

# Introduction to Probabilistic Machine Learning

Piyush Rai

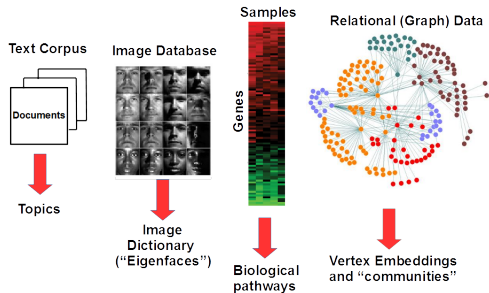
Dept. of CSE, IIT Kanpur

**(Mini-course 1)**

Nov 03, 2015

# Machine Learning

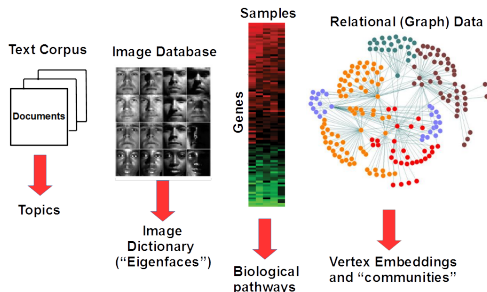
- Detecting trends/patterns in the data



- Making **predictions** about **future data**

# Machine Learning

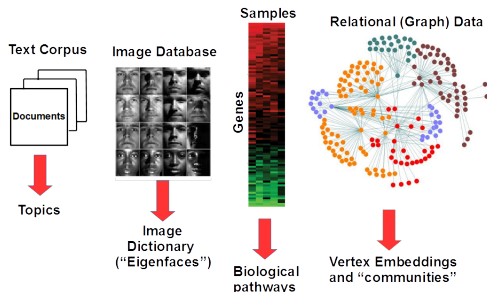
- Detecting trends/patterns in the data



- Making **predictions** about **future data**
- Two schools of thoughts
  - **Learning as optimization:** **fit** a model to minimize some **loss function**
  - **Learning as inference:** **infer** parameters of the **data generating distribution**

# Machine Learning

- Detecting trends/patterns in the data



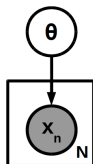
- Making **predictions** about **future data**
- Two schools of thoughts
  - **Learning as optimization:** **fit** a model to minimize some **loss function**
  - **Learning as inference:** **infer** parameters of the **data generating distribution**
- The two are not really completely disjoint ways of thinking about learning

# Plan for the mini-course

- A series of 4 talks
  - Introduction to Probabilistic and Bayesian Machine Learning (today)
  - Case Study: Bayesian Linear Regression, Approx. Bayesian Inference (Nov 5)
  - Nonparametric Bayesian modeling for [function approximation](#) (Nov 7)
  - Nonparam. Bayesian modeling for [clustering/dimensionality reduction](#) (Nov 8)

# Machine Learning via Probabilistic Modeling

- Assume data  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$  generated from a probabilistic model:

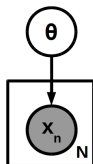


- Data usually assumed i.i.d. (independent and identically distributed)

$$\mathbf{x}_1, \dots, \mathbf{x}_N \sim p(\mathbf{x}|\theta)$$

# Machine Learning via Probabilistic Modeling

- Assume data  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$  generated from a probabilistic model:



- Data usually assumed i.i.d. (independent and identically distributed)

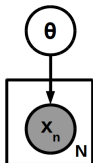
$$\mathbf{x}_1, \dots, \mathbf{x}_N \sim p(\mathbf{x}|\theta)$$

- For i.i.d. data, **probability of observed data**  $\mathbf{X}$  given **model parameters**  $\theta$

$$p(\mathbf{X}|\theta) = p(\mathbf{x}_1, \dots, \mathbf{x}_N|\theta) = \prod_{n=1}^N p(\mathbf{x}_n|\theta)$$

# Machine Learning via Probabilistic Modeling

- Assume data  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$  generated from a probabilistic model:



- Data usually assumed i.i.d. (independent and identically distributed)

$$\mathbf{x}_1, \dots, \mathbf{x}_N \sim p(\mathbf{x}|\theta)$$







- For i.i.d. data, **probability of observed data**  $\mathbf{X}$  given **model parameters**  $\theta$

$$p(\mathbf{X}|\theta) = p(\mathbf{x}_1, \dots, \mathbf{x}_N|\theta) = \prod_{n=1}^N p(\mathbf{x}_n|\theta)$$

- $p(\mathbf{x}_n|\theta)$  denotes the **likelihood** w.r.t. data point  $n$
- The form of  $p(\mathbf{x}_n|\theta)$  depends on the type/characteristics of the data

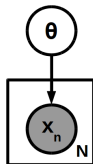


# Some common probability distributions

<u>Distribution</u>	<u>Domain</u>	<u>Picture</u>	<u>Parametric Form</u>
Binomial	Binary		$Bin(x   N, \theta) \propto \theta^n (1-\theta)^{N-n}$
Multinomial	K classes		$Mult(\bar{x}   \bar{\theta}) \propto \prod \theta_k^{x_k}$
Beta	$[0, 1]$		$Beta(\theta   \alpha, \beta) \propto \theta^{\alpha-1} (1-\theta)^{\beta-1}$
Gamma	$[0, \infty)$		$Gam(x   a, b) \propto x^{a-1} \exp(-bx)$
Dirichlet	Simplex		$Dir(\bar{\theta}   \bar{\alpha}) \propto \prod \theta_k^{\alpha_k-1}$
Gaussian	Reals		$Nor(\mathbf{x}   \mu, \sigma^2) \propto \exp\left(-\frac{(\mathbf{x}-\mu)^2}{2\sigma^2}\right)$

# Maximum Likelihood Estimation (MLE)

- We wish to estimate parameters  $\theta$  from observed data  $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$

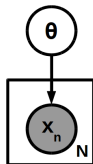


- MLE does this by finding  $\theta$  that maximizes the (log)likelihood  $p(\mathbf{X}|\theta)$

$$\hat{\theta} = \arg \max_{\theta} \log p(\mathbf{X}|\theta)$$

# Maximum Likelihood Estimation (MLE)

- We wish to estimate parameters  $\theta$  from observed data  $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$

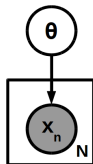


- MLE does this by finding  $\theta$  that maximizes the (log)likelihood  $p(\mathbf{X}|\theta)$

$$\hat{\theta} = \arg \max_{\theta} \log p(\mathbf{X}|\theta) = \arg \max_{\theta} \log \prod_{n=1}^N p(x_n|\theta)$$

# Maximum Likelihood Estimation (MLE)

- We wish to estimate parameters  $\theta$  from observed data  $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$

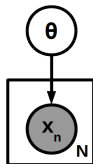


- MLE does this by finding  $\theta$  that maximizes the (log)likelihood  $p(\mathbf{X}|\theta)$

$$\hat{\theta} = \arg \max_{\theta} \log p(\mathbf{X}|\theta) = \arg \max_{\theta} \log \prod_{n=1}^N p(\mathbf{x}_n|\theta) = \arg \max_{\theta} \sum_{n=1}^N \log p(\mathbf{x}_n|\theta)$$

# Maximum Likelihood Estimation (MLE)

- We wish to estimate parameters  $\theta$  from observed data  $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$



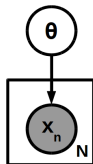
- MLE does this by finding  $\theta$  that maximizes the (log)likelihood  $p(\mathbf{X}|\theta)$

$$\hat{\theta} = \arg \max_{\theta} \log p(\mathbf{X}|\theta) = \arg \max_{\theta} \log \prod_{n=1}^N p(x_n|\theta) = \arg \max_{\theta} \sum_{n=1}^N \log p(x_n|\theta)$$

- MLE now reduces to solving an optimization problem w.r.t.  $\theta$

# Maximum Likelihood Estimation (MLE)

- We wish to estimate parameters  $\theta$  from observed data  $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$



- MLE does this by finding  $\theta$  that maximizes the (log)likelihood  $p(\mathbf{X}|\theta)$

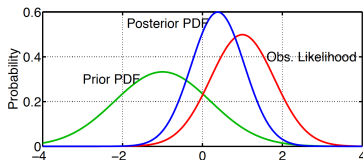
$$\hat{\theta} = \arg \max_{\theta} \log p(\mathbf{X}|\theta) = \arg \max_{\theta} \log \prod_{n=1}^N p(x_n|\theta) = \arg \max_{\theta} \sum_{n=1}^N \log p(x_n|\theta)$$

- MLE now reduces to solving an optimization problem w.r.t.  $\theta$
- MLE has some nice theoretical properties (e.g., consistency as  $N \rightarrow \infty$ )

# Injecting Prior Knowledge

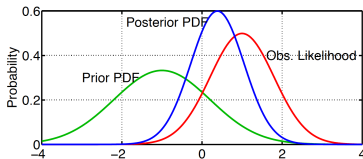
- Often, we might *a priori* know something about the parameters
- A **prior distribution**  $p(\theta)$  can encode/specify this knowledge
- **Bayes rule** gives us the **posterior distribution** over  $\theta$ :  $p(\theta|\mathbf{X})$
- Posterior reflects our updated knowledge about  $\theta$  using **observed data**

$$p(\theta|\mathbf{X}) = \frac{p(\mathbf{X}|\theta)p(\theta)}{p(\mathbf{X})} = \frac{p(\mathbf{X}|\theta)p(\theta)}{\int_{\theta} p(\mathbf{X}|\theta)p(\theta)d\theta} \propto \text{Likelihood} \times \text{Prior}$$



- Note:  $\theta$  is now a random variable

# Maximum-a-Posteriori (MAP) Estimation



- MAP estimation finds  $\theta$  that maximizes the posterior  $p(\theta|\mathbf{X}) \propto p(\mathbf{X}|\theta)p(\theta)$

$$\hat{\theta} = \arg \max_{\theta} \log \prod_{n=1}^N p(x_n|\theta)p(\theta) = \arg \max_{\theta} \sum_{n=1}^N \log p(x_n|\theta) + \log p(\theta)$$

- MAP now reduces to solving an optimization problem w.r.t.  $\theta$
- Objective function very similar to MLE, except for the  $\log p(\theta)$  term
- In some sense, MAP is just a “regularized” MLE



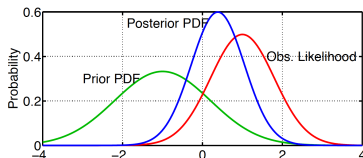
# Bayesian Learning

- Both MLE and MAP only give a **point estimate** (single best answer) of  $\theta$
- How can we capture/quantify the uncertainty in  $\theta$ ?

# Bayesian Learning

- Both MLE and MAP only give a **point estimate** (single best answer) of  $\theta$
- How can we capture/quantify the uncertainty in  $\theta$ ?
- Need to infer the **full posterior distribution**

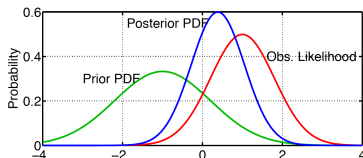
$$p(\theta|\mathbf{X}) = \frac{p(\mathbf{X}|\theta)p(\theta)}{p(\mathbf{X})} = \frac{p(\mathbf{X}|\theta)p(\theta)}{\int_{\theta} p(\mathbf{X}|\theta)p(\theta)d\theta} \propto \text{Likelihood} \times \text{Prior}$$



# Bayesian Learning

- Both MLE and MAP only give a **point estimate** (single best answer) of  $\theta$
- How can we capture/quantify the uncertainty in  $\theta$ ?
- Need to infer the **full posterior distribution**

$$p(\theta|\mathbf{X}) = \frac{p(\mathbf{X}|\theta)p(\theta)}{p(\mathbf{X})} = \frac{p(\mathbf{X}|\theta)p(\theta)}{\int_{\theta} p(\mathbf{X}|\theta)p(\theta)d\theta} \propto \text{Likelihood} \times \text{Prior}$$



- Requires doing a “fully Bayesian” inference
- Inference sometimes a somewhat easy and sometimes a (very) hard problem

# A Simple Example of Bayesian Inference

- We want to estimate a coin's bias  $\theta \in (0, 1)$  based on  $N$  tosses
- The likelihood model:  $\{\mathbf{x}_1, \dots, \mathbf{x}_N\} \sim \text{Bernoulli}(\theta)$

$$p(\mathbf{x}_n|\theta) = \theta^{x_n}(1 - \theta)^{1-x_n}$$

- The prior:  $\theta \sim \text{Beta}(a, b)$

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

- The posterior  $p(\theta|\mathbf{X}) \propto \prod_{n=1}^N p(\mathbf{x}_n|\theta)p(\theta|a, b)$   
 $\propto \prod_{n=1}^N \theta^{x_n}(1-\theta)^{1-x_n} \theta^{a-1}(1-\theta)^{b-1}$   
 $= \theta^{a+\sum_{n=1}^N x_n-1}(1-\theta)^{b+N-\sum_{n=1}^N x_n-1}$
- Thus the posterior is:  $\text{Beta}(a + \sum_{n=1}^N x_n, b + N - \sum_{n=1}^N x_n)$
- Here, the posterior has the same form as the prior (both Beta)
- Also very easy to perform **online inference**

# Conjugate Priors

- Recall  $p(\theta|\mathbf{X}) = \frac{p(\mathbf{X}|\theta)p(\theta)}{p(\mathbf{X})}$
- Given some data distribution (likelihood)  $p(\mathbf{X}|\theta)$  and a prior  $p(\theta) = \pi(\theta|\alpha)$ ..
- The prior is conjugate if the posterior also has the same form, i.e.,

$$p(\theta|\alpha, \mathbf{X}) = \frac{P(\mathbf{X}|\theta)\pi(\theta|\pi)}{p(\mathbf{X})} = \pi(\theta|\alpha_*)$$

- Several pairs of distributions are conjugate to each other, e.g.,
  - Gaussian-Gaussian
  - Beta-Bernoulli
  - Beta-Binomial
  - Gamma-Poisson
  - Dirichlet-Multinomial
  - ..

# A Non-Conjugate Case

- Want to learn a classifier  $\theta$  for predicting label  $x \in \{-1, +1\}$  for a point  $\mathbf{z}$
- Assume a **logistic likelihood** model for the labels

$$p(x_n|\theta) = \frac{1}{1 + \exp(-\mathbf{x}_n\theta^\top \mathbf{z}_n)}$$

# A Non-Conjugate Case

- Want to learn a classifier  $\theta$  for predicting label  $x \in \{-1, +1\}$  for a point  $\mathbf{z}$
- Assume a **logistic likelihood** model for the labels

$$p(\mathbf{x}_n|\theta) = \frac{1}{1 + \exp(-\mathbf{x}_n\theta^\top \mathbf{z}_n)}$$

- The **prior**:  $\theta \sim \text{Normal}(\mu, \Sigma)$  (Gaussian, not conjugate to the logistic)

$$p(\theta|\mu, \Sigma) \propto \exp\left(-\frac{1}{2}(\theta - \mu)^\top \Sigma^{-1}(\theta - \mu)\right)$$

# A Non-Conjugate Case

- Want to learn a classifier  $\theta$  for predicting label  $x \in \{-1, +1\}$  for a point  $\mathbf{z}$
- Assume a **logistic likelihood** model for the labels

$$p(\mathbf{x}_n|\theta) = \frac{1}{1 + \exp(-\mathbf{x}_n\theta^\top \mathbf{z}_n)}$$

- The **prior**:  $\theta \sim \text{Normal}(\mu, \Sigma)$  (Gaussian, not conjugate to the logistic)

$$p(\theta|\mu, \Sigma) \propto \exp\left(-\frac{1}{2}(\theta - \mu)^\top \Sigma^{-1}(\theta - \mu)\right)$$

- The **posterior**  $p(\theta|\mathbf{X}) \propto \prod_{n=1}^N p(\mathbf{x}_n|\theta)p(\theta|\mu, \Sigma)$  does not have a closed form



# A Non-Conjugate Case

- Want to learn a classifier  $\theta$  for predicting label  $x \in \{-1, +1\}$  for a point  $\mathbf{z}$
- Assume a **logistic likelihood** model for the labels

$$p(\mathbf{x}_n|\theta) = \frac{1}{1 + \exp(-\mathbf{x}_n\theta^\top \mathbf{z}_n)}$$

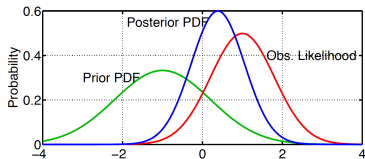
- The **prior**:  $\theta \sim \text{Normal}(\mu, \Sigma)$  (Gaussian, not conjugate to the logistic)

$$p(\theta|\mu, \Sigma) \propto \exp\left(-\frac{1}{2}(\theta - \mu)^\top \Sigma^{-1}(\theta - \mu)\right)$$

- The **posterior**  $p(\theta|\mathbf{X}) \propto \prod_{n=1}^N p(\mathbf{x}_n|\theta)p(\theta|\mu, \Sigma)$  does not have a closed form
- Approximate Bayesian inference needed in such cases
  - Sampling based approximations: MCMC methods
  - Optimization based approximations: Variational Bayes, Laplace, etc.

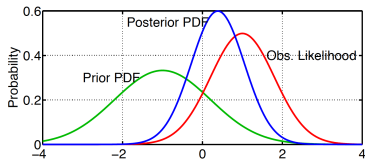
# Benefits of Bayesian Modeling

- Our estimate of  $\theta$  is not a single value (“point”) but a distribution
- Can model and quantify the uncertainty (or “variance”) in  $\theta$  via  $p(\theta|\mathbf{X})$



# Benefits of Bayesian Modeling

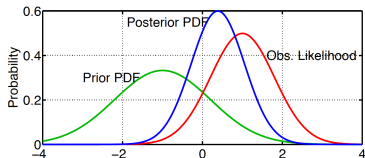
- Our estimate of  $\theta$  is not a single value (“point”) but a distribution
- Can model and quantify the uncertainty (or “variance”) in  $\theta$  via  $p(\theta|\mathbf{X})$



- Can use the uncertainty in various tasks such as diagnosis, predictions

# Benefits of Bayesian Modeling

- Our estimate of  $\theta$  is not a single value (“point”) but a distribution
- Can model and quantify the uncertainty (or “variance”) in  $\theta$  via  $p(\theta|\mathbf{X})$



- Can use the uncertainty in various tasks such as diagnosis, predictions , e.g.,
- Making predictions by averaging over all possible values of  $\theta$

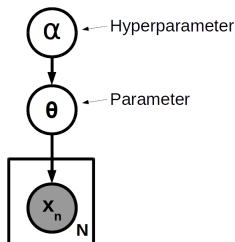
$$p(y|x, X, Y) = \mathbb{E}_{p(\theta|.)}[p(y|x, \theta)]$$

$$p(y|x, X, Y) = \int p(y|x, \theta)p(\theta|X, Y)d\theta$$

- Allows also quantifying the **uncertainty in the predictions**

# Other Benefits of Bayesian Modeling

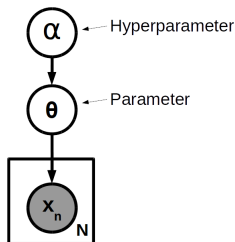
- Hierarchical model construction: parameters can depend on **hyperparameters**



- hyperparameters need not be tuned but can be inferred from data
  - .. by maximizing the **marginal likelihood**  $p(\mathbf{X}|\alpha)$

# Other Benefits of Bayesian Modeling

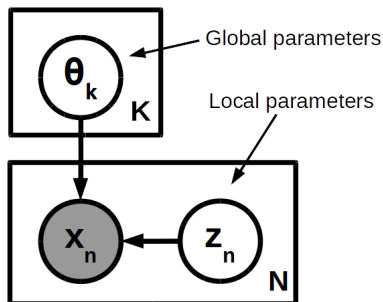
- Hierarchical model construction: parameters can depend on **hyperparameters**



- hyperparameters need not be tuned but can be inferred from data
  - .. by maximizing the **marginal likelihood**  $p(\mathbf{X}|\alpha)$
- Provides robustness: E.g., learning the **sparsity hyperparameter** in sparse regression, learning **kernel hyperparameters** in kernel methods

# Other Benefits of Probabilistic/Bayesian Modeling

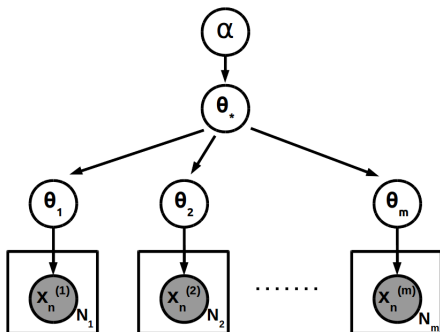
- Can introduce “local parameters” (latent variables) associated with each data point and infer those as well



- Used in many problems: Gaussian mixture model, probabilistic principal component analysis, factor analysis, topic models

# Other Benefits of Probabilistic/Bayesian Modeling

- Enables a modular architecture: Simple models can be neatly combined to solve more complex problems

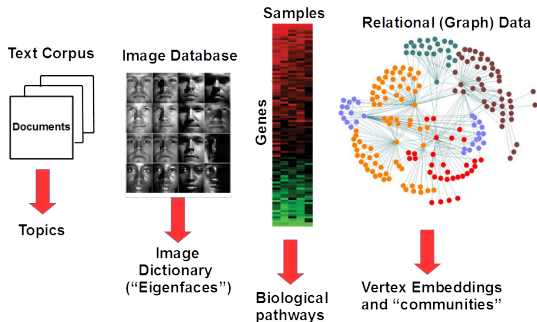


- Allows jointly learning across multiple data sets (sometimes also known as multitask learning or transfer learning)



# Other Benefits of Bayesian Modeling

- **Nonparametric Bayesian modeling:** a principled way to learn **model size**
- E.g., how many clusters (Gaussian mixture model or graph clustering), how many basis vectors (PCA) or dictionary elements (sparse coding or dictionary learning), how many topics (topic models such as LDA), etc..



- NPBayes modeling allows the model size to grow with data

# Other Benefits of Bayesian Modeling

- Sequential data acquisition or “active learning”
- Can check how confident the learned model is w.r.t. a new data point

$p(\theta \lambda)$	=	Normal( $\theta 0, \lambda^2$ )	Prior
$p(y \mathbf{x}, \theta)$	=	Normal( $y \theta^\top \mathbf{x}, \sigma^2$ )	Likelihood
$p(\theta Y, \mathbf{X})$	=	Normal( $\theta \mu_\theta, \Sigma_\theta$ )	Posterior
$p(y_0 \mathbf{x}_0, Y, \mathbf{X})$	=	Normal( $y_0 \mu_0, \sigma_0^2$ )	Predictive dist.
$\mu_0$	=	$\mu_\theta^\top \mathbf{x}_0$	Predictive mean
$\sigma_0^2$	=	$\sigma^2 + \mathbf{x}_0^\top \Sigma_\theta \mathbf{x}_0$	Predictive variance

# Other Benefits of Bayesian Modeling

- Sequential data acquisition or “active learning”
- Can check how confident the learned model is w.r.t. a new data point

$p(\theta \lambda)$	=	Normal( $\theta 0, \lambda^2$ )	Prior
$p(y \mathbf{x}, \theta)$	=	Normal( $y \theta^\top \mathbf{x}, \sigma^2$ )	Likelihood
$p(\theta Y, \mathbf{X})$	=	Normal( $\theta \mu_\theta, \Sigma_\theta$ )	Posterior
$p(y_0 \mathbf{x}_0, Y, \mathbf{X})$	=	Normal( $y_0 \mu_0, \sigma_0^2$ )	Predictive dist.
$\mu_0$	=	$\mu_\theta^\top \mathbf{x}_0$	Predictive mean
$\sigma_0^2$	=	$\sigma^2 + \mathbf{x}_0^\top \Sigma_\theta \mathbf{x}_0$	Predictive variance

- Gives a strategy to choose data points sequentially for improved learning with a budget on the amount of data available

# Next Talk

- Case study on Bayesian sparse linear regression
- Hyperparameter estimation
- Introduction to approximate Bayesian inference

Thanks! Questions?