

Probabilistic Modeling of Data (contd)

CS771: Introduction to Machine Learning

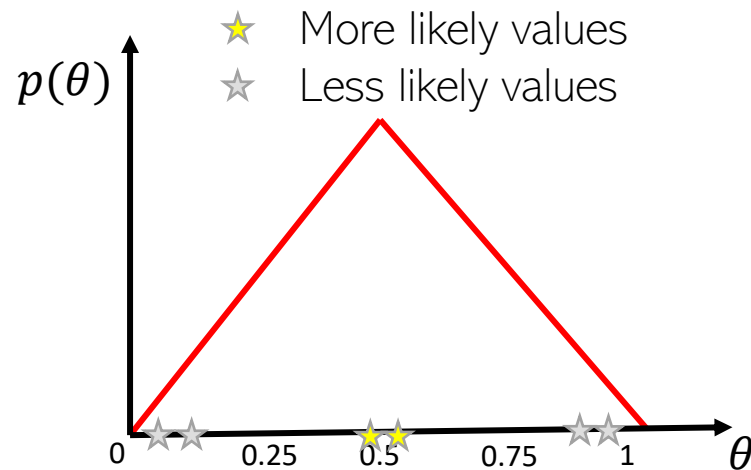
Plan today

- Probabilistic modeling of data
 - MAP estimation
 - Bayesian inference (computing the posterior distribution)
- Supervised learning using probabilistic modeling

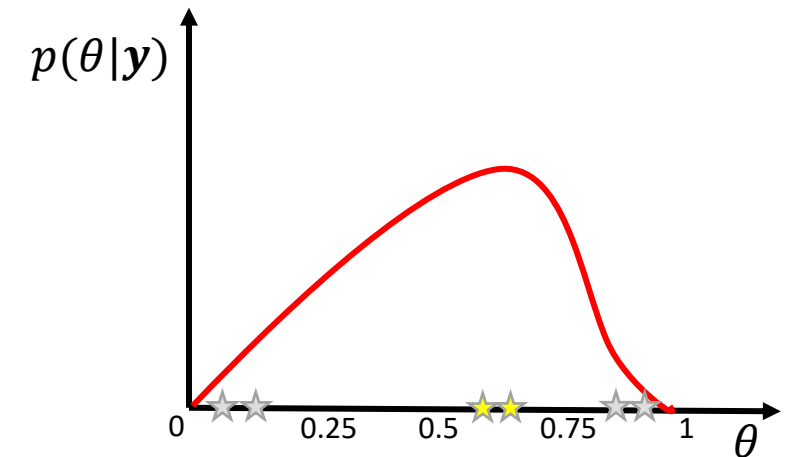


Recap: Prior and Posterior

- Prior distribution reflects our prior belief about the unknown θ
- Posterior distribution reflects our updated belief once we have seen the data



$$p(\theta|\mathbf{y}) = \frac{p(\theta)p(\mathbf{y}|\theta)}{p(\mathbf{y})}$$



- Posterior's maxima is the MAP (maximum-a-posteriori) solution
- Posterior contains more information than just the MAP solution
 - It also tells us about the uncertainty in our estimate of θ



Recap: MLE and MAP

- MLE: Find the optimal parameter θ that maximizing the (log) **likelihood**

$$\theta_{MLE} = \operatorname{argmax}_{\theta} \log p(\mathbf{y}|\theta) = \operatorname{argmax}_{\theta} \sum_{n=1}^N \log p(y_n|\theta)$$

- MAP: Find the optimal parameter θ that has the maximum **posterior probability**

$$\theta_{MAP} = \operatorname{argmax}_{\theta} \log p(\theta|\mathbf{y}) = \operatorname{argmax}_{\theta} [\log p(\mathbf{y}|\theta) + \log p(\theta)]$$

Important: Computing the MAP solution does not require computing the posterior (this maximization can be done even without computing the posterior)!

$$= \operatorname{argmax}_{\theta} \left[\sum_{n=1}^N \log p(y_n|\theta) + \log p(\theta) \right]$$

- MAP's advantage over MLE: The prior acts as a regularizer on θ
- Negative log likelihood is akin to loss function, negative log prior is akin to regularizer

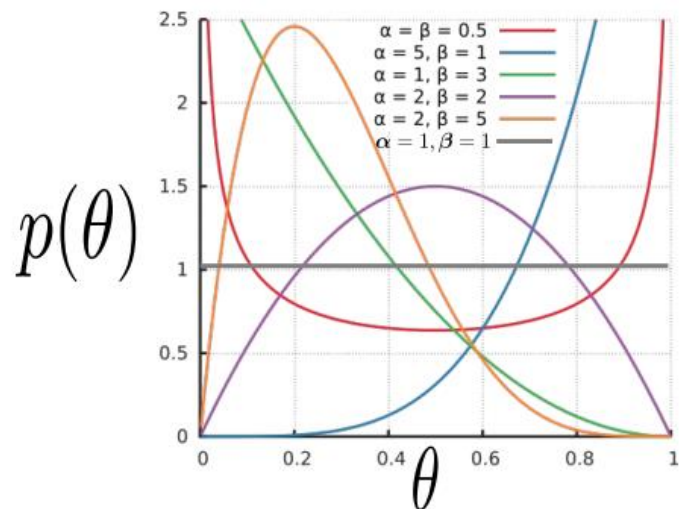


MAP Estimation: An Example

- Let's again consider the coin-toss problem (estimating the bias of the coin)
- Each likelihood term is Bernoulli

$$p(y_n|\theta) = \text{Bernoulli}(y_n|\theta) = \theta^{y_n} (1 - \theta)^{1-y_n}$$

- Also need a prior since we want to do MAP estimation
- Since $\theta \in (0,1)$, a reasonable choice of prior for θ would be [Beta distribution](#)



$$p(\theta|\alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta^{\alpha-1} (1 - \theta)^{\beta-1}$$

The gamma function

Using $\alpha = 1$ and $\beta = 1$ will make the Beta prior a uniform prior

α and β (both non-negative reals) are the two hyperparameters of this Beta prior

Can set these based on intuition, cross-validation, or even learn them

MAP Estimation: An Example (Contd)

- The log posterior for the coin-toss model is log-lik + log-prior

$$LP(\theta) = \sum_{n=1}^N \log p(y_n|\theta) + \log p(\theta|\alpha, \beta)$$

- Plugging in the expressions for Bernoulli and Beta and ignoring any terms that don't depend on θ , the log posterior simplifies to

$$LP(\theta) = \sum_{n=1}^N [y_n \log \theta + (1 - y_n) \log(1 - \theta)] + (\alpha - 1) \log \theta + (\beta - 1) \log(1 - \theta)$$

- Maximizing the above log post. (or min. of its negative) w.r.t. θ gives

Using $\alpha = 1$ and $\beta = 1$ gives us the same solution as MLE

Recall that $\alpha = 1$ and $\beta = 1$ for Beta distribution is in fact equivalent to a uniform prior (hence making MAP equivalent to MLE)

$$\theta_{MAP} = \frac{\sum_{n=1}^N y_n + \alpha - 1}{N + \alpha + \beta - 2}$$

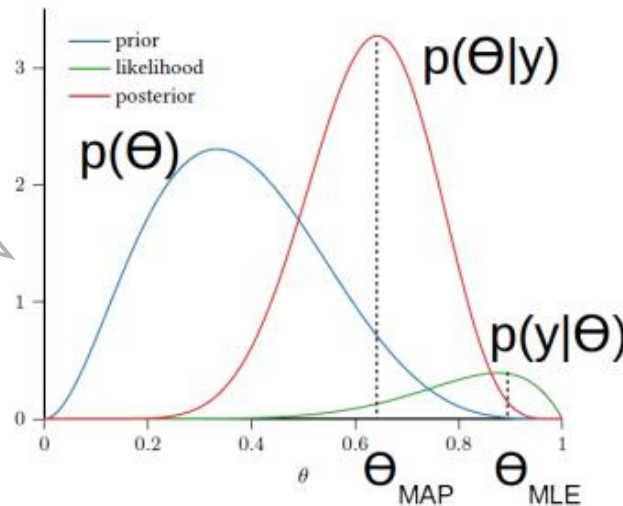
Such interpretations of prior's hyperparameters as being "pseudo-observations" exist for various other prior distributions as well (in particular, distributions belonging to "exponential family" of distributions)

Prior's hyperparameters have an interesting interpretation. Can think of $\alpha - 1$ and $\beta - 1$ as the number of heads and tails, respectively, before starting the coin-toss experiment (akin to "pseudo-observations")

Fully Bayesian Inference

- MLE/MAP only give us a point estimate of θ

MAP estimate is more robust than MLE (due to the regularization effect) but the estimate of uncertainty is missing in both approaches – both just return a single “optimal” solution by solving an optimization problem



Interesting fact to keep in mind: Note that the use of the prior is making the MLE solution move towards the prior (MAP solution is kind of a “compromise between MLE solution of the mode of the prior”) 😊



Fully Bayesian inference

- If we want more than just a point estimate, we can compute the full posterior

Computable analytically only when the prior and likelihood are “friends” with each other (i.e., they form a **conjugate pair** of distributions (distributions from **exponential family** have conjugate priors

$$p(\theta|y) = \frac{p(\theta)p(y|\theta)}{p(y)}$$

An example: Bernoulli and Beta are conjugate. Will see some more such pairs

In other cases, the posterior needs to be approximated (will see 1-2 such cases in this course; more detailed treatment in the advanced course on probabilistic modeling and inference)

Fully Bayesian Inference: An Example

- Let's again consider the coin-toss problem
- Bernoulli likelihood: $p(y_n|\theta) = \text{Bernoulli}(y_n|\theta) = \theta^{y_n} (1 - \theta)^{1-y_n}$
- Beta prior: $p(\theta) = \text{Beta}(\theta|\alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta^{\alpha-1} (1 - \theta)^{\beta-1}$
- The posterior can be computed as

Also, if you get more observations, you can treat the current posterior as the new prior and obtain a new posterior using these extra observations

Posterior in this example is the same distribution as the prior (both Beta), just with updated hyperparameters (property when likelihood and prior are conjugate to each other)



Number of heads (N_1)

Number of tails (N_0)

$$\theta^{\sum_{n=1}^N y_n} (1 - \theta)^{N - \sum_{n=1}^N y_n}$$

$$p(\theta|\mathbf{y}) = \frac{p(\theta)p(\mathbf{y}|\theta)}{p(\mathbf{y})} = \frac{p(\theta) \prod_{n=1}^N p(y_n|\theta)}{p(\mathbf{y})} = \frac{\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta^{\alpha-1} (1-\theta)^{\beta-1} \prod_{n=1}^N \theta^{y_n} (1-\theta)^{1-y_n}}{\int \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta^{\alpha-1} (1-\theta)^{\beta-1} \prod_{n=1}^N \theta^{y_n} (1-\theta)^{1-y_n} d\theta}$$

This is the numerator integrated/marginalized over θ : $p(\mathbf{y}) = \int p(\theta, \mathbf{y}) d\theta = \int p(\theta) p(\mathbf{y}|\theta) d\theta$

In general, hard but with conjugate pairs of prior and likelihood, we don't need to compute this, as we will see in this example ☺

Parts coming from the numerator, which consist of θ terms. We have ignored other constants in the numerator, and the whole denominator which is also constant w.r.t. θ

$$\propto \theta^{\alpha+N_1-1} (1 - \theta)^{\beta+N_0-1}$$

Aha! This is nothing but **Beta($\theta|\alpha + N_1, \beta + N_0$)**

This, of course, is not always possible but only in simple cases like this

Found the posterior just by simple inspection without having to calculate the constant of proportionality ☺

Conjugacy

- Many pairs of distributions are conjugate to each other
 - Bernoulli (likelihood) + Beta (prior) \Rightarrow Beta posterior
 - Binomial (likelihood) + Beta (prior) \Rightarrow Beta posterior
 - Multinomial (likelihood) + Dirichlet (prior) \Rightarrow Dirichlet posterior
 - Poisson (likelihood) + Gamma (prior) \Rightarrow Gamma posterior
 - Gaussian (likelihood) + Gaussian (prior) \Rightarrow Gaussian posterior
 - and many other such pairs ..
- Tip: If two distr are conjugate to each other, their functional forms are similar
 - Example: Bernoulli and Beta have the forms

$$\text{Bernoulli}(y|\theta) = \theta^y (1 - \theta)^{1-y}$$

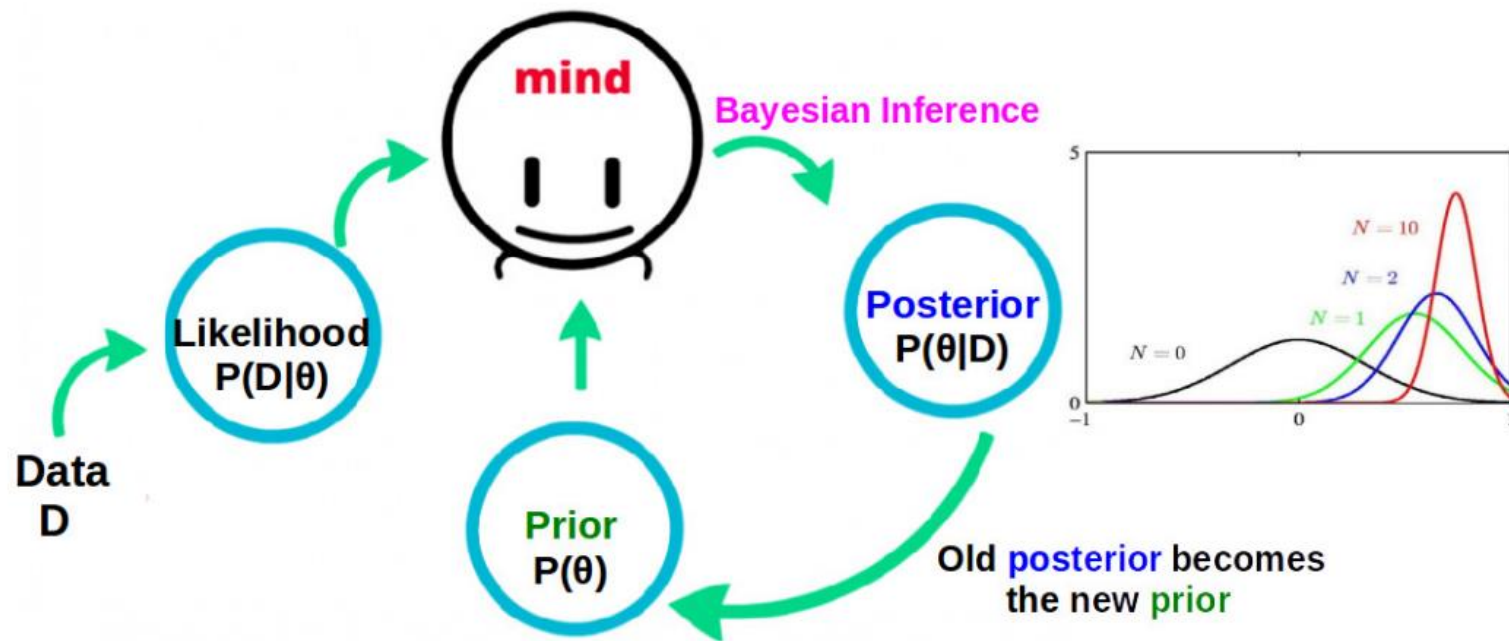
$$\text{Beta}(\theta|\alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta^{\alpha-1} (1 - \theta)^{\beta-1}$$

This is why, when we multiply them while computing the posterior, the exponents get added and we get the same form for the posterior as the prior but with just updated hyperparameter. Also, we can identify the posterior and its hyperparameters simply by inspection



“Online” Nature of Bayesian Inference

- Fully Bayesian inference fits naturally into an “online” learning setting



Also, the posterior becomes more and more “concentrated” as the number of observations increases. For very large N , you may expect it to be peak around the MLE solution



- Our belief about θ keeps getting updated as we see more and more data



Probabilistic Models: Making Predictions

- Having estimated θ (MLE/MAP/posterior), we can now use it to make predictions
- Prediction entails computing posterior predictive distribution of a new observation y_*

$$p(y_*|\mathbf{y}) = \int p(y_*, \theta|\mathbf{y})d\theta$$

Marginalizing (summing/integrating) over the unknown random variable θ

Conditional distribution of the new observation, given past observations

$$= \int p(y_*|\theta, \mathbf{y})p(\theta|\mathbf{y})d\theta$$

Decomposing the joint using chain rule

In some simple cases, this can be computed exactly but, in general, it can't be and needs to be approximated

This step assumes i.i.d. data, i.e., given θ , y_* does not depend on \mathbf{y}

$$= \int p(y_*|\theta)p(\theta|\mathbf{y})d\theta$$

$$= \mathbb{E}_{p(\theta|\mathbf{y})} [p(y_*|\theta)]$$

This computes the predictive distribution by averaging over the full posterior – basically calculates $p(y_*|\theta)$ for each possible θ , weighs it by the probability of θ under the posterior $p(\theta|\mathbf{y})$, and sums all such posterior weighted predictions. Note that not each value of θ is given equal importance here in the averaging

- When doing MLE/MAP, we approximate the posterior $p(\theta|\mathbf{y})$ by a single point θ_{opt}

$$p(y_*|\mathbf{y}) = \int p(y_*|\theta)p(\theta|\mathbf{y})d\theta \approx p(y_*|\theta_{opt})$$

A "plug-in predictive distribution" – simply plugged in the single best estimate (MLE/MAP) that we have

- Plug-in prediction which uses MLE/MAP of θ is cheaper since no integral involved



Making Predictions: An Example

- For coin-toss example, prediction means computing $p(y_{N+1} = 1|\mathbf{y})$
- This can be done using the MLE/MAP estimate, or using the full posterior

$$\theta_{MLE} = \frac{N_1}{N} \quad \theta_{MAP} = \frac{N_1 + \alpha - 1}{N + \alpha + \beta - 2} \quad p(\theta|\mathbf{y}) = \text{Beta}(\theta|\alpha + N_1, \beta + N_0)$$

- Thus for this example (where observations are assumed iid from a Bernoulli)

$$\text{MLE prediction: } p(y_{N+1} = 1|\mathbf{y}) = \int p(y_{N+1} = 1|\theta)p(\theta|\mathbf{y})d\theta \approx p(y_{N+1} = 1|\theta_{MLE}) = \theta_{MLE} = \frac{N_1}{N}$$

$$\text{MAP prediction: } p(y_{N+1} = 1|\mathbf{y}) = \int p(y_{N+1} = 1|\theta)p(\theta|\mathbf{y})d\theta \approx p(y_{N+1} = 1|\theta_{MAP}) = \theta_{MAP} = \frac{N_1 + \alpha - 1}{N + \alpha + \beta - 2}$$

$$\text{Fully Bayesian: } p(y_{N+1} = 1|\mathbf{y}) = \int p(y_{N+1} = 1|\theta)p(\theta|\mathbf{y})d\theta = \int \theta p(\theta|\mathbf{y})d\theta = \int \theta \text{Beta}(\theta|\alpha + N_1, \beta + N_0)d\theta = \frac{N_1 + \alpha}{N + \alpha + \beta}$$



Again, keep in mind that the posterior weighted averaged prediction used in the fully Bayesian case would usually not be as simple to compute as it was in this case. We will look at some hard cases later

Expectation of θ under the Beta posterior that we computed using fully Bayesian inference

Predictive Distribution: About notation

- We used the following notation for the predictive distribution

$$p(y_* | \mathbf{y}) = \int p(y_* | \theta) p(\theta | \mathbf{y}) d\theta$$

Posterior predictive distribution (PPD)

$$p(y_* | \mathbf{y}) \approx p(y_* | \theta_{opt})$$

Plug-in predictive distribution

An approximation of the PPD using a single best value θ_{opt}

- In some case, as we will see, it may additionally depend on other given quantities

$$p(y_* | \mathbf{x}_*, \mathbf{X}, \mathbf{y})$$

The PPD for a supervised learning model: Depends on the test input \mathbf{x}_* and the training data \mathbf{X}, \mathbf{y}

$$p(y_* | \mathbf{x}_*, \theta_{opt})$$

The plug-in predictive for a supervised learning model: depends on the test input \mathbf{x}_* and the point estimate of θ

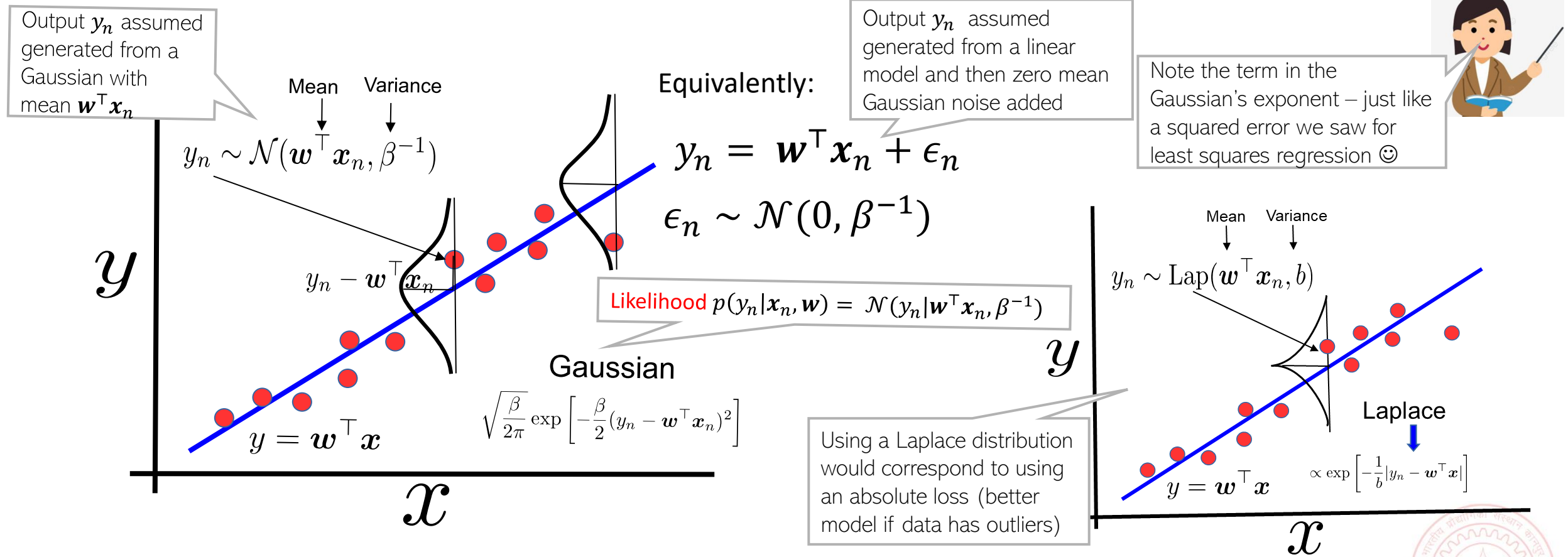


Probabilistic Models for Supervised Learning



Why Probabilistic Models for Supervised Learning?

- We can use a probability distribution to model the input-output relationship



- Depending on the nature of the data, we can choose a suitable likelihood model (and it naturally corresponds to a loss function, e.g., squared, cross-entropy, etc)

Why Probabilistic Models for Supervised Learning?¹⁶

- We can use suitable priors for the model parameters \mathbf{w} (and these priors naturally correspond to regularizers) and also compute the posterior of \mathbf{w}

In form of $p(\mathbf{y}_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y})$ or $p(\mathbf{y}_*|\mathbf{x}_*, \mathbf{w}_{opt})$

- For test input \mathbf{x}_* , we can compute distribution over the predicted label
 - This gives us not just a single “mean” prediction \mathbf{y}_* but also its uncertainty/variance
 - Variance in the prediction of \mathbf{y}_* can be used as a measure of confidence
 - Variance/confidence in predictions is useful in many domains such as healthcare, and in [active learning](#) (using a learned model to collect more training data to improve it)
- There are various other benefits (will see later)
 - Handling missing data (missing features, missing labels, etc)
 - Hyperparameter estimation



Prob. Models for Supervised Learning

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



- Goal: Learn the conditional distribution $p(\mathbf{y}|\mathbf{x})$. Broadly, two approaches

Discriminative Approach

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

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

Some examples

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

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

Generative Approach

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

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

- Both approaches have their pros and cons (discussed later)

The Discriminative Approach

- This approach models $p(y|\mathbf{x})$ directly using a suitable prob. distribution, e.g.,

Regression
model likelihood

$$p(y_i|\mathbf{x}_i, \mathbf{w}) = \mathcal{N}(y_i|\mathbf{w}^\top \mathbf{x}_i, \beta^{-1}) = \sqrt{\frac{\beta}{2\pi}} \exp \left[-\frac{\beta}{2} (y_i - \mathbf{w}^\top \mathbf{x}_i)^2 \right]$$

Binary classification
model likelihood

$$p(y_i|\mathbf{x}_i, \mathbf{w}) = \text{Bernoulli}(y_i|\sigma(\mathbf{w}^\top \mathbf{x}_i)) = \mu_i^{y_i} (1 - \mu_i)^{1-y_i}$$

$\mu_i = \sigma(\mathbf{w}^\top \mathbf{x}_i)$

- The negative log-likelihood (assuming i.i.d. outputs) for the above two models

$$NLL(\mathbf{w}) = \sum_{i=1}^N -\log p(y_i|\mathbf{x}_i, \mathbf{w}) = \sum_{i=1}^N -\log \sqrt{\frac{\beta}{2\pi}} \exp \left[-\frac{\beta}{2} (y_i - \mathbf{w}^\top \mathbf{x}_i)^2 \right]$$

$$NLL(\mathbf{w}) = \frac{\beta}{2} \sum_{n=1}^N (y_i - \mathbf{w}^\top \mathbf{x}_i)^2 \quad (\text{same as squared loss } \text{😊})$$

Thus minimization of NLL
(i.e., doing MLE) is
equivalent to minimization
of the training loss



$$NLL(\mathbf{w}) = \sum_{i=1}^N -\log p(y_i|\mathbf{x}_i, \mathbf{w}) = \sum_{i=1}^N -\log \mu_i^{y_i} (1 - \mu_i)^{1-y_i}$$

$$NLL(\mathbf{w}) = -\sum_{n=1}^N [y_i \log \mu_i + (1 - y_i) \log(1 - \mu_i)] \quad (\text{same as cross-entropy loss } \text{😊})$$

Lacks regularization (and
thus can overfit) but we
can use a prior on \mathbf{w} to
do MAP estimation



A prior over \mathbf{w}

- For MAP estimation (and to compute full posterior), we need a prior $\mathbf{w} \in \mathbb{R}^D$
- A reasonable prior for real-valued vectors can be a multivariate Gaussian

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{w}_0, \Sigma)$$

Equivalent to saying that *a priori* we expect the solution to be close to some vector \mathbf{w}_0

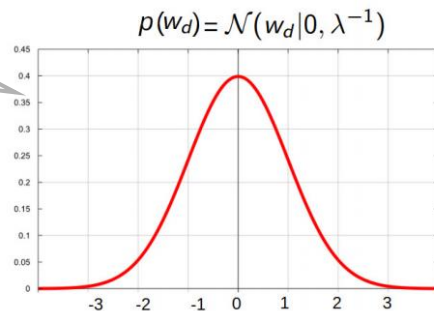
Σ specifies how strong our belief is that \mathbf{w} to be close to \mathbf{w}_0

- A specific example of a multivariate Gaussian prior in this problem

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

Omitting λ for brevity

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



This is essentially like a regularizer that pushes elements of \mathbf{w} to be small (we will see shortly)

Equivalent to saying that *a priori* we expect each element of the solution to be close to 0 (i.e., "small")

$$\mathcal{N}(w_d|0, \lambda^{-1}) = \sqrt{\frac{\lambda}{2\pi}} \exp\left[-\frac{\lambda}{2} w_d^2\right]$$

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

Aha! This $\mathbf{w}^T \mathbf{w}$ term reminds me of the ℓ_2 regularizer 😊



That's indeed the case 😊



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

- The MAP objective (log-posterior) will be the log-likelihood + $\log p(\mathbf{w})$
- Ignoring terms that don't depend on \mathbf{w} the log-posterior will be

$$NLL(\mathbf{w}) - \log \exp \left[-\frac{\lambda}{2} \mathbf{w}^\top \mathbf{w} \right] = NLL(\mathbf{w}) + \frac{\lambda}{2} \mathbf{w}^\top \mathbf{w}$$

- Exercise: For regression with likelihood $p(y_i|\mathbf{x}_i, \mathbf{w}) = \mathcal{N}(y_i|\mathbf{w}^\top \mathbf{x}_i, \beta^{-1})$

$$\hat{\mathbf{w}}_{MAP} = \operatorname{argmin}_{\mathbf{w}} \frac{\beta}{2} \sum_{i=1}^N (y_i - \mathbf{w}^\top \mathbf{x}_i)^2 + \frac{\lambda}{2} \mathbf{w}^\top \mathbf{w} = (\mathbf{X}^\top \mathbf{X} + \frac{\lambda}{\beta} \mathbf{I}_D)^{-1} \mathbf{X}^\top \mathbf{y}$$

- Classification with likelihood $p(y_i|\mathbf{x}_i, \mathbf{w}) = \text{Bernoulli}(y_i|\sigma(\mathbf{w}^\top \mathbf{x}_i))$ is the same as logistic regression. When also using the above prior, the MAP solution will be the same as cross-entropy loss minimization + L2 regularization on the weights



Prob. Linear Regression: The Full Posterior

- If we want the **full posterior distribution** over \mathbf{w} , it can be computed as

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

For brevity, we have not shown the dependence of the various distributions here on the hyperparameters λ and β

- For regression, with Gaussian likelihood and zero mean Gaussian prior are conjugate (both Gaussians), and the posterior will be Gaussian

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

$$\boldsymbol{\mu}_N = (\mathbf{X}^\top \mathbf{X} + \frac{\lambda}{\beta} \mathbf{I}_D)^{-1} \mathbf{X}^\top \mathbf{y}$$

$$\boldsymbol{\Sigma}_N = (\beta \mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_D)^{-1}$$

Will provide the proof separately but it is straightforward when using properties of Gaussian

Posterior's mean is the same as the MAP solution since the mean and mode of a Gaussian are the same!

Note: λ and β are assumed to be fixed; otherwise, the problem is a bit harder (beyond the scope of CS771)

- For binary classification with Bernoulli likelihood and zero mean Gaussian prior, we don't have conjugacy, and the posterior can't be computed in closed form



Logistic Regression: The Full Posterior

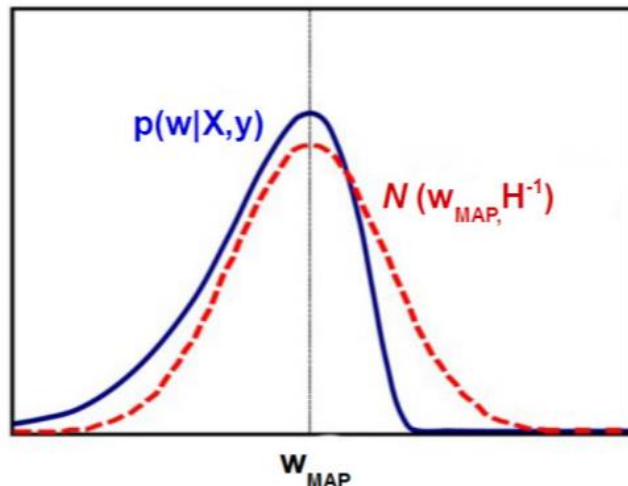
- The posterior distribution for the logistic regression model

$$p(\mathbf{w}|\mathbf{X}, \mathbf{y}) = \frac{p(\mathbf{w})p(\mathbf{y}|\mathbf{X}, \mathbf{w})}{p(\mathbf{y}|\mathbf{X})} = \frac{\overset{\text{Gaussian}}{p(\mathbf{w})} \prod_{n=1}^N \overset{\text{Bernoulli}}{p(y_n|\mathbf{w}, \mathbf{x}_n)}}{\int p(\mathbf{w}) \prod_{n=1}^N p(y_n|\mathbf{w}, \mathbf{x}_n) d\mathbf{w}}$$

Unfortunately, Gaussian and Bernoulli are not conjugate with each other, so analytic expression for the posterior can't be obtained unlike prob. linear regression



- Need to approximate the posterior in this case
- We will use a simple approximation called **Laplace approximation**



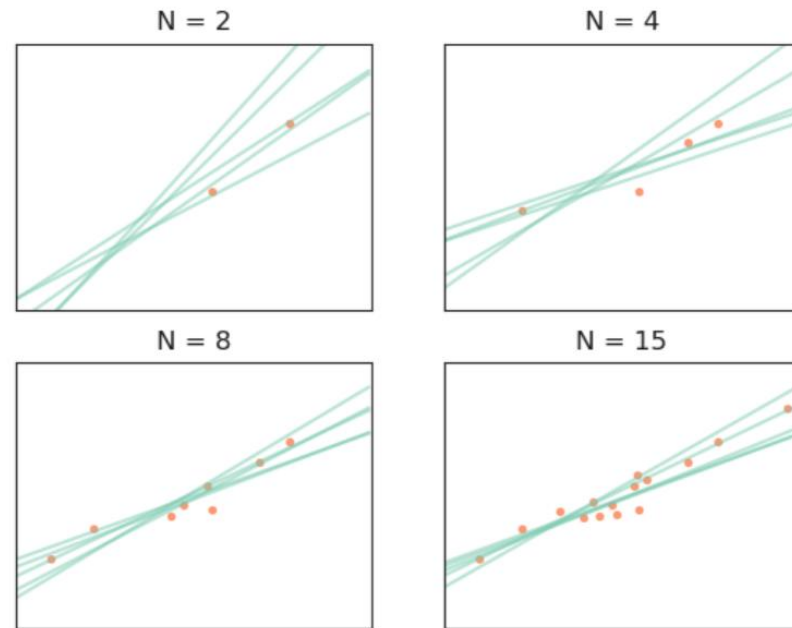
Approximates the posterior of \mathbf{w} by a Gaussian whose mean is the MAP solution $\hat{\mathbf{w}}_{MAP}$ and covariance matrix is the inverse of the Hessian (Hessian: **second derivative of the negative log-posterior of the LR model**)

Can also employ more advanced posterior approximation methods, like MCMC and variational inference (beyond the scope of CS771)



What does posterior of a linear model look like ? ²³

- Each sample from posterior $p(\mathbf{w}|\mathbf{X}, \mathbf{y}) = \mathcal{N}(\boldsymbol{\mu}_N, \boldsymbol{\Sigma}_N)$ will give a weight vector \mathbf{w}
 - In case of lin. reg., each weight vector corresponds to a regression line



The posterior sort of represents an ensemble of solutions (not all are equally good but we can use all of them in an “importance-weighted” fashion to make the prediction using the posterior predictive distribution)

Importance of each solution in this ensemble is its posterior probability $p(\mathbf{w}|\mathbf{X}, \mathbf{y})$

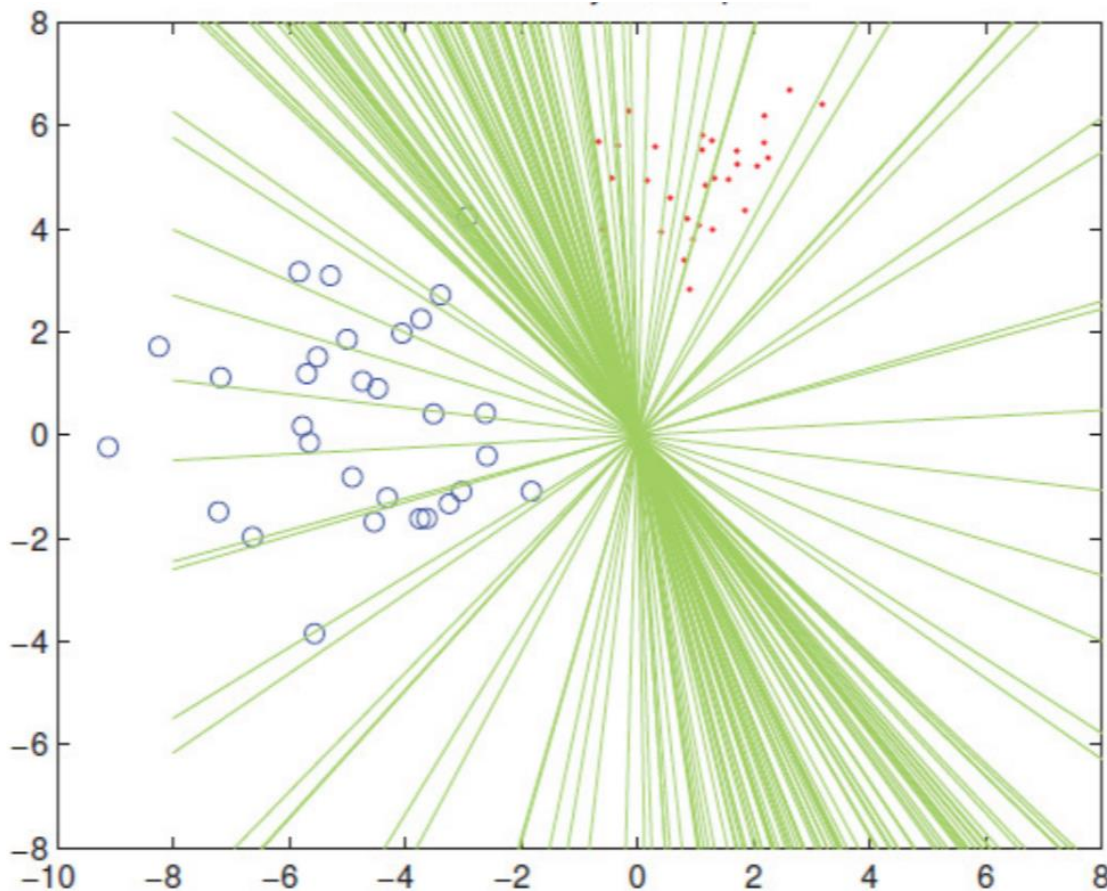


- Each weight vector will give a different set of predictions on test data
 - These different predictions will give us a variance (uncertainty) estimate in model's prediction
 - The uncertainty decreases as N increases (we become more sure when we see more training data)



What does posterior of a linear model look like ? ²⁴

- Can also sample from the Laplace approximate posterior of logistic regression
- Each sample will give a weight vec defining a hyperplane separator



Not all separators are equally good; their goodness depends on their posterior probabilities

When making predictions, we can still use all of them but weighted by their importance based on their posterior probabilities

That's exactly what we do when computing the predictive distribution



Prob. Linear Regression: Predictive Distribution

- Want the predictive distribution $p(y_* | \mathbf{x}_*, \mathbf{X}, \mathbf{y})$ of the output y_* for a new input \mathbf{x}_* .
- With MLE/MAP estimate of \mathbf{w} , we will use the plug-in predictive

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

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

- If we have the full posterior over \mathbf{w} , can compute the **posterior predictive dist.**

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

Assuming the hyperparameters λ and β are known (otherwise, the PPD can't be computed exactly)

- Requires an integral but has a closed form

$$p(y_* | \mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N}(\mu_N^\top \mathbf{x}_*, \beta^{-1} + \mathbf{x}_*^\top \Sigma_N \mathbf{x}_*)$$

Mean prediction

Input-specific predictive variance unlike the MLE/MAP based predictive where it was β^{-1} (and was same for all test inputs)

Will provide the proof separately but it is straightforward when using properties of Gaussian

- Input-specific predictive uncertainty useful in problems where we want confidence estimates of the predictions made by the model (e.g., Active Learning)

Logistic Regression: Predictive Distribution

- When using MLE/MAP solution $\hat{\mathbf{w}}_{opt}$, can use the [plug-in predictive distribution](#)

$$p(y_* = 1 | \mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \int p(y_* = 1 | \mathbf{w}, \mathbf{x}_*) p(\mathbf{w} | \mathbf{X}, \mathbf{y}) d\mathbf{w} \\ \approx p(y_* = 1 | \hat{\mathbf{w}}_{opt}, \mathbf{x}_*) = \sigma(\hat{\mathbf{w}}_{opt}^\top \mathbf{x}_*)$$

$$p(y_* | \mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \text{Bernoulli}[\sigma(\hat{\mathbf{w}}_{opt}^\top \mathbf{x}_*)]$$

- When using fully Bayesian inference, we must compute the posterior predictive

$$p(y_* = 1 | \mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \int p(y_* = 1 | \mathbf{w}, \mathbf{x}_*) p(\mathbf{w} | \mathbf{X}, \mathbf{y}) d\mathbf{w}$$

Integral not tractable and must be approximated

sigmoid

Gaussian (if using Laplace approx.)

Monte-Carlo approximation of this integral is one possible way

Generate M samples $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_M$, from the Gaussian approx. of posterior and use $p(y_* = 1 | \mathbf{x}_*, \mathbf{X}, \mathbf{y}) \approx \frac{1}{M} \sum_{m=1}^M p(y_* = 1 | \mathbf{w}_m, \mathbf{x}_*) = \frac{1}{M} \sum_{m=1}^M \sigma(\mathbf{w}_m^\top \mathbf{x}_*)$



Probabilistic Modeling: A Summary

- Likelihood corresponds to a loss function; prior corresponds to a regularizer
- Can choose likelihoods and priors based on the nature/property of data/parameters
- MLE estimation = unregularized loss function minimization
- MAP estimation = regularized loss function minimization
- Allows us to do fully Bayesian learning (learning the full distribution of the parameters)
- Makes robust predictions by posterior averaging (rather than using point estimate)
- Many other benefits, such as
 - Estimate of confidence in the model's prediction (useful for doing [Active Learning](#))
 - Can do automatic model selection, hyperparameter estimation, handle missing data, etc.
 - Formulate latent variable models
 - .. and many other benefits (a proper treatment deserves a separate course, but we will see some of these in this course, too)

