Probabilistic ML: Some Basic Ideas

CS772A: Probabilistic Machine Learning

Piyush Rai

Probabilistic Modeling of Data: The Setup

- We are given some training data $\mathcal{D}$
- For supervised learning, $\mathcal{D}$ contains N input-label pairs $(\boldsymbol{x}_i, y_i)_{i=1}^N$
- For unsupervised learning, $\mathcal{D}$ contains N inputs $(\boldsymbol{x}_i)_{i=1}^N$
- Other settings are also possible (e.g., semi-sup., reinforcement learning, etc)
- Assume that the observations are generated by a probabilistic distribution
  - For now, assume the form of the distribution to be known (e.g. a Gaussian)
- Parameters of this distribution, collectively denoted by $\boldsymbol{\theta}$ are unknown
- Our goal is to estimate the distribution (and thus $\boldsymbol{\theta}$) using training data
- Once the distribution is estimated, we can do things such as
  - Predict labels of new inputs, along with our confidence in these predictions
  - Generate new data with similar properties as training data
  - .. and a lot of other useful tasks, e.g., detecting outliers

Probabilistic Modeling of Data: The Setup

- We will denote the data distribution as $p_\theta(\mathcal{D})$ or $p(\mathcal{D}|\theta)$
- Assume that, conditioned on θ, observations are independently and identically distributed (i.i.d. assumption). Depending on the problem, this may look like:

Supervised generative model (both inputs and output are modeled using a distribution)

$$(\boldsymbol{x}_n, y_n) \overset{\text{i.i.d.}}{\sim} p(\boldsymbol{x}, y|\theta) \Rightarrow p(\mathcal{D}|\theta) = \prod_{i=1}^{N} p(\boldsymbol{x}_i, y_i|\theta)$$

Supervised discriminative model (only the output is modeled using a distribution); input is assumed "given" and not modeled

$$y_n \overset{\text{i.i.d.}}{\sim} p(y|\boldsymbol{x}, \theta) \Rightarrow p(\mathcal{D}|\theta) = \prod_{i=1}^{N} p(y_i|\boldsymbol{x}_i, \theta)$$

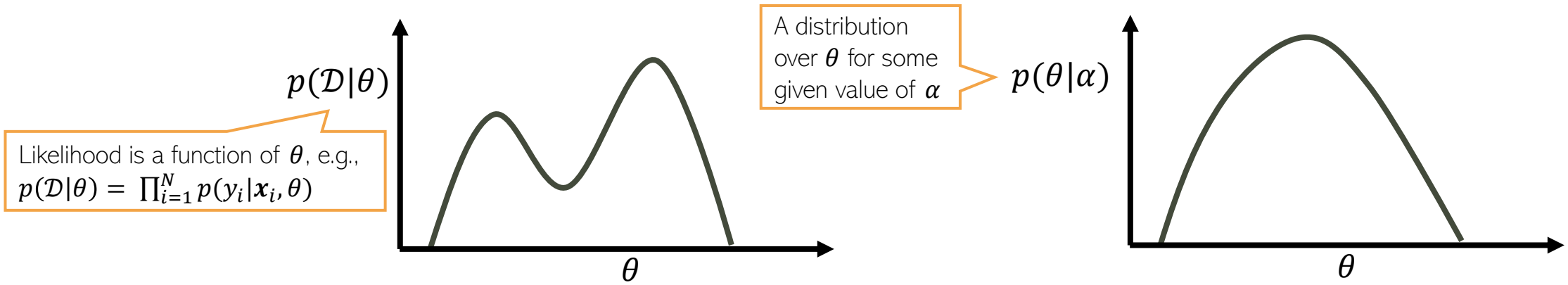
Unsupervised generative model (there are only inputs; no labels)

$$\boldsymbol{x}_n \overset{\text{i.i.d.}}{\sim} p(\boldsymbol{x}|\theta) \Rightarrow p(\mathcal{D}|\theta) = \prod_{i=1}^{N} p(\boldsymbol{x}_i|\theta)$$

- Assume that both training and test data come from the same distribution
  - This assumption, although standard, may be violated in real-world applications of ML and there are "adaptation" methods to handle that

Probabilistic ML Modeling: The Basic Ingredients

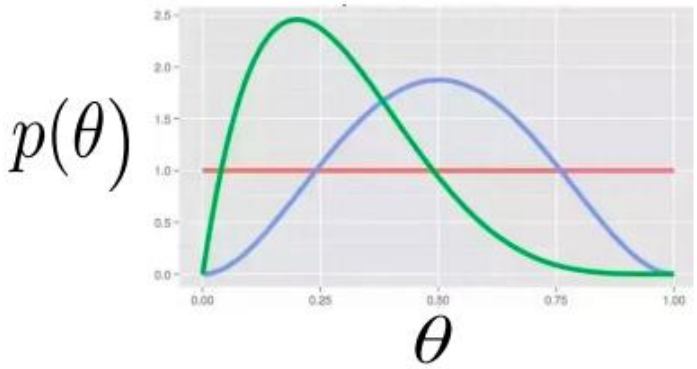
- Likelihood model $p(\mathcal{D}|\theta)$ for data $\mathcal{D}$; prior distribution $p(\theta|\alpha)$ over parameters θ



- Likelihood defined in terms of distribution(s) we assume data is generated from
  - It's like a measure of "fit" between observed data and each possible value of parameters
  - Its negative is like the "loss function" (high likelihood value = low loss; and vice-versa)
- Prior specifies our prior knowledge about θ before we have seen the data
  - It also acts as a regularizer for θ (will see the reason formally later)
- Note: The prior itself depends on other parameters α (also unknown)
  - These are sometimes called "hyperparameters" (can set by hand or estimate from data)

The Prior: Where does it come from?

- The prior $p(\theta|\alpha)$ plays an important role in probabilistic/Bayesian modeling
  - Reflects our prior beliefs about possible parameter values before seeing the data



  - Can be "subjective" or "objective" (also a topic of debate, which we won't get into)
  - Subjective: Prior (our beliefs) derived from past experiments
  - Objective: Prior represents "neutral knowledge" (e.g.. uniform, vague prior)
  - Can also be seen as a regularizer (connection with non-probabilistic view)

Parameter Estimation

- The parameters θ are unknown and need to be estimated from training data $\mathcal{D}$
- When estimating θ, we may take one of the following approaches

Approach 1

- θ has an unknown with fixed value
- Estimate the single best estimate of θ by optimizing a loss function

$$\hat{\theta} = \text{argmin}_{\theta}\, \mathcal{L}(\mathcal{D}; \theta)$$

Approach 2

- Treat θ as a random variable
- Estimate θ by computing its distribution conditioned on $\mathcal{D}$

Posterior distribution $p(\theta|\mathcal{D})$

- Approach 2 also gives uncertainty about our estimate of θ; Approach 1 doesn't
  - But possible to estimate uncertainty in θ even with Approach 1 (e.g., using ensembles)
- Approach 1 is also a simplified/special case/approximation of Approach 2
- Can also take a hybrid (Approach 2 for some parameters; Approach 1 for others)

The Posterior Distribution

- The posterior distribution is computed using Bayes rule (Bayesian inference)

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

$$= \frac{p(\mathcal{D}|\theta)p(\theta|\alpha)}{\int p(\mathcal{D}|\theta)p(\theta|\alpha)d\theta} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}$$

Assuming α is known so the posterior is conditioned on α as well

Given θ, the data is conditionally independent of the prior's hyperparameters α so $p(\mathcal{D}|\theta,\alpha) = p(\mathcal{D}|\theta)$

- Marginal likelihood is an important quantity

$$p(\mathcal{D}|\alpha) = \int p(\mathcal{D}|\theta)p(\theta|\alpha)d\theta = \mathbb{E}_{p(\theta|\alpha)}p(\mathcal{D}|\theta)]$$

The average" likelihood (average taken w.r.t. all values of θ from the prior distribution)

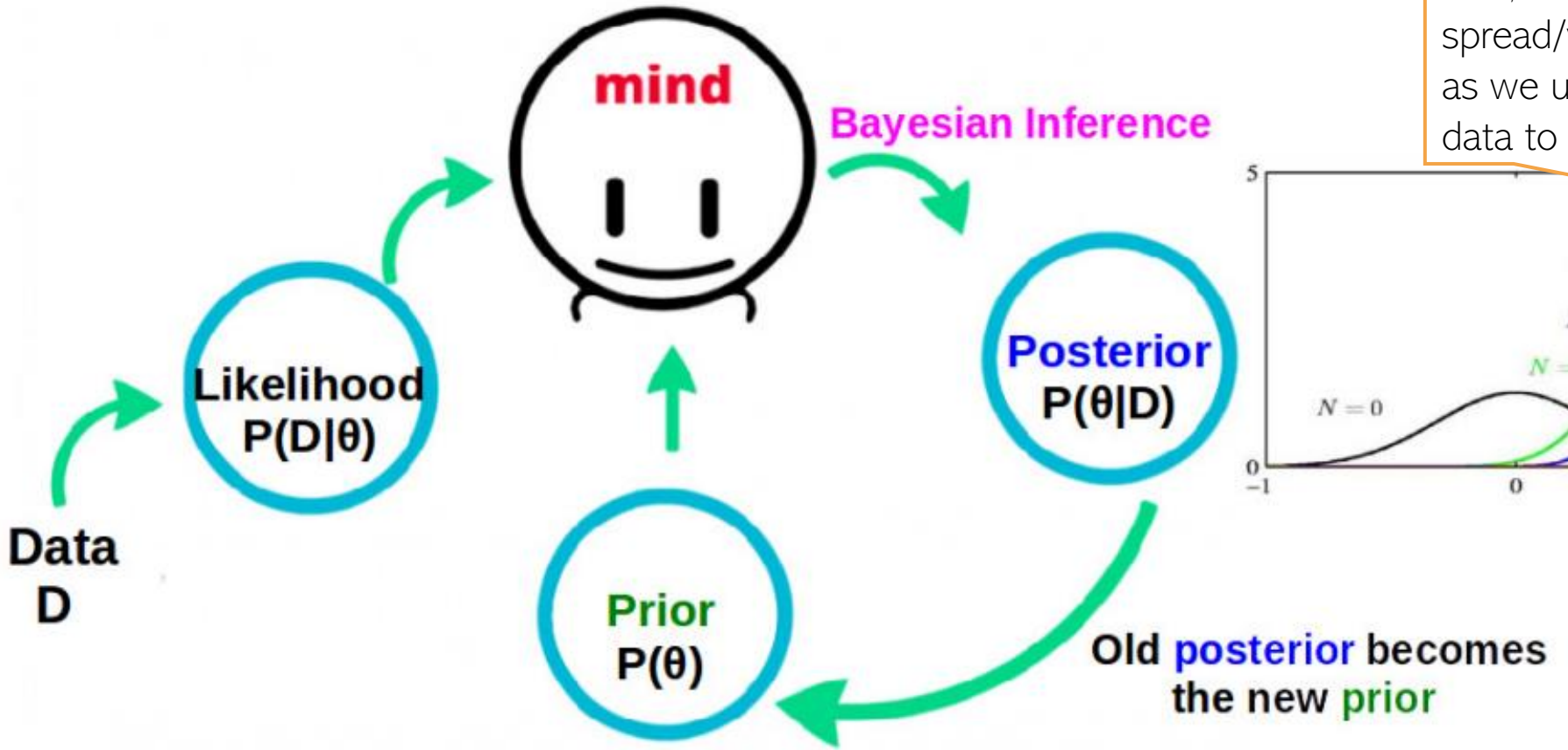
Hard to compute in general (that's why posterior is difficult to compute in general) but be computed exactly in some cases

We can use it also to find the best value of hyperparameters α

For example, $\hat{\alpha} = \text{argmax}_\alpha \log p(\mathcal{D}|\alpha)$

"Online" Nature of Bayesian Inference Updates

- Bayesian inference can naturally be done in an online fashion



Point Estimation

- Recall that the posterior is

$$p(\theta|\mathcal{D},\alpha) = \frac{p(\mathcal{D}|\theta)p(\theta|\alpha)}{p(\mathcal{D}|\alpha)}$$

Intractable mainly because the marginal likelihood (the denominator on the RHS is intractable in general)

Intractable to compute except for some very simple models or if the likelihood and prior are conjugate (discussed later) to each other

However, point estimation throws away all the uncertainty information about θ

- If posterior is intractable, can use MLE/MAP to get point estimates
  - Maximum likelihood (ML) estimation: Find θ for which likelihood is highest

Meaning the observed data has the largest probability for this value of θ

Negative Log likelihood (equivalent to a loss function)

$$\hat{\theta}_{ML} = \operatorname*{argmax}_{\theta} \log p(\mathcal{D}|\theta) = \operatorname*{argmin}_{\theta} -\log p(\mathcal{D}|\theta) = \operatorname*{argmin}_{\theta} NLL(\theta)$$

  - Maximum a posteriori (MAP) estimation: Find θ with largest posterior prob.

$$\hat{\theta}_{MAP} = \operatorname*{argmax}_{\theta} \log p(\theta|\mathcal{D},\alpha) = \operatorname*{argmax}_{\theta} [\log p(\mathcal{D}|\theta) + \log p(\theta|\alpha)]$$

$$= \operatorname*{argmin}_{\theta} [NLL(\theta) - \log p(\theta|\alpha)]$$

Like MLE with info from prior added

Akin to a regularizer added to the loss

The regularizer hyperparameter is part of prior

The Predictive Distribution

- Predictive distribution is the distribution of test data $\mathcal{D}_*$ given training data $\mathcal{D}$
- In the general form, we can write it as

$p(\mathcal{D}_*|\mathcal{D})$ is known as posterior predictive distribution (PPD)

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

In the posterior, not showing prior's hyperparameters α for brevity of notation

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

PPD is more robust (less chance of overfitting) than plug-in prediction because we aren't relying on a single best estimate

Assuming observations are i.i.d. given θ

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

The "averaged" prediction using all possible θ values with each prediction weighted by how important θ is as per the posterior distribution

This expectation may not be computable exactly and may itself require approximations (will see later)

An expectation over the posterior distribution (averaging over the posterior)

$$= \mathbb{E}_{p(\theta|\mathcal{D})}[p(\mathcal{D}_*|\theta)]$$

- If we only have point estimate of θ (say $\hat{\theta}$ obtained from MLE/MAP) then

This approximation of PPD is called "plug-in" predictive distribution

$$p(\mathcal{D}_*|\mathcal{D}) \approx p(\mathcal{D}_*|\hat{\theta})$$

Because now the posterior is just a point mass at $\hat{\theta}$

Predictive Distribution: An Example

- Consider a supervised discriminative model $p(\mathcal{D}|\boldsymbol{w}) = \prod_{i=1}^{N} p(y_i|\boldsymbol{x}_i, \boldsymbol{w})$
- Here the training data $\mathcal{D} = (\boldsymbol{x}_i, y_i)_{i=1}^{N}$ and $\boldsymbol{w}$ denotes the unknown parameters
- Consider two situations:
  - We have a posterior distribution $p(\boldsymbol{w}|\mathcal{D})$ over $\boldsymbol{w}$
  - We have a point estimate $\widehat{\boldsymbol{w}}$ (by minimizing a loss function on $\mathcal{D}$) of $\boldsymbol{w}$
- Suppose, as $\mathcal{D}_*$, we just have a single test input $\boldsymbol{x}_*$ and want to predict y_*
- The PPD $p(\mathcal{D}_*|\mathcal{D}) = \int p(\mathcal{D}_*|\boldsymbol{w}) p(\boldsymbol{w}|\mathcal{D})\, d\boldsymbol{w}$ in this case would be

$$p(y_*|\boldsymbol{x}_*, \mathcal{D}) = \int p(y_*|\boldsymbol{x}_*, \boldsymbol{w}) \times p(\boldsymbol{w}|\mathcal{D})\, d\theta$$

Instead of a single best value of param $\boldsymbol{w}$ for prediction, use all possible values of θ, weighted by their importance

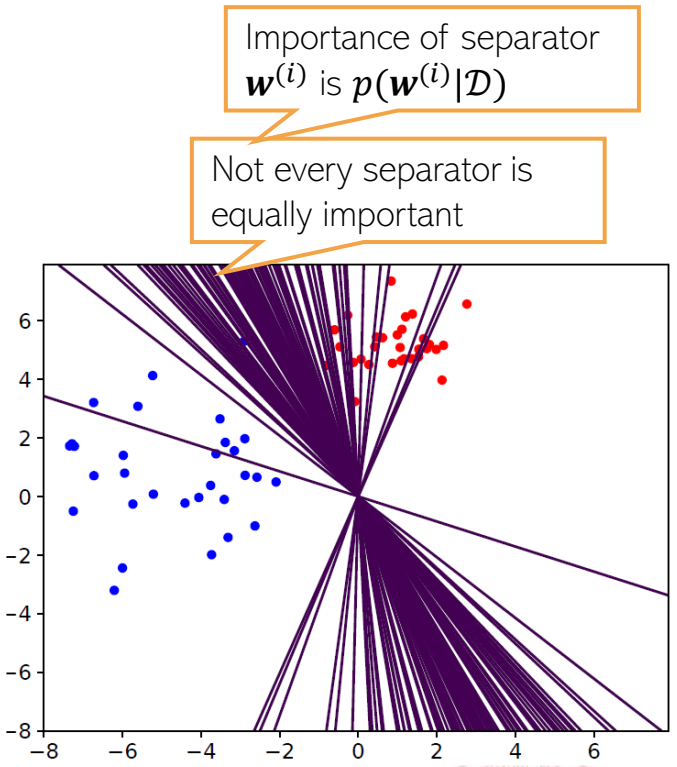
- On the other hand, the plug-in prediction uses the point estimate $\widehat{\boldsymbol{w}}$

$$p(y_*|\boldsymbol{x}_*, \mathcal{D}) \approx p(y_*|\boldsymbol{x}_*, \widehat{\boldsymbol{w}})$$

No averaging here; only the single best estimate of $\boldsymbol{w}$ used

A Look at Posterior Distribution over Parameters

- Consider a linear classification model for 2-dim inputs
- Classifier weight will be a 2-dim vector $\boldsymbol{w} = [w_1, w_2]$
- Its posterior will be some 2-dim distribution $p(\boldsymbol{w}|\mathcal{D})$
- Sampling from this distribution will generate 2-dim vectors
- Each vector will correspond to a linear separator (right fig)
- Thus posterior in this case is equivalent to a "collection" or "ensemble" of weights, each representing a different linear separator



A "Shortcut": PPD using Marginal Likelihood

- PPD, by definition, is obtained by the following marginalization

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

- Can also compute PPD without computing the posterior! Some ways:

1. Using a ratio of marginal likelihoods as follows

Follows simply from Bayes rule

$$p(a|b) = \frac{p(a,b)}{p(b)}$$

$$p(\mathcal{D}_*|\mathcal{D}) = \frac{p(\mathcal{D}_*, \mathcal{D})}{p(\mathcal{D})}$$

Joint marginal likelihood for training and test data

Marginal likelihood for training data

2. If $p(\mathcal{D}_*|\mathcal{D})$ can be obtained easily from the joint $p(\mathcal{D}_*, \mathcal{D})$
   - Note that the PPD $p(\mathcal{D}_*|\mathcal{D})$ is also a conditional distribution
   - For some distributions (e.g., Gaussian), conditionals can be easily derived from joint

Will see this being used we we study Gaussian Process (GP)