Probabilistic Supervised Learning: Linear Regression

CS772A: Probabilistic Machine Learning

Piyush Rai

Plan Today

- Quick overview of parameter estimation and predictive distributions for
  - Multinoulli observation model
  - Gaussian (univariate) observation model
- Probabilistic Supervised Learning
  - (Probabilistic) Linear Regression

Multinoulli Observation Model

The Posterior Distribution

MLE/MAP left as an exercise

- Assume N discrete obs $\boldsymbol{y} = \{y_1, y_2, \ldots, y_N\}$ with each $y_n \in \{1,2,\ldots,K\}$, e.g.,
  - y_n represents the outcome of a dice roll with K faces
  - y_n represents the class label of the n^{th} example in a classification problem (total K classes)
  - y_n represents the identity of the n^{th} word in a sequence of words
- Assume likelihood to be multinoulli with unknown params $\boldsymbol{\pi} = [\pi_1, \pi_2, \ldots, \pi_K]$

These sum to 1

$$p(y_n|\pi) = \text{multinoulli}(y_n|\pi) = \prod_{k=1}^{K} \pi_k^{\mathbb{I}[y_n=k]}$$

Generalization of Bernoulli to $K > 2$ discrete outcomes

- $\boldsymbol{\pi}$ is a vector of probabilities ("probability vector"), e.g.,
  - Biases of the K sides of the dice
  - Prior class probabilities in multi-class classification $(p(y_n = k) = \pi_k)$
  - Probabilities of observing each word of the K words in a vocabulary

Called the concentration parameter of the Dirichlet (assumed known for now)

Large values of α will give a Dirichlet peaked around its mean (next slides illustrates this)

Each $\alpha_k \geq 0$

- Assume a conjugate prior (Dirichlet) on $\boldsymbol{\pi}$ with hyperparams $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_K]$

$$p(\boldsymbol{\pi}|\boldsymbol{\alpha}) = \text{Dirichlet}(\boldsymbol{\pi}|\alpha_1, \ldots, \alpha_K) = \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \pi_k^{\alpha_k - 1} = \frac{1}{B(\boldsymbol{\alpha})} \prod_{k=1}^{K} \pi_k^{\alpha_k - 1}$$

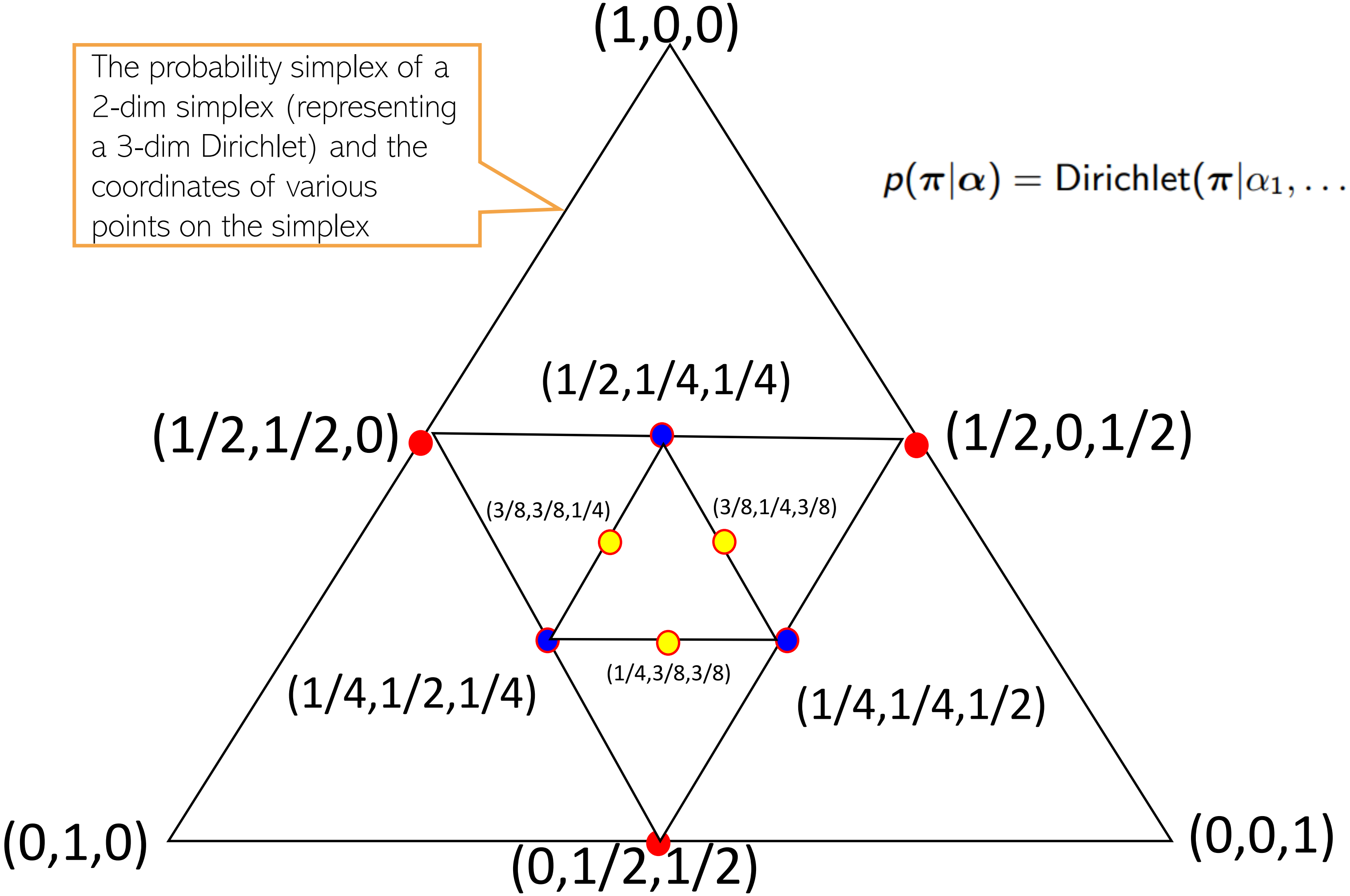
Generalization of Beta to K-dimensional probability vectors

Brief Detour: Dirichlet Distribution

Basically, probability vectors

- An important distribution. Models non-neg. vectors $\boldsymbol{\pi}$ that also sum to one
- A random draw from K-dim Dirich. will be a point under $(K$-1$)$-dim probability simplex

The probability simplex of a 2-dim simplex (representing a 3-dim Dirichlet) and the coordinates of various points on the simplex



$$p(\boldsymbol{\pi}|\boldsymbol{\alpha}) = \text{Dirichlet}(\boldsymbol{\pi}|\alpha_1, \ldots, \alpha_K) = \frac{\Gamma(\sum_{k=1}^K \alpha_k)}{\prod_{k=1}^K \Gamma(\alpha_k)} \prod_{k=1}^K \pi_k^{\alpha_k - 1} = \frac{1}{B(\boldsymbol{\alpha})} \prod_{k=1}^K \pi_k^{\alpha_k - 1}$$

$$\text{Mean} = \left[\frac{\alpha_1}{\sum_{k=1}^K \alpha_k}, \ldots, \frac{\alpha_K}{\sum_{k=1}^K \alpha_k}\right]$$

$$\text{Mode} = \left[\frac{\alpha_1 - 1}{\sum_{k=1}^K \alpha_k - K}, \ldots, \frac{\alpha_K - 1}{\sum_{k=1}^K \alpha_k - K}\right] (\alpha_k > 1)$$

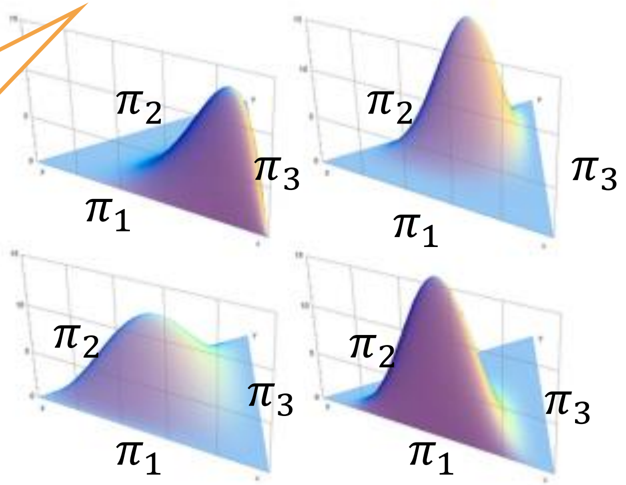
$$\text{var}(\pi_k) = \frac{\alpha_k(\alpha_0 - \alpha_k)}{\alpha_0^2(\alpha_0 + 1)} \qquad \alpha_0 = \sum_{k=1}^K \alpha_k$$

Brief Detour: Dirichlet Distribution

- A visualization of Dirichlet distribution for different values of concentration param

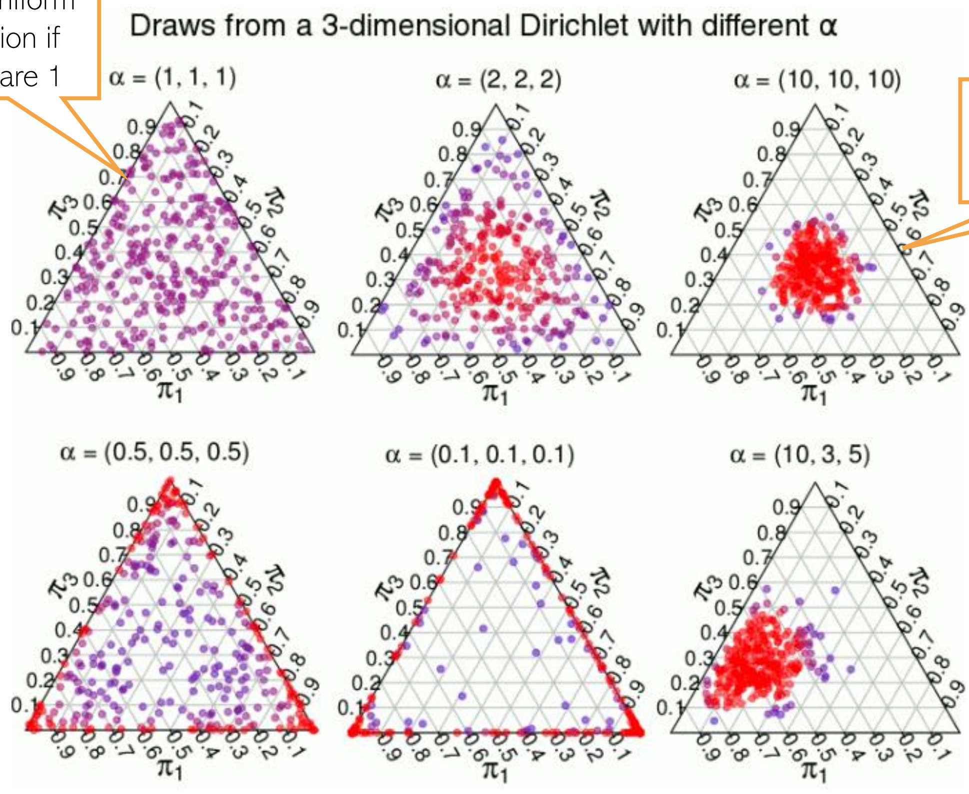
Visualizations of PDFs of some 3-dim Dirichlet distributions (each generated using a different conc. Param vector $\boldsymbol{\alpha}$)

$\boldsymbol{\alpha}$ controls the shape of the Dirichlet (just like Beta distribution's hyperparameters)



Like a uniform distribution if all α_k's are 1

All α_k's large results in peak around the center of the simplex



- Interesting fact: Can generate a K-dim Dirichlet random variable by independently generating K gamma random variables and normalizing them to sum to 1

The Posterior Distribution

Likelihood

Prior

- Posterior $p(\boldsymbol{\pi}|\boldsymbol{y})$ is easy to compute due to conjugacy b/w multinoulli and Dir.

$$p(\boldsymbol{\pi}|\boldsymbol{y},\boldsymbol{\alpha}) = \frac{p(\boldsymbol{\pi},\boldsymbol{y}|\boldsymbol{\alpha})}{p(\boldsymbol{y}|\boldsymbol{\alpha})} = \frac{p(\boldsymbol{\pi}|\boldsymbol{\alpha})p(\boldsymbol{y}|\boldsymbol{\pi},\boldsymbol{\alpha})}{p(\boldsymbol{y}|\boldsymbol{\alpha})} = \frac{p(\boldsymbol{\pi}|\boldsymbol{\alpha})p(\boldsymbol{y}|\boldsymbol{\pi})}{p(\boldsymbol{y}|\boldsymbol{\alpha})}$$

Don't need to compute for this case because of conjugacy

Marg-lik = $\int p(\boldsymbol{\pi}|\boldsymbol{\alpha})p(\boldsymbol{y}|\boldsymbol{\pi})d\boldsymbol{\pi}$

- Assuming y_n's are i.i.d. given $\boldsymbol{\pi}$, $p(\boldsymbol{y}|\boldsymbol{\pi}) = \prod_{n=1}^{N} p(y_n|\boldsymbol{\pi})$, and therefore

$$p(\boldsymbol{\pi}|\boldsymbol{y},\boldsymbol{\alpha}) \propto \prod_{k=1}^{K} \pi_k^{\alpha_k - 1} \times \prod_{n=1}^{N} \prod_{k=1}^{K} \pi_k^{\mathbb{I}[y_n=k]} = \prod_{k=1}^{K} \pi_k^{\alpha_k + \sum_{n=1}^{N} \mathbb{I}[y_n=k] - 1}$$

- Even without computing marg-lik, $p(\boldsymbol{y}|\boldsymbol{\alpha})$, we can see that the posterior is Dirichlet
- Denoting $N_k = \sum_{n=1}^{N} \mathbb{I}[y_n = k]$, number of observations with with value k

$$p(\boldsymbol{\pi}|\boldsymbol{y},\boldsymbol{\alpha}) = \text{Dirichlet}(\boldsymbol{\pi}|\alpha_1 + N_1, \alpha_2 + N_2, \ldots, \alpha_K + N_K)$$

Similar to number of heads and tails for the coin bias estimation problem

- Note: $N_1, , N_2 \ldots, N_K$ are the sufficient statistics for this estimation problem
  - We only need the suff-stats to estimate the parameters and values of individual observations aren't needed (another property from exponential family of distributions – more on this later)

The Predictive Distribution

- Finally, let's also look at the posterior predictive distribution for this model
- PPD is the prob distr of a new $y_* \in \{1,2,\ldots,K\}$, given training data $\boldsymbol{y} = \{y_1, y_2, \ldots, y_N\}$

Will be a multinoulli. Just need to estimate the probabilities of each of the K outcomes

$$p(y_*|\boldsymbol{y},\boldsymbol{\alpha}) = \int p(y_*|\boldsymbol{\pi})\boldsymbol{p}(\boldsymbol{\pi}|\boldsymbol{y},\boldsymbol{\alpha})\boldsymbol{d\pi}$$

- $p(y_*|\boldsymbol{\pi}) = \text{multinoulli}(y_*|\boldsymbol{\pi})$, $p(\boldsymbol{\pi}|\boldsymbol{y},\boldsymbol{\alpha}) = \text{Dirichlet}(\boldsymbol{\pi}|\alpha_1+N_1,\alpha_2+N_2,\ldots,\alpha_K+N_K)$
- Can compute the posterior predictive probability for each of the K possible outcomes

$$p(y_* = k|\boldsymbol{y},\boldsymbol{\alpha}) = \int p(y_* = k|\boldsymbol{\pi})\boldsymbol{p}(\boldsymbol{\pi}|\boldsymbol{y},\boldsymbol{\alpha})\boldsymbol{d\pi}$$

$$= \int \pi_k \times \text{Dirichlet}(\boldsymbol{\pi}|\alpha_1+N_1,\alpha_2+N_2,\ldots,\alpha_K+N_K)d\pi$$

$$= \frac{\alpha_k + N_k}{\sum_{k=1}^{K}\alpha_k + N}$$ (Expectation of π_k w.r.t the Dirichlet posterior)

- Thus PPD is multinoulli with probability vector $\left\{\frac{\alpha_k+N_k}{\sum_{k=1}^{K}\alpha_k+N}\right\}_{k=1}^{K}$

Note how these probabilities have been "smoothened" due to the use of the prior + the averaging over the posterior

A similar effect was achieved in the Beta-Bernoulli model, too

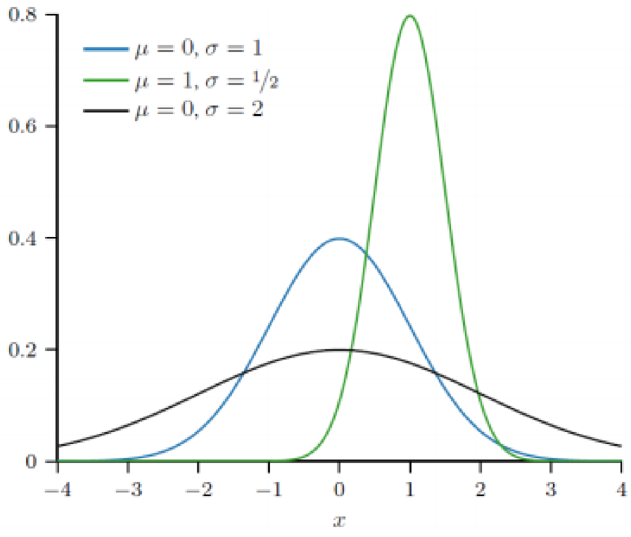
- Plug-in predictive will also be multinoulli but with prob vector given by the point estimate of $\boldsymbol{\pi}$

Gaussian Observation Model

Gaussian Distribution (Univariate)

- Distribution over real-valued scalar random variables $\boldsymbol{X} \in \mathbb{R}$, e.g., height of students in a class
- Defined by a scalar mean $\boldsymbol{\mu}$ and a scalar variance $\boldsymbol{\sigma}^2$

$$\mathcal{N}(X = x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$$



- Mean: $\mathbb{E}[X] = \mu$
- Variance: $\text{var}[X] = \sigma^2$
- Inverse of variance is called precision: $\beta = \frac{1}{\sigma^2}$.

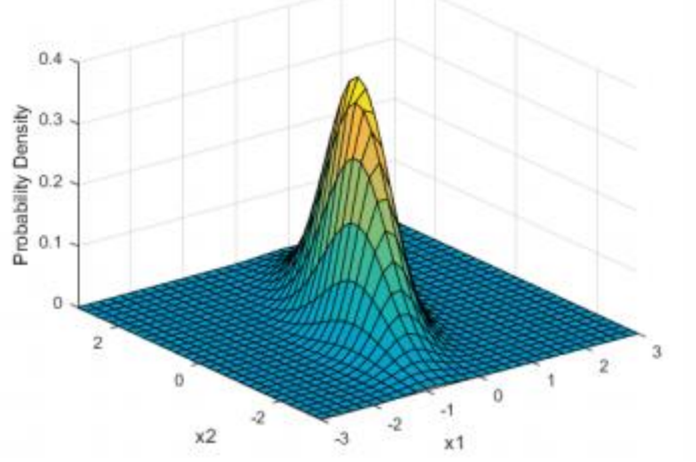
Gaussian PDF in terms of precision

$$\mathcal{N}(X = x|\mu, \beta) = \sqrt{\frac{\beta}{2\pi}} \exp\left[-\frac{\beta}{2}(x-\mu)^2\right]$$

Gaussian Distribution (Multivariate)

- Distribution over real-valued vector random variables $\boldsymbol{X} \in \mathbb{R}^D$
- Defined by a mean vector $\boldsymbol{\mu} \in \mathbb{R}^D$ and a covariance matrix $\boldsymbol{\Sigma}$

A two-dimensional Gaussian



$$\mathcal{N}(\boldsymbol{X} = \boldsymbol{x} | \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^D |\boldsymbol{\Sigma}|}} \exp[-(\boldsymbol{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu})]$$

- Note: The cov. matrix $\boldsymbol{\Sigma}$ must be symmetric and PSD
  - All eigenvalues are positive
  - $\boldsymbol{z}^\top \boldsymbol{\Sigma} \boldsymbol{z} \geq \boldsymbol{0}$ for any real vector $\boldsymbol{z}$
- The covariance matrix also controls the shape of the Gaussian

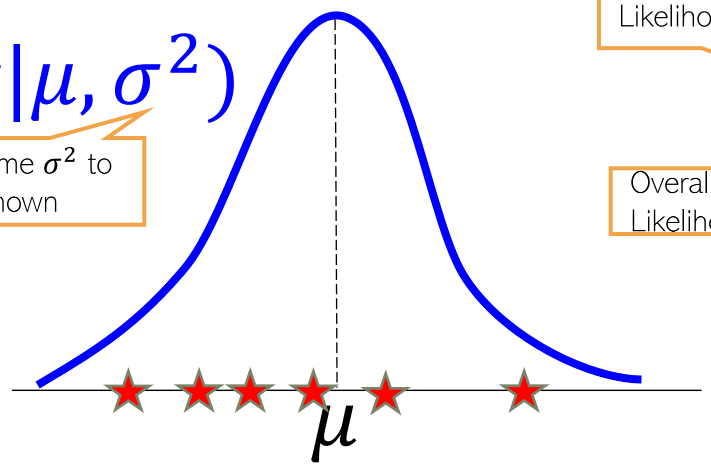
Posterior Distribution for Gaussian's Mean

Its MLE/MAP estimation left as an exercise

- Given: N i.i.d. scalar observations $\boldsymbol{y} = \{y_1, y_2, \ldots, y_N\}$ assumed drawn from $\mathcal{N}(y|\mu, \sigma^2)$

$\mathcal{N}(y|\mu, \sigma^2)$

Assume σ^2 to be known



Likelihood

$$p(y_n|\mu, \sigma^2) = \mathcal{N}(y_n|\mu, \sigma^2) \propto \exp - \left[\frac{(y_n - \mu)^2}{2\sigma^2}\right]$$

Overall Likelihood

$$p(\boldsymbol{y}|\mu, \sigma^2) = \prod_{n=1}^{N} p(y_n|\mu, \sigma^2)$$

- Note: Easy to see that each y_n drawn from $\mathcal{N}(y|\mu, \sigma^2)$ is equivalent to the following

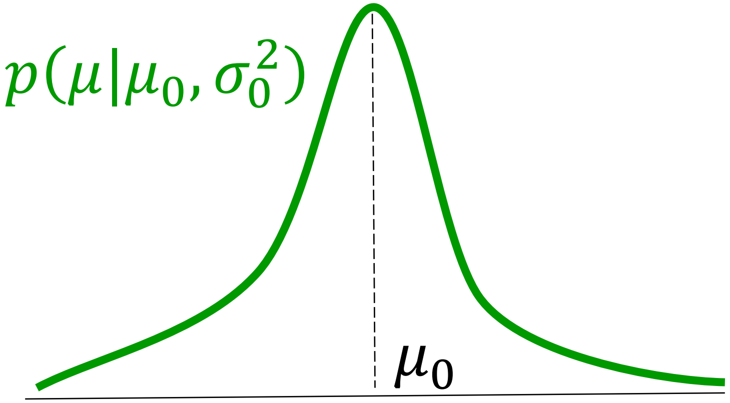
Thus y_n is like a noisy version of μ with zero mean Gaussian noise added to it

$$y_n = \mu + \epsilon_n \qquad \text{where } \epsilon_n \sim \mathcal{N}(0, \sigma^2)$$

- Let's estimate mean μ given $\boldsymbol{y}$ using fully Bayesian inference (not point estimation)

A prior distribution for the mean

- To computer posterior, need a prior over $\boldsymbol{\mu}$
- Let's choose a Gaussian prior



$$p(\mu|\mu_0, \sigma_0^2) = \mathcal{N}(\mu|\mu_0, \sigma_0^2)$$
$$\propto \exp\left[-\frac{(\mu - \mu_0)^2}{2\sigma_0^2}\right]$$

- The prior basically says that *a priori* we believe $\boldsymbol{\mu}$ is close to $\boldsymbol{\mu_0}$
- The prior's variance $\boldsymbol{\sigma_0^2}$ denotes how certain we are about our belief
- We will assume that the prior's hyperparameters $(\boldsymbol{\mu_0}, \boldsymbol{\sigma_0^2})$ are known
- Since $\boldsymbol{\sigma^2}$ in the likelihood $\mathcal{N}(y|\mu, \sigma^2)$ is known, Gaussian prior $\mathcal{N}(\mu|\mu_0, \sigma_0^2)$ on $\boldsymbol{\mu}$ is also conjugate to the likelihood (thus posterior of $\boldsymbol{\mu}$ will also be Gaussian)

The posterior distribution for the mean

- The posterior distribution for the unknown mean parameter μ

On conditioning side, skipping all fixed params and hyperparams from the notation

$$p(\mu|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\mu)p(\mu)}{p(\boldsymbol{y})} \propto \prod_{n=1}^{N} \exp\left[-\frac{(y_n-\mu)^2}{2\sigma^2}\right] \exp\left[-\frac{(\mu-\mu_0)^2}{2\sigma_0^2}\right]$$

- Easy to see that the above will be prop. to exp of a quadratic function of μ. Simplifying:

$$p(\mu|\boldsymbol{y}) \propto \exp\left[-\frac{(\mu-\mu_N)^2}{2\sigma_N^2}\right]$$

Gaussian posterior (not a surprise since the chosen prior was conjugate to the likelihood)

$$\frac{1}{\sigma_N^2} = \frac{1}{\sigma_0^2} + \frac{N}{\sigma^2}$$

Gaussian posterior's precision is the sum of the prior's precision and sum of the noise precisions of all the observations

$$\mu_N = \frac{\sigma^2}{N\sigma_0^2+\sigma^2}\mu_0 + \frac{N\sigma_0^2}{N\sigma_0^2+\sigma^2}\bar{y} \quad \left(\text{where } \bar{y} = \frac{\sum_{n=1}^{N} y_n}{N}\right)$$

Gaussian posterior's mean is a convex combination of prior's mean and the MLE solution

Contribution from the prior

Contribution from the data

Also the MLE solution for μ

- What happens to the posterior as N (number of observations) grows very large?
  - Data (likelihood part) overwhelms the prior
  - Posterior's variance σ_N^2 will approximately be σ^2/N (and goes to 0 as $N \to \infty$)
  - The posterior's mean μ_N approaches $\bar{y}$ (which is also the MLE solution)

Meaning, we become very-very certain about the estimate of μ

The Predictive Distribution

- If given a point estimate $\hat{\mu}$, the plug-in predictive distribution for a test y_* would be

The best point estimate

This is an approximation of the true PPD $p(y_*|\boldsymbol{y})$

$$p(y_*|\hat{\mu},\sigma^2) = \mathcal{N}(y_*|\hat{\mu},\sigma^2)$$

- On the other hand, the posterior predictive distribution of y_* would be

$$p(y_*|\boldsymbol{y}) = \int p(y_*|\mu,\sigma^2)p(\mu|\boldsymbol{y})d\mu$$

$$= \int \mathcal{N}(y_*|\mu,\sigma^2)\mathcal{N}(\mu|\mu_N,\sigma_N^2)d\mu$$

$$= \mathcal{N}(y_*|\mu_N,\sigma^2+\sigma_N^2)$$

This "extra" variance σ_N^2 in PPD is due to the averaging over the posterior's uncertainty

If conditional is Gaussian then marginal is also Gaussian

A useful fact: When we have conjugacy, the posterior predictive distribution also has a closed form (will see this result more formally when talking about exponential family distributions)

PRML [Bis 06], 2.115, and also mentioned in prob-stats refresher slides

- For an alternative way to get the above result, note that, for test data

$$y_* = \mu + \epsilon \qquad \mu \sim \mathcal{N}(\mu_N,\sigma_N^2) \qquad \epsilon \sim \mathcal{N}(0,\sigma^2)$$

Using the **posterior** of μ since we are at test stage now

$$\Rightarrow \quad p(y_*|\boldsymbol{y}) = \mathcal{N}(y_*|\mu_N,\sigma^2+\sigma_N^2)$$

Since both μ and ϵ are Gaussian r.v., and are independent, y_* also has a Gaussian posterior predictive, and the respective means and variances of μ and ϵ get added up

Gaussian Observation Model: Some Other Facts

- MLE/MAP for μ, σ^2 (or both) is straightforward in Gaussian observation models.
- Posterior also straightforward in most situations for such models
  - (As we saw) computing posterior of μ is easy (using Gaussian prior) if variance σ^2 is known
  - Likewise, computing posterior of σ^2 is easy (using gamma prior on σ^2) if mean μ is known
- If μ, σ^2 both are unknown, posterior computation requires computing $p(\mu, \sigma^2|\boldsymbol{y})$
  - Computing joint posterior $p(\mu, \sigma^2|\boldsymbol{y})$ exactly requires a jointly conjuage prior $p(\mu, \sigma^2)$
  - "Gaussian-gamma" ("Normal-gamma") is such a conjugate prior – a product of normal and gamma
  - Note: Computing joint posteriors exactly is possible only in rare cases such this one
- If each observation $\boldsymbol{y}_n \in \mathbb{R}^D$, can assume a likelihood/observation model $\mathcal{N}(\boldsymbol{y}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$
  - Need to estimate a vector-valued mean $\boldsymbol{\mu} \in \mathbb{R}^D$. Can use a multivariate Gaussian prior
  - Need to estimate a $D \times D$ positive definite covariance matrix $\boldsymbol{\Sigma}$. Can use a Wishart prior
  - If $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ both are unknown, can use Normal-Wishart as a conjugate prior

Linear Gaussian Model (LGM)

- LGM defines a noisy lin. transform of a Gaussian r.v. $\boldsymbol{\theta}$ with $p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1})$

Both $\boldsymbol{\theta}$ and $\boldsymbol{y}$ are vectors (can be of different sizes)

Also assume $\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{\Lambda}, \boldsymbol{L}$ to be known; only $\boldsymbol{\theta}$ is unknown

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{\theta} + \boldsymbol{b} + \boldsymbol{\epsilon}$$

Noise vector - independently and drawn from $\mathcal{N}(\boldsymbol{\epsilon}|\boldsymbol{0}, \boldsymbol{L}^{-1})$

- Easy to see that, conditioned on $\boldsymbol{\theta}$, $\boldsymbol{y}$ too has a Gaussian distribution

Conditional distribution

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

- Assume $p(\boldsymbol{\theta})$ as prior and $p(\boldsymbol{y}|\boldsymbol{\theta})$ as the likelihood, and defining $\boldsymbol{\Sigma} = (\boldsymbol{\Lambda} + \boldsymbol{A}^\top \boldsymbol{L}\boldsymbol{A})^{-1}$

Posterior of $\boldsymbol{\theta}$

$$p(\boldsymbol{\theta}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{y})} = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\Sigma}(\boldsymbol{A}^\top \boldsymbol{L}(\boldsymbol{y} - \boldsymbol{b}) + \boldsymbol{\Lambda}\boldsymbol{\mu}), \boldsymbol{\Sigma})$$

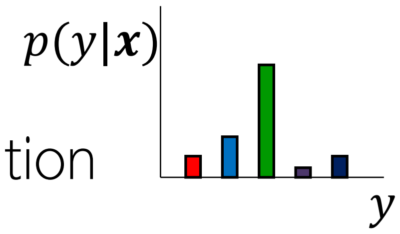
Marginal distribution

$$p(\boldsymbol{y}) = \int p(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta} = \mathcal{N}(\boldsymbol{y}|\boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{b}, \boldsymbol{A}\boldsymbol{\Lambda}^{-1}\boldsymbol{A}^\top + \boldsymbol{L}^{-1})$$

- Many probabilistic ML models are LGMs
- These results are very widely used (PRML Chap. 2 contains a proof)

Probabilistic Supervised Learning

- Goal: To learn the conditional distribution $p(y|\boldsymbol{x})$ of output given input
- The form of the distribution $p(y|\boldsymbol{x})$ depends on output type, e.g.,
  - Real: Model $p(y|\boldsymbol{x})$ using a Gaussian (or some other suitable real-valued distribution)
  - Binary: Model $p(y|\boldsymbol{x})$ using a Bernoulli
  - Categorical/multiclass: Model $p(y|\boldsymbol{x})$ using a multinoulli/categorical distribution
  - Various other types (e.g., count, positive reals, etc) can also be modeled using appropriate distributions (e.g., Poisson for count, gamma for positive reals)



- The distribution $p(y|\boldsymbol{x})$ can be defined directly or indirectly

"Direct" way without modeling the inputs $\boldsymbol{x}_n$

Parameters of this distribution are the outputs of function f

$$p(y|\boldsymbol{x}) = p(y|f(\boldsymbol{x}, \boldsymbol{w}))$$

"Indirect" way by modeling the outputs as well as the inputs

"Indirect" way requires first learning the joint distribution of inputs and outputs

$$p(y|\boldsymbol{x}) = \frac{p(y, \boldsymbol{x})}{p(\boldsymbol{x})}$$

Discriminative vs Generative Sup. Learning

Non-probabilistic supervised learning approaches (e.g., SVM) are usually considered discriminative since $p(x)$ is never modeled

- Direct way of sup. learning is discriminative, indirect way is generative

Discriminative Approach

$$p(y|\boldsymbol{x}) = p(y|f(\boldsymbol{x}, \boldsymbol{w}))$$

f can be any function which uses inputs and weights $\boldsymbol{w}$ to defines parameters of distr. $\boldsymbol{p}$

Some examples

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

$$p(y|\boldsymbol{x}) = \text{Bernoulli}(y|\sigma(\boldsymbol{w}^\top \boldsymbol{x}))$$

Generative Approach

$$p(y|\boldsymbol{x}) = \frac{p(y, \boldsymbol{x})}{p(\boldsymbol{x})}$$

Requires estimating the joint distribution of inputs and outputs to get the conditional $p(y|\boldsymbol{x})$ (unlike the discriminative approach which directly estimates the conditional $p(y|\boldsymbol{x})$ and does not model the distribution of $\boldsymbol{x}$)

- Note: Generative approach can also be used for other settings too, such as unsupervised learning and semi-supervised learning (will see later)

Probabilistic Linear Regression

A discriminative model for regression problems

- Assume training data $\{\boldsymbol{x}_n, y_n\}_{n=1}^N$, with features $\boldsymbol{x}_n \in \mathbb{R}^D$ and responses $y_n \in \mathbb{R}$

Unknown to be estimated

Each weight assumed real-valued

- Assume y_n generated by a noisy linear model with wts $\boldsymbol{w} = [w_1, \ldots, w_D] \in \mathbb{R}^D$

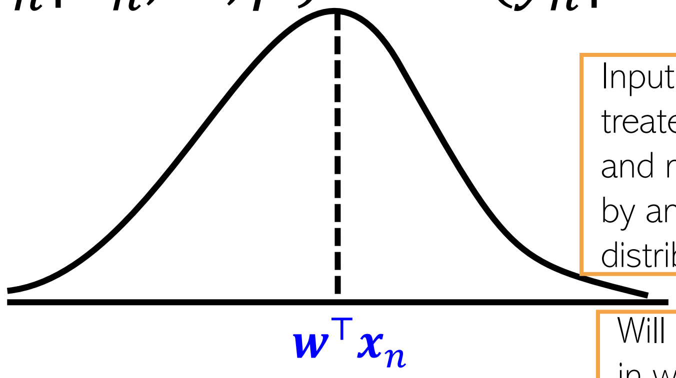
$$y_n = \boldsymbol{w}^\top \boldsymbol{x}_n + \epsilon_n$$

Gaussian noise drawn from $\mathcal{N}(\epsilon_n | 0, \beta^{-1})$

- Notation alert: β is the precision of Gaussian noise (and β^{-1} the variance)

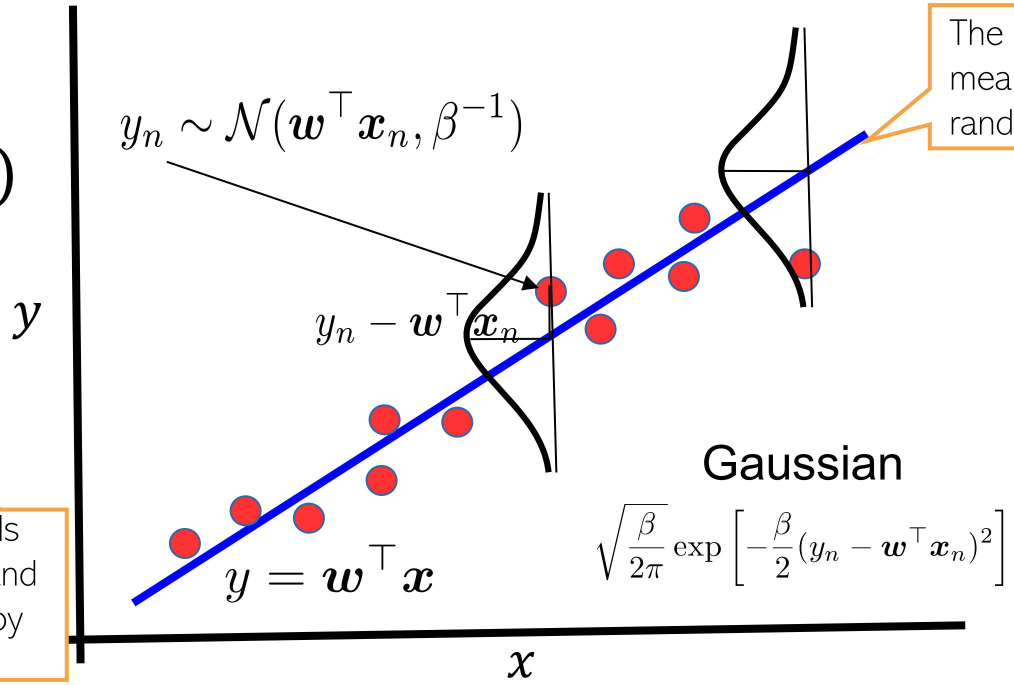
Likelihood model

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



Input x_n being treated as given and not modeled by any probability distribution

Will later study models in which both input and output are modeled by distributions



The line represents the mean $\boldsymbol{w}^\top \boldsymbol{x}_n$ of the output random variable y_n

The zero mean Gaussian noise perturbs the output from its mean

Thus NLL is like squared loss

Probabilistic Linear Regression

- For all the training data, we can write the above model in matrix-vector notation

$\boldsymbol{y} = [y_1; y_2; \dots; y_N]$ is the $N \times 1$ response vector

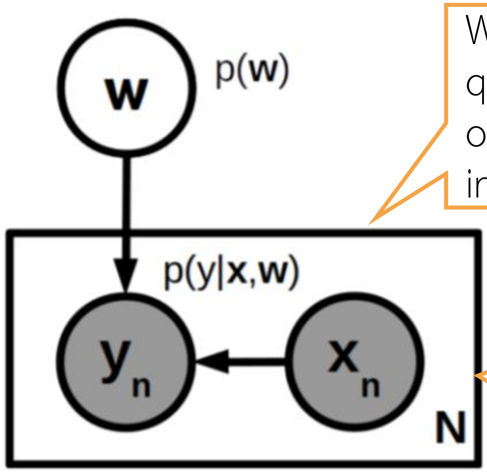
$\boldsymbol{X} = [\boldsymbol{x}_1^\top; \boldsymbol{x}_2^\top; \dots; \boldsymbol{x}_N^\top]$ is the $N \times D$ input matrix

$\boldsymbol{\epsilon} = [\epsilon_1; \epsilon_2; \dots; \epsilon_N]$ is the $N \times 1$ noise vector drawn from $\mathcal{N}(\boldsymbol{0}, \beta^{-1}\boldsymbol{I}_N)$

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w} + \boldsymbol{\epsilon}$$

Same as writing $p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w}, \beta) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{X}\boldsymbol{w}, \beta^{-1}\boldsymbol{I}_N)$

- This is a linear Gaussian model with $\boldsymbol{w}$ being the unknown Gaussian r.v.
- A simple "plate diagram" for this model would look like this (hyperparameters not shown in the diagram)



Direction of arrow show dependency

White nodes denote unknown quantities, grey nodes denote observed quantities (training input-output pairs)

The plate/box with number N shows that we have N such i.i.d. observations

On compact notations..

- When writing the likelihood (assuming $\boldsymbol{y_n}$'s are i.i.d. given $\boldsymbol{w}$ and $\boldsymbol{x}_n$)

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

$$= \mathcal{N}(\boldsymbol{y}|\boldsymbol{X}\boldsymbol{w}, \beta^{-1}\boldsymbol{I}_N)$$

- Thus a product of $\boldsymbol{N}$ univariate Gaussians here (not always) is equivalent to an $\boldsymbol{N}$-dim Gaussian over the vector $\boldsymbol{y} = [y_1, y_2, \ldots, y_N]$

- We will prefer to use this equivalence at other places too whenever we have multiple i.i.d. random variables, each having a univariate Gaussian distribution

Prior on weights

- Assume a zero-mean Gaussian prior on $\boldsymbol{w}$

$$p(\boldsymbol{w}|\lambda) = \prod_{d=1}^{D} p(w_d|\lambda) = \prod_{d=1}^{D} \mathcal{N}(w_d|0, \lambda^{-1})$$

$$= \mathcal{N}(\boldsymbol{w}|\boldsymbol{0}, \lambda^{-1}\mathbf{I}_D)$$

$$\propto \left(\frac{\lambda}{2\pi}\right)^{\frac{D}{2}} \exp\left[-\frac{\lambda}{2}\boldsymbol{w}^\top\boldsymbol{w}\right]$$

May also use a non-zero mean Gaussian prior, e.g., $\mathcal{N}(w_d|\mu, \lambda^{-1})$ if we expect weights to be close to some value μ

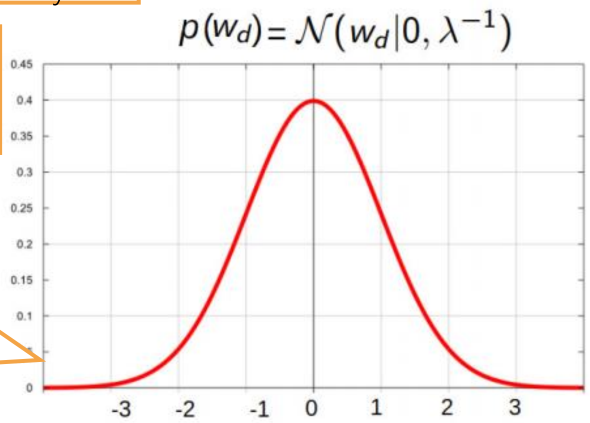
This prior assumes that *a priori* each weight has a small value (close to zero)

λ controls the uncertainty around our prior belief about value of w_d

In zero-mean case, λ sort of denotes each feature's importance. Think why?

Large λ means more aggressive push towards zero

The precision λ controls how aggressively the prior pushes w_d towards mean (0)



Can also use a full covariance matrix Λ^{-1} for the prior to impose a priori correlations among different weights

Prior's hyperparameters ($\lambda/\Lambda/\mu$) etc can be learned as well using point estimation (e.g., MLE-II) or fully Bayesian inference

- Zero-mean Gaussian prior corresponds to ℓ_2 regularizer

Reason: The negative log prior $-\log p(\boldsymbol{w}) \propto \frac{\lambda}{2}\boldsymbol{w}^\top\boldsymbol{w}$

The Posterior

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

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

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

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

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

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

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

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

The form is also similar to the solution to ridge regression $\arg\min_w ||y - Xw||^2 + \lambda w^\top w = (X^\top X + \lambda I)^{-1}X^\top y$

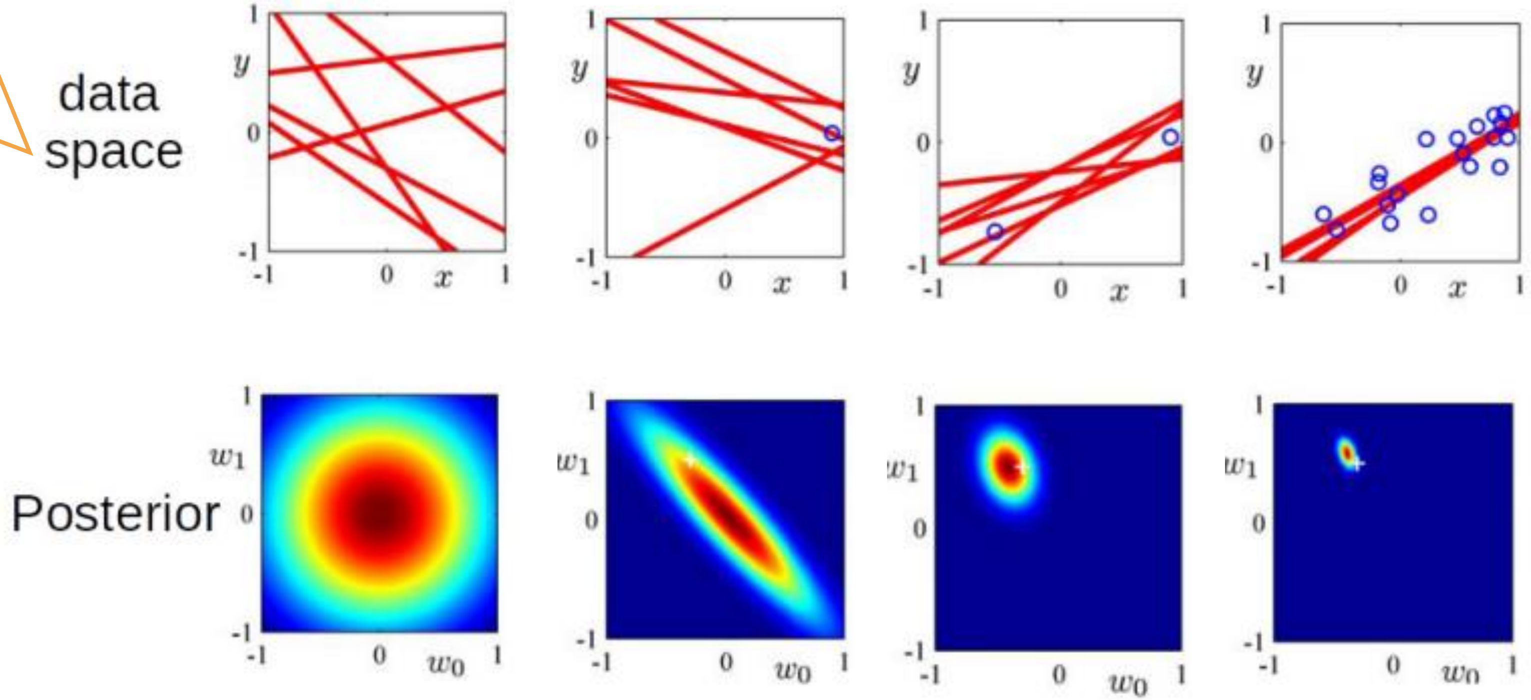
MAP solution turns out to be exactly the same (reason: Gaussian's mean and mode are the same)

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

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

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

$p(y_*|\mathbf{x}_*, \mathbf{w}, \beta)$: $\mathcal{N}(y_*|\boldsymbol{w}^\top \boldsymbol{x}_*, \beta^{-1})$

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

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 LGM results

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

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 $\boldsymbol{w}$)

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

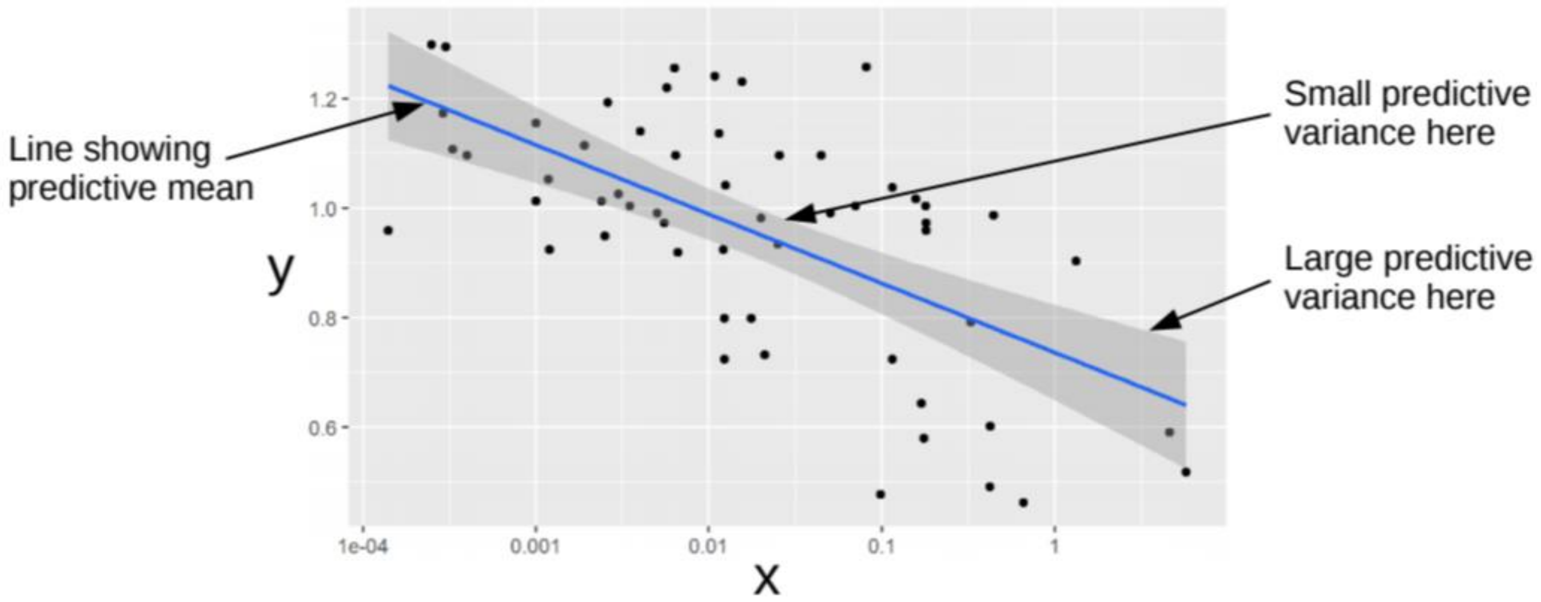
$$p(y_*|\mathbf{x}_*, \mathbf{w}_{MAP}) = \mathcal{N}(\mathbf{w}_{MAP}^\top \mathbf{x}_*, \beta^{-1}) \quad \text{- MAP prediction}$$

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)

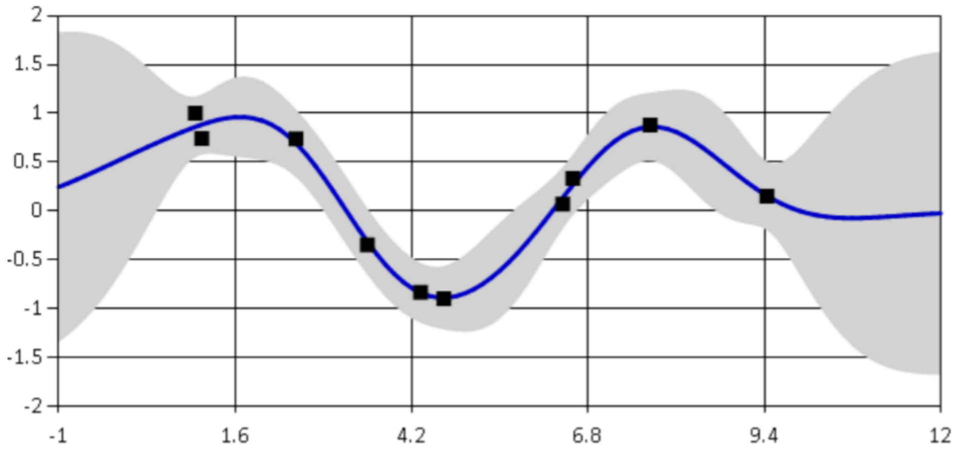
Posterior Predictive Distribution: An Illustration

- 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 $\boldsymbol{\phi(x)}$

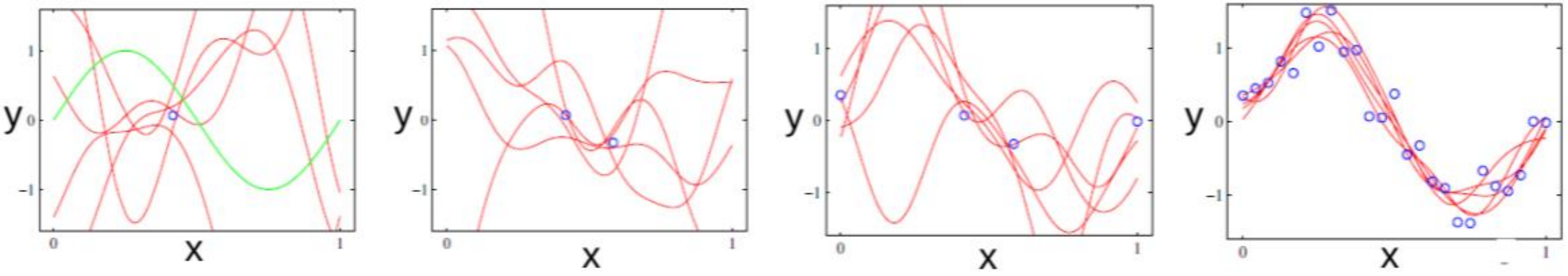
$$p(y|\mathbf{x}, \mathbf{w}) = \mathcal{N}(\mathbf{w}^\top \phi(\mathbf{x}), \beta^{-1})$$

Can be pre-defined (e.g., replace a scalar x by polynomial mapping $[1, x, x^2]$) or extracted by a pretrained deep neural net

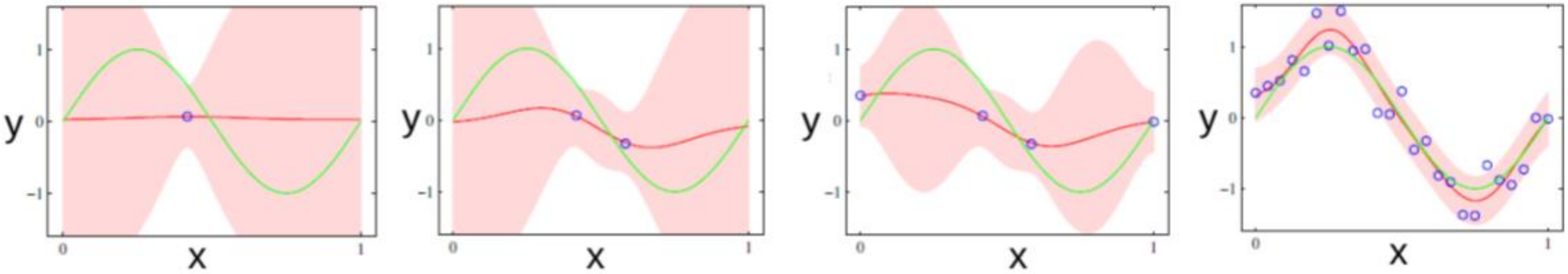
- Alternatively, a kernel function can be used to implicitly define the nonlinear mapping
- More on nonlinear regression when we discuss Gaussian Processes

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



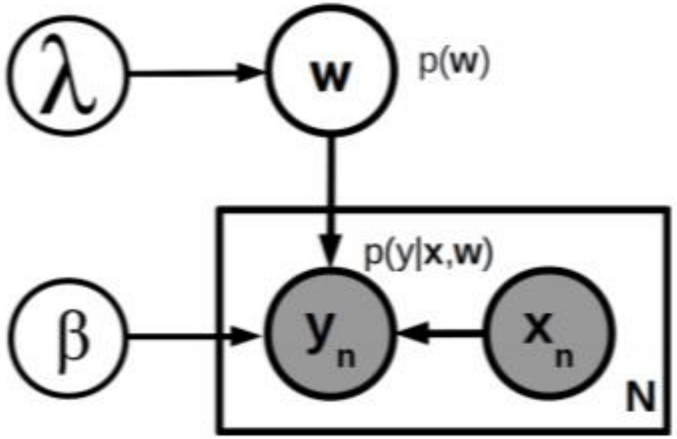
- PPD: Red curve is predictive mean, shaded region denotes predictive uncertainty



Estimating Hyperparameters via MLE-II

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

Need posterior over all the 3 unknowns



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

PPD would require integrating out all 3 unknowns

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

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

- Posterior and PPD computation is intractable.

Called "MLE-II" because we are maximizing marginal likelihood, not the likelihood

- If we just want point estimates for (β, λ) then MLE-II is an option

Will see various other methods like EM, variational inference, MCMC, etc later

And then compute $p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{y}, \hat{\beta}, \hat{\lambda})$ treating $\hat{\beta}, \hat{\lambda}$ as given

$$(\hat{\beta}, \hat{\lambda}) = \text{argmax}_{\beta,\lambda} \log p(\boldsymbol{y}|\boldsymbol{X}, \beta, \lambda)$$

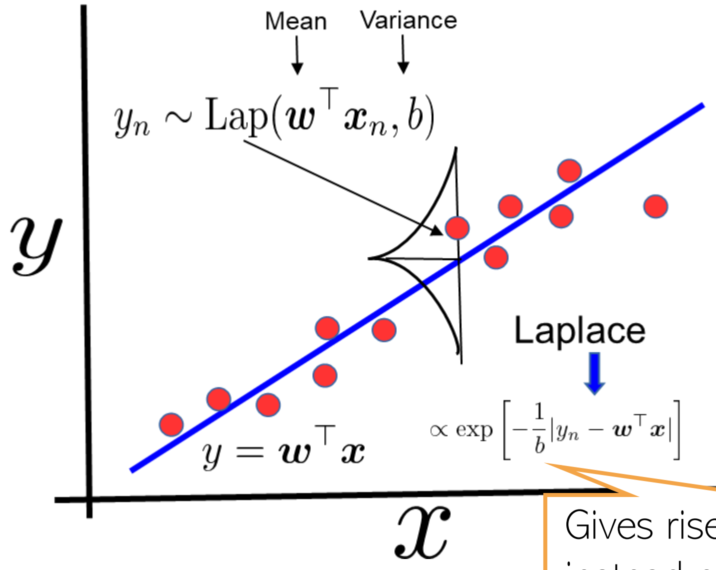
For regression with Gaussian likelihood and Gaussian prior on $\boldsymbol{w}$, the marginal likelihood has an exact expression

Prob. Linear Regression: Some Other Variations

- Can use other likelihoods $p(y_n|\boldsymbol{x}_n, \boldsymbol{w})$ and/or prior distribution $p(\boldsymbol{w})$

- Laplace distribution for the likelihood

$$p(y_n|\boldsymbol{x}_n, \boldsymbol{w}) = \text{Lap}(y_n|\boldsymbol{w}^\top \boldsymbol{x}_n, b)$$



More robust to outliers than squared loss

Gives rise to "absolute loss" instead of squared loss

- Heteroskedastic noise in the likelihood, e.g.,

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

Can even assume β_n to depend on input $\boldsymbol{x}_n$

Different noise distribution $\mathcal{N}(0, \beta_n^{-1})$ for each y_n

- Feature-specific variances in the prior for $\boldsymbol{w}$

$$p(\boldsymbol{w}) = \prod_{d=1}^{D} \mathcal{N}(w_d|0, \lambda_d^{-1}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{0}, \boldsymbol{\Lambda}^{-1})$$

Diagonal precision/covariance matrix with λ_d's along the columns of Λ

This has the effect of having feature-specific regularization

Since we can also learn these precisions (e.g., using MLE-II), using such a prior, we can learn the importance of different features (feature selection) which isn't possible with a $\mathcal{N}(\boldsymbol{w}|\boldsymbol{0}, \lambda^{-1}\mathbf{I})$ prior with spherical covariance