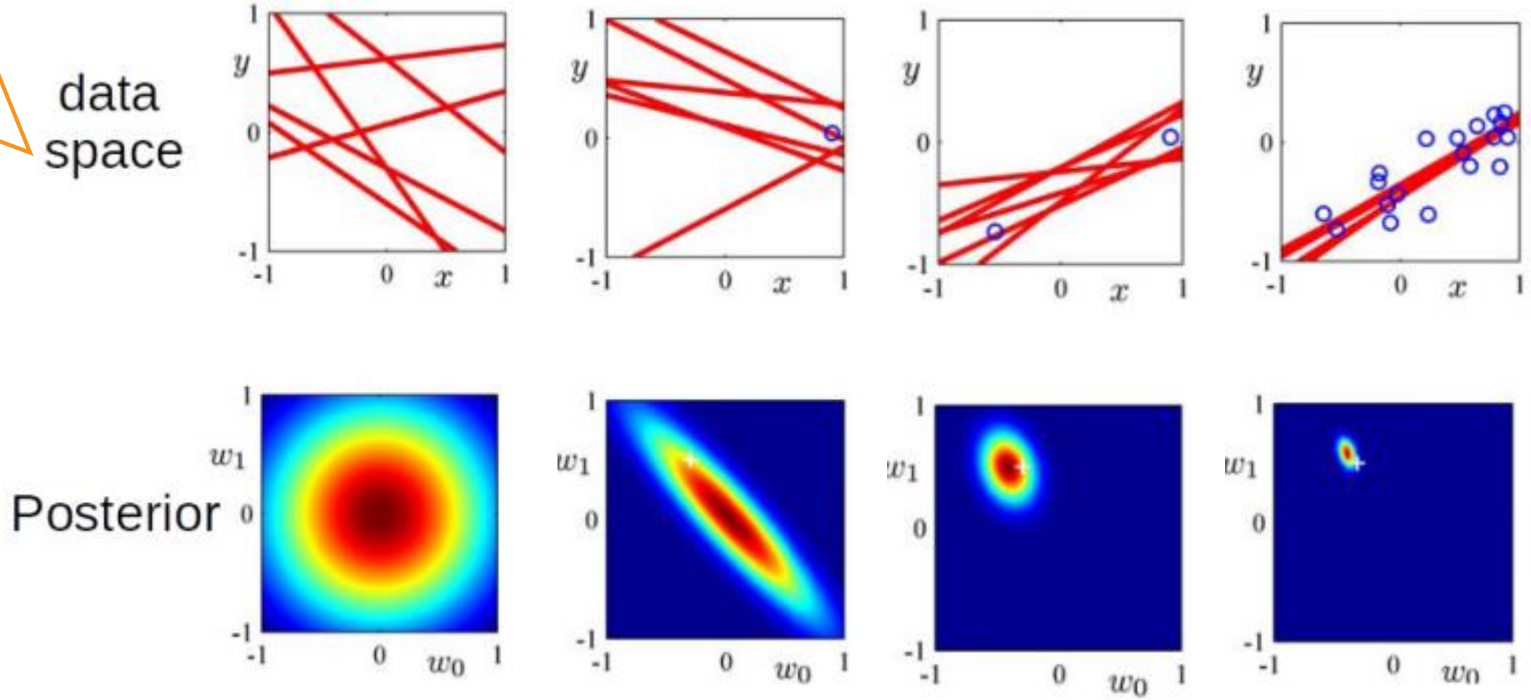
$$\text{where} \quad \boldsymbol{\Sigma}_N \quad = \quad (\beta\sum_{n=1}^{N}\boldsymbol{x}_n\boldsymbol{x}_n^\top + \lambda\mathbf{I}_D)^{-1} = (\beta\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_D)^{-1} \quad \text{(posterior's covariance matrix)}$$

$$\boldsymbol{\mu}_N \quad = \quad \boldsymbol{\Sigma}_N\left[\beta\sum_{n=1}^{N}y_n\boldsymbol{x}_n\right] = \boldsymbol{\Sigma}_N\left[\beta\mathbf{X}^\top\boldsymbol{y}\right] = (\mathbf{X}^\top\mathbf{X} + \frac{\lambda}{\beta}\mathbf{I}_D)^{-1}\mathbf{X}^\top\boldsymbol{y} \quad \text{(posterior's mean)}$$

The Posterior: A Visualization

- Assume a lin. reg. problem with true $\boldsymbol{w} = [w_0, w_1], w_0 = -0.3, w_1 = 0.5$
- Assume data generated by a linear regression model $y = w_0 + w_1 x + \text{"noise"}$
  - Note: It's actually 1-D regression (w_0 is just a bias term), or 2-D reg. with feature $[1, x]$
- Figures below show the "data space" and posterior of $\boldsymbol{w}$ for different number of observations (note: with no observations, the posterior = prior)

Each red line represents the "data" generated for a randomly drawn $\boldsymbol{w}$ from the current posterior



Posterior Predictive Distribution

- To get the prediction $\boldsymbol{y}_*$ for a new input $\boldsymbol{x}_*$, we can compute its PPD

$$p(y_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}, \beta, \lambda) = \int \underbrace{p(y_*|\mathbf{x}_*, \mathbf{w}, \beta)}_{\mathcal{N}(y_*|\boldsymbol{w}^\top \boldsymbol{x}_*, \beta^{-1})} \underbrace{p(\mathbf{w}|\mathbf{X}, \mathbf{y}, \beta, \lambda)}_{\mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}_N, \boldsymbol{\Sigma}_N)} d\mathbf{w}$$

Only $\boldsymbol{w}$ is unknown with a posterior distribution so only $\boldsymbol{w}$ has to be integrated out

- The above is the marginalization of $\boldsymbol{w}$ from $\mathcal{N}(y_*|\boldsymbol{w}^\top \boldsymbol{x}_*, \beta^{-1})$. Using Gaussian results

$$\boxed{p(y_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}, \beta, \lambda) = \mathcal{N}(\boldsymbol{\mu}_N^\top \mathbf{x}_*, \beta^{-1} + \mathbf{x}_*^\top \boldsymbol{\Sigma}_N \mathbf{x}_*)}$$

Can also derive it by writing $y_* = \boldsymbol{w}^\top \boldsymbol{x}_* + \epsilon$ where $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{\mu}_N, \boldsymbol{\Sigma}_N)$ and $\epsilon \sim \mathcal{N}(0, \beta^{-1})$

- So we have a predictive mean $\boldsymbol{\mu}_N^\top \boldsymbol{x}_*$ as well as an input-specific predictive variance
- In contrast, MLE and MAP make "plug-in" predictions (using the point estimate of $\boldsymbol{w}$)

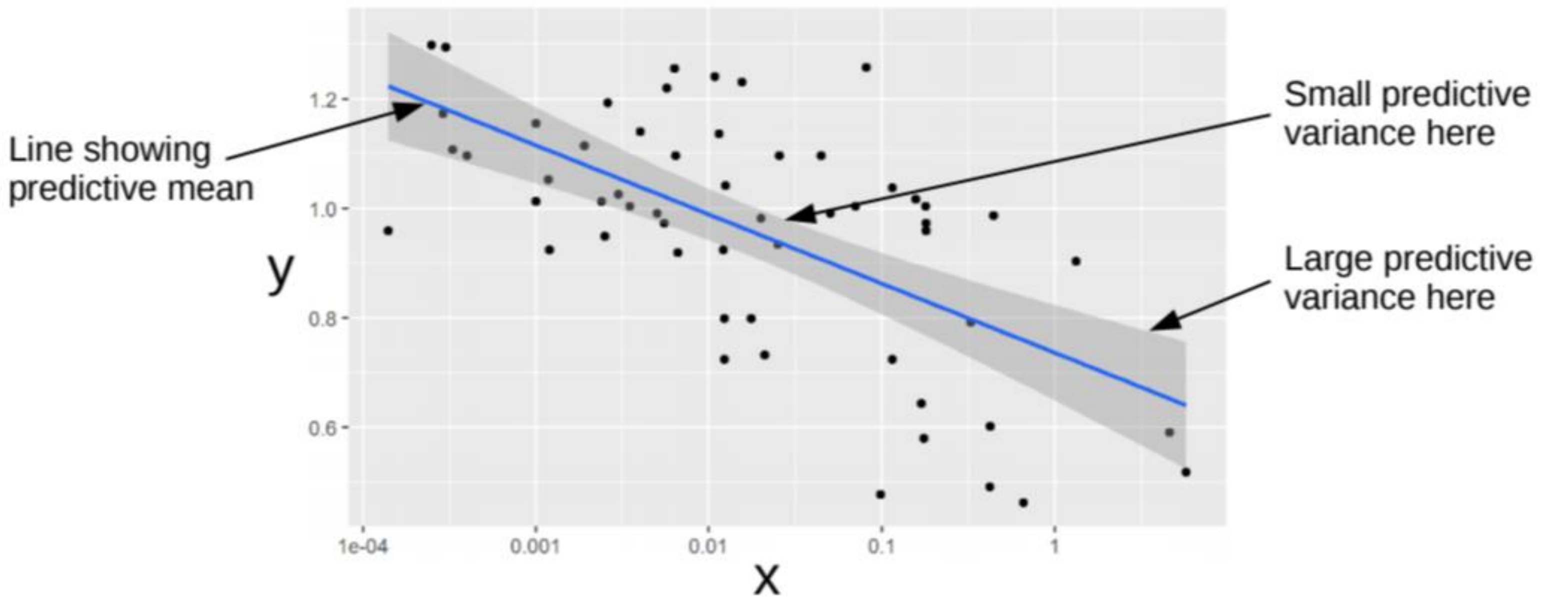
$$p(y_*|\mathbf{x}_*, \mathbf{w}_{MLE}) = \mathcal{N}(\mathbf{w}_{MLE}^\top \mathbf{x}_*, \beta^{-1}) \quad \text{- MLE prediction}$$
$$p(y_*|\mathbf{x}_*, \mathbf{w}_{MAP}) = \mathcal{N}(\mathbf{w}_{MAP}^\top \mathbf{x}_*, \beta^{-1}) \quad \text{- MAP prediction}$$

Since PPD also takes into account the uncertainty in $\boldsymbol{w}$, the predictive variance is larger

- Unlike MLE/MAP, variance of y_* also depends on the input $\boldsymbol{x}_*$ (this, as we will see later, will be very useful in sequential decision-making problems such as active learning)

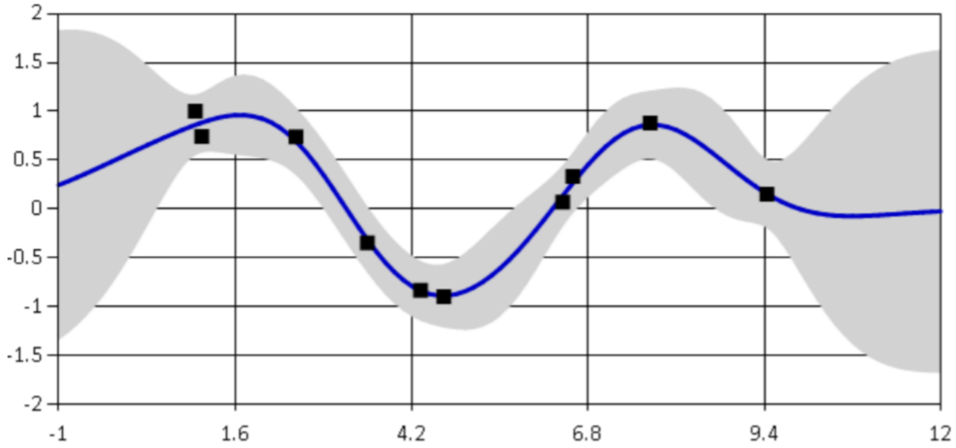
Posterior Predictive Distribution: An Illustration

- Black dots are training examples



- Width of the shaded region at any x denotes the predictive uncertainty at that x (+/- one std-dev)
- Regions with more training examples have smaller predictive variance

Nonlinear Regression



- Can extend the linear regression model to handle nonlinear regression problems
- One way is to replace the feature vectors $\boldsymbol{x}$ by a nonlinear mapping $\boldsymbol{\phi(x)}$

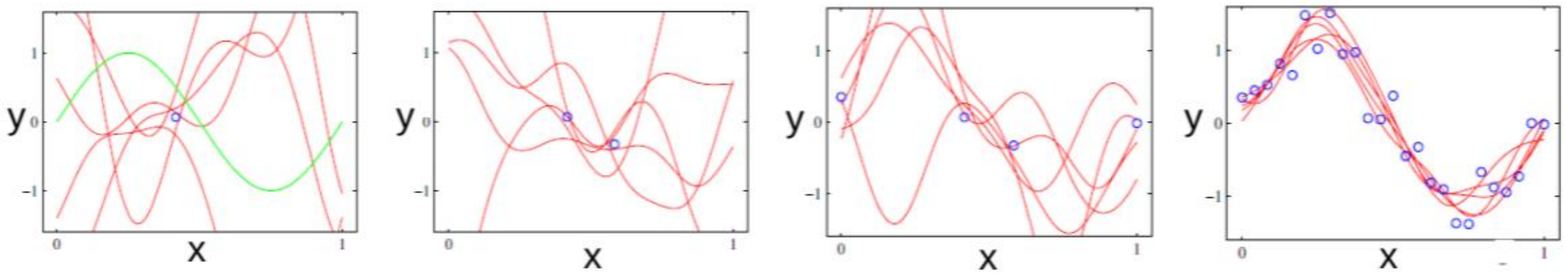
$$p(y|\mathbf{x}, \mathbf{w}) = \mathcal{N}(\mathbf{w}^\top \phi(\mathbf{x}), \beta^{-1})$$

Can be pre-defined or extracted by a pretrained deep neural net

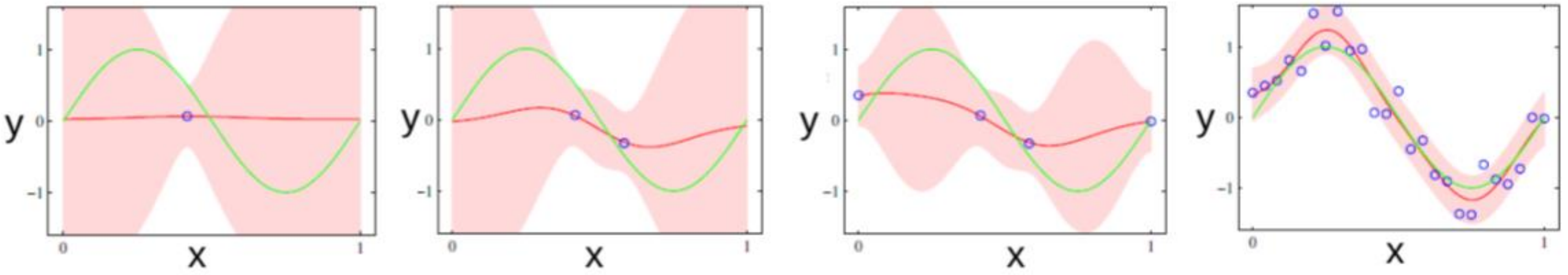
- Alternatively, a kernel function can be used to implicitly define the nonlinear mapping
- More on nonlinear regression when we discuss Gaussian Processes

More on Visualization of Uncertainty

- Figures below: Green curve is the true function and blue circles are observations



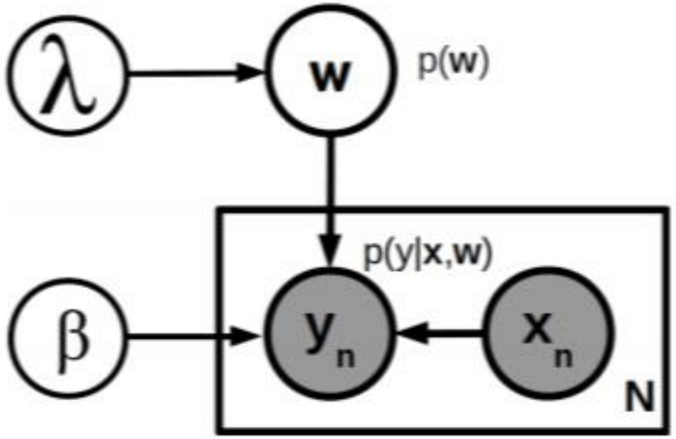
- Posterior of the nonlinear regression model: Some curves drawn from the posterior



- PPD: Red curve is predictive mean, shaded region denotes predictive uncertainty

Hyperparameters

- The probabilistic linear reg. model we saw had two hyperparams $(\boldsymbol{\beta}, \boldsymbol{\lambda})$
  - Thus total three unknowns $(\boldsymbol{w}, \boldsymbol{\beta}, \boldsymbol{\lambda})$



Need posterior over all the 3 unknowns

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

PPD would require integrating out all 3 unknowns

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

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

- Posterior and PPD computation is intractable. Several ways to address this
  - MLE-II for $(\boldsymbol{\beta}, \boldsymbol{\lambda})$: $\hat{\beta}, \hat{\lambda} = \arg\max_{\beta, \lambda} p(\boldsymbol{y}|\mathbf{X}, \beta, \lambda)$. Use them to infer the posterior of $\boldsymbol{\theta}$ and PPD
  - Use alternating estimation like EM (e.g., E step computes $\boldsymbol{\theta}$, M step computes $(\boldsymbol{\beta}, \boldsymbol{\lambda})$)
  - Use MCMC or Variational Inference to approximate the above posterior and PPD

MLE-II

For any model where hyperparams are estimated by MLE-II, the posterior and PPD is approximated in a similar fashion

- For the probabilistic linear regression model, the overall posterior over unknowns

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

- With MLE-II approx of $(\boldsymbol{\beta}, \boldsymbol{\lambda})$, $p(\beta, \lambda|\mathbf{X}, \mathbf{y}) \approx \delta(\hat{\beta}, \hat{\lambda})$, a point mass at $\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\lambda}}$

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

Same as the posterior of $\boldsymbol{w}$ with the hyperparameters fixed

- Likewise, the PPD will be approximated as follows

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

Same form for the PPD as in the case of fixed hyperparams

Only need to integrate over $\boldsymbol{w}$, since other two are fixed at their MLE-II solutions

Road So Far and What Lies Ahead

- Seen Bayesian inference for several models with a single unknown parameter (and another simple case where we had two unknown parameters - Gaussian with unknown mean and precision)
- Focused on the cases where the likelihood and prior are conjugate
- Both posterior as well as posterior predictive are computable easily in such cases
- Saw various nice properties of exp. family distributions and parameter estimation for such distributions. Also saw estimation in a conditional model (linear regression)
- Things become more challenging/interesting for more complex models, e.g.,
  - Multiple unknown parameters (e.g., hyperparams, latent variables, hierarchical models etc)
  - Likelihood and prior are not conjugate. Approximate inference methods (MCMC, VI, etc)
- Basic ideas we saw will turn out to be useful in more complex models as well
  - Conditionally-conjugate or locally-conjugate models (conjugacy exists in sub-parts of the model)
  - Some approximate inference methods, e.g., Gibbs sampling, VI, also rely on conjugacy

Coming Up Next

- Hyperparameter estimation for Bayesian linear regression
- Sparse modeling