Introduction to Machine Learning (CSCI-UA 473): Fall 2021

Lecture 5: Bayesian Decision Theory and Bayesian Parameter Estimation

Sumit Chopra
Courant Institute of Mathematical Sciences
Department of Radiology - Grossman School of Medicine
NYU

Slides derived from materials from Benjamin Peherstorfer, Kyunghyun Cho, Andrew Gordon Wilson

Lecture Outline

Validation (from previous lecture)

Bayesian Decision Theory

Frequentist vs Bayesian Approach

Bayesian Parameter Estimation

Maximum A Posteriori Estimation

Bayesian Linear Regression

Generative vs Discriminative Models

Naive Bayes Classifiers

Validation

For any hypothesis h

$E_{out}(h) = E_{in}(h)$ + Penalty for overfitting

Validation

For any hypothesis h

$E_{out}(h) = E_{in}(h)$ + Penalty for overfitting

Regularization estimates this quantity

Validation

For any hypothesis h

$$E_{out}(h) = E_{in}(h) + \text{Penalty for overfitting}$$

Validation cuts to the chase and tries to directly estimate this

Regularization estimates this quantity

Validation

We briefly discussed carving out a validation set from your training set to estimate out-of-sample error

Can we say something more formally?

For any hypothesis h

$$E_{out}(h) = E_{in}(h) + \text{Penalty for overfitting}$$

Validation cuts to the chase and tries to directly estimate this

Regularization estimates this quantity

Validation

Dataset $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$

Training set $D_{train} \in D \leftarrow N - K$ samples

Validation set $D_{val} \in D \leftarrow K$ samples

g^-: the hypothesis selected after training the model on D_{train}

$$E_{val}(g^-) = \frac{1}{K} \sum_{x_n \in D_{val}} e\left(g^-(x_n), y_n\right)$$

How do we know whether $E_{val}(g^-)$ is a reasonable estimate of $E_{out}(g^-)$?

Validation

$$\mathbb{E}_{D_{val}}\left[E_{val}(g^-)\right] = \mathbb{E}_{D_{val}}\left[\frac{1}{K}\sum_{x_n \in D_{val}} e\left(g^-(x_n), y_n\right)\right]$$

$$= \frac{1}{K}\sum_{x_n \in D_{val}} \mathbb{E}_{D_{val}}\left[e\left(g^-(x_n), y_n\right)\right]$$

$$= \frac{1}{K}\sum_{x_n \in D_{val}} E_{out}(g^-)$$

$$= E_{out}(g^-)$$

This is because

$$\mathbb{E}_{D_{val}}\left[e\left(g^-(x_n), y_n\right)\right] = E_{x_n}\left[e\left(g^-(x_n), y_n\right)\right] = E_{out}(g^-)$$

Validation

Thus we have

$$\mathbb{E}\left[E_{val}(g^-)\right] = \frac{1}{K}\sum_{x_n \in D_{val}} \mathbb{E}\left[e\left(g^-(x_n), y_n\right)\right] = E_{out}(g^-)$$

$$\mathbb{V}\left[E_{val}(g^-)\right] = \frac{1}{K^2}\sum_{x_n \in D_{val}} \mathbb{V}\left[e\left(g^-(x_n), y_n\right)\right] = \frac{\sigma^2}{K}$$

$$E_{out}(g^-) \leq E_{val}(g^-) + O\left(\frac{1}{\sqrt{K}}\right)$$

Small K will lead to a poor estimate of the error

But what about large K ?

Validation

Note that the validation set is carved out of the full data set

Large K means small $N - K$ (size of D_{train})

So your chosen hypothesis g^- is poor and hence the estimates will be completely off

Figure from the book Learning from Data

Validation

Note that the validation set is carved out of the full data set

Large K means small $N - K$ (size of D_{train})

So your chosen hypothesis g^- is poor and hence the estimates will be completely off

Practical rule of thumb: use 20% of D as D_{val}

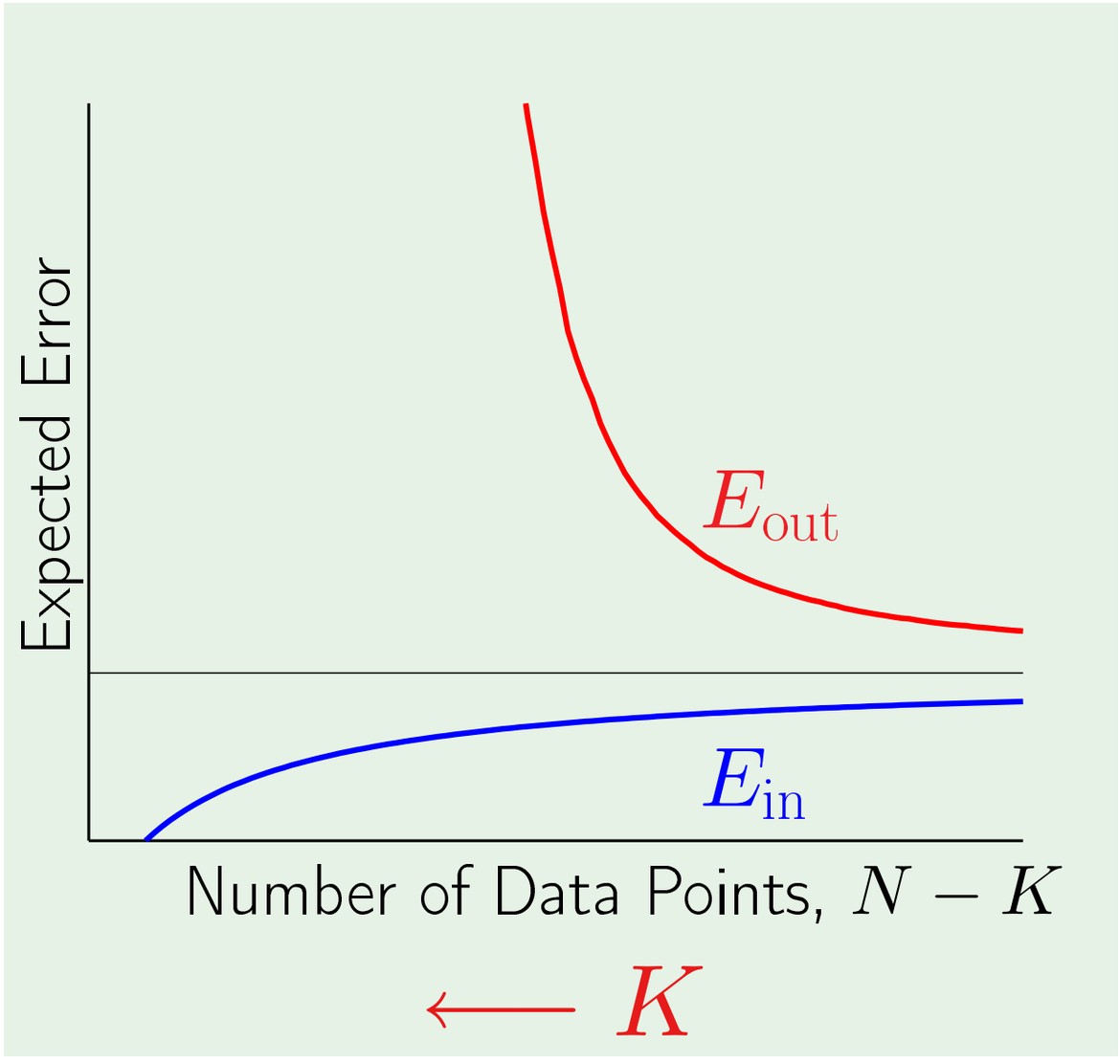


Figure from the book Learning from Data

Validation: Fold Back In

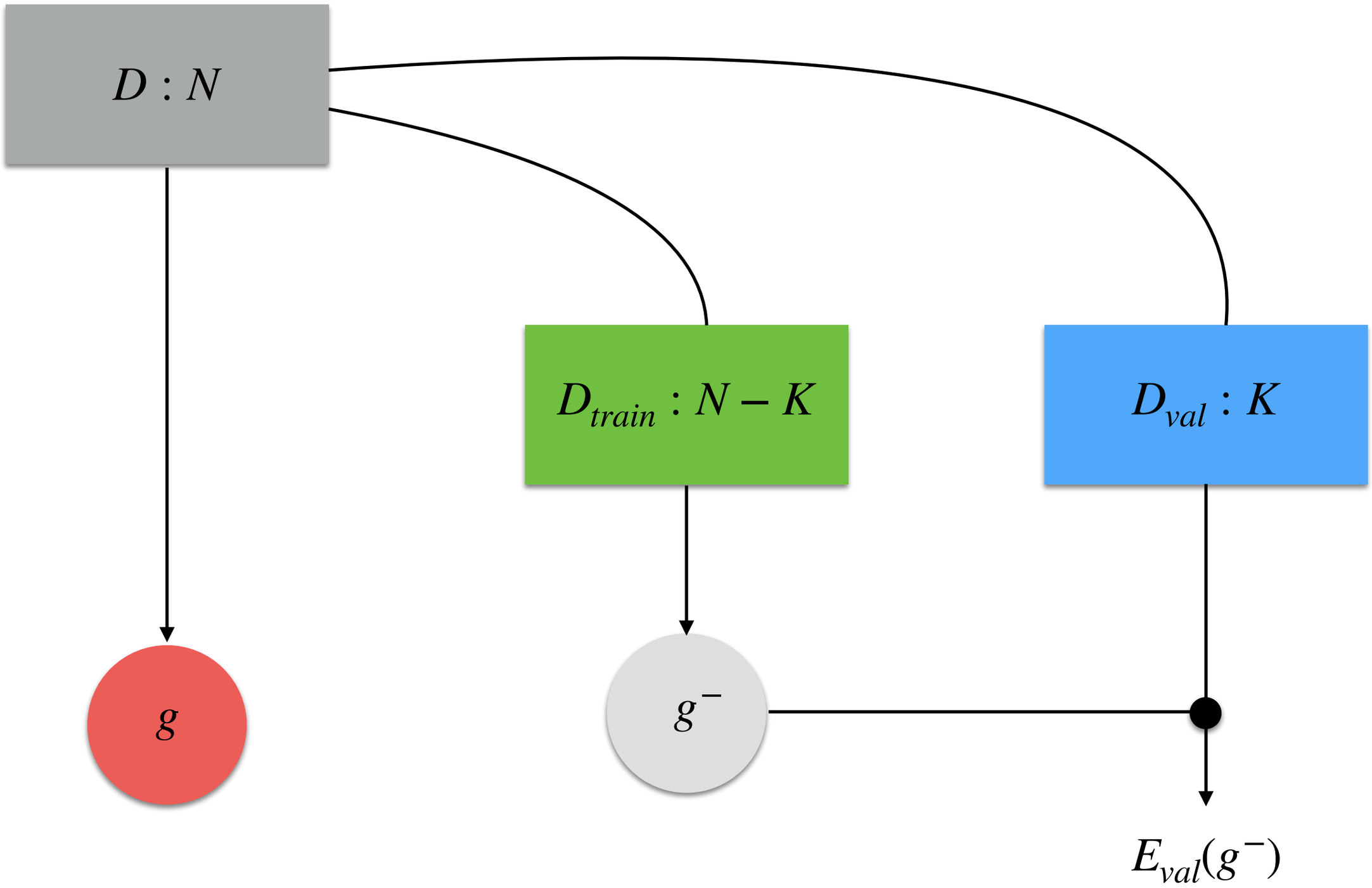


Figure adapted from the book Learning from Data

Model Selection

Turns out that you can use the same validation set multiple times without loosing guarantees

Assume you have M models (hypothesis sets): $\{\mathscr{H}_1, \mathscr{H}_2, \ldots, \mathscr{H}_M\}$

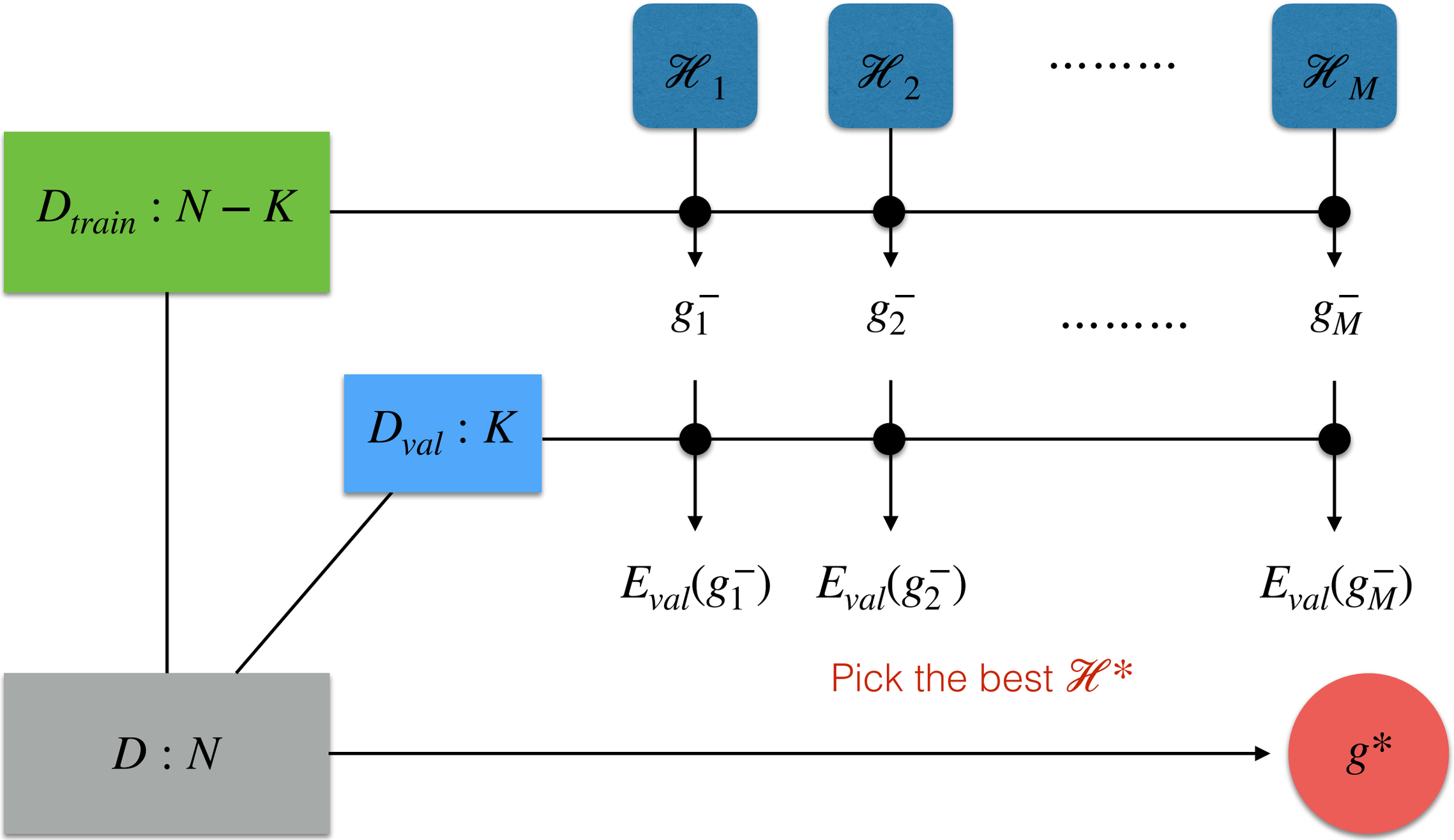


Figure adapted from the book Learning from Data

Cross Validation

$$E_{out}(g) \approx E_{out}(g^{-})$$

Small K

Cross Validation

Large K

$$E_{out}(g) \approx E_{out}(g^-) \approx E_{val}(g^-)$$

Small K

Cross Validation

Large K

$$E_{out}(g) \approx E_{out}(g^-) \approx E_{val}(g^-)$$

Small K

Leave One Out Analysis

$$D = (x_1, y_1), \ldots (x_{n-1}, y_{n-1}), (x_n, y_n), (x_{n+1}, y_{n+1}), \ldots, (x_N, y_N)$$

Cross Validation

Large K

$$E_{out}(g) \approx E_{out}(g^-) \approx E_{val}(g^-)$$

Small K

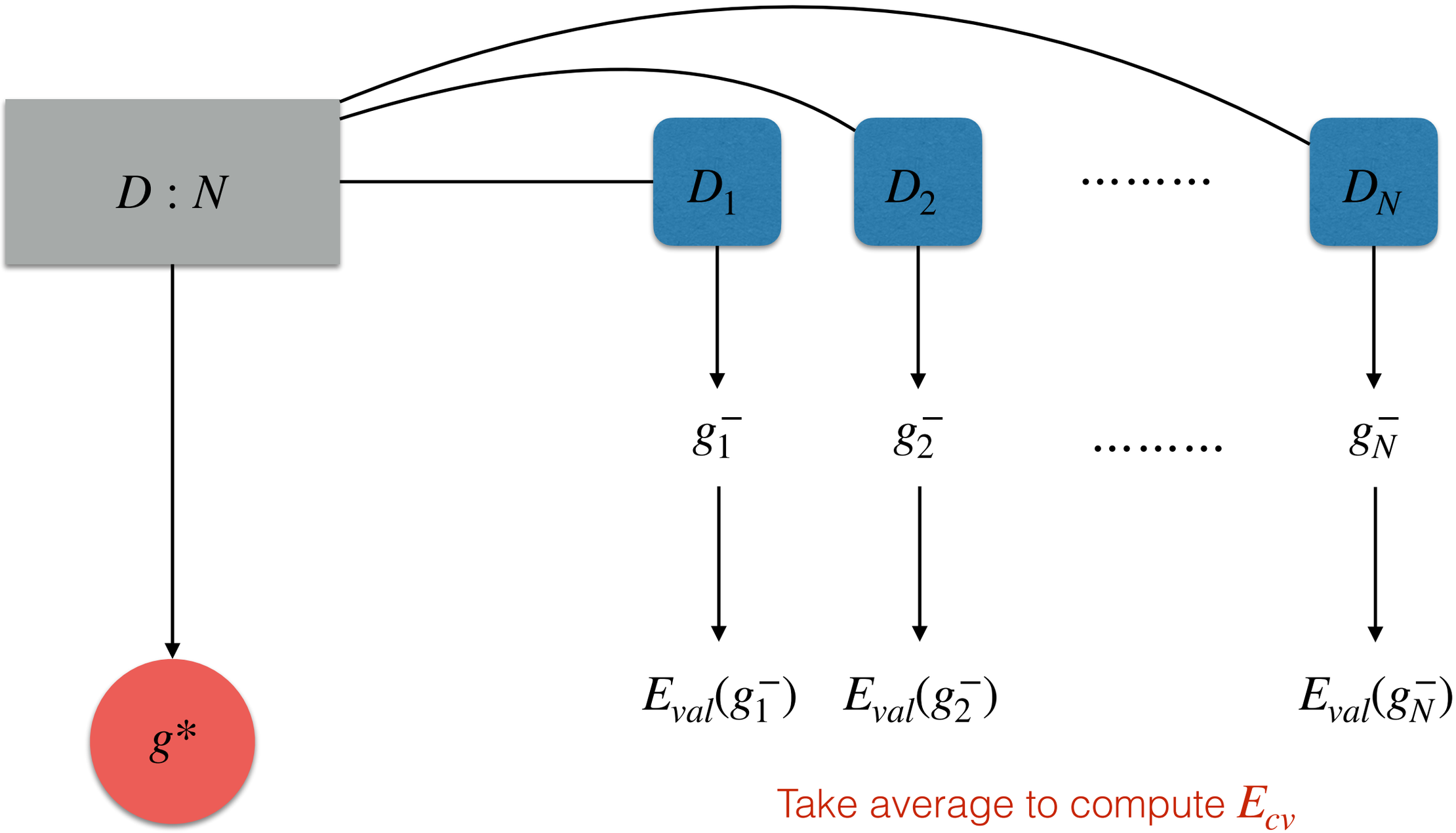
Leave One Out Analysis

$$D = (x_1, y_1), \ldots (x_{n-1}, y_{n-1}), (x_n, y_n), (x_{n+1}, y_{n+1}), \ldots, (x_N, y_N)$$

$$E_{val}(g^-) = e_n = e(g_n^-(x_n), y_n)$$

$$E_{cv} = \frac{1}{N} \sum_{n=1}^{N} e_n$$

Leave One Out Analysis

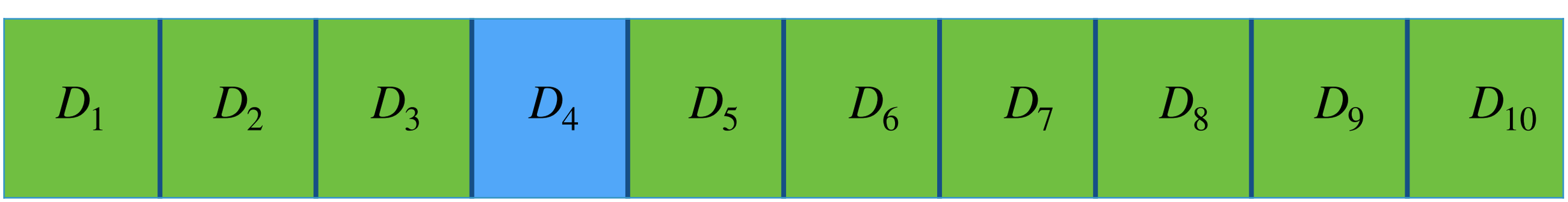


Take average to compute E_{cv}

Figure adapted from the book Learning from Data

K-fold Cross Validation

Train Validation Train



Model Selection with Cross Validation

Define M models by choosing different values of λ: $(\mathscr{H}, \lambda_1), (\mathscr{H}, \lambda_2) \ldots, (\mathscr{H}, \lambda_M)$

For each model $m = 1,2,\ldots,M$ do

Run the cross validation module to get an estimate of the cross validation error $E_{cv}(g^m)$

Pick the model $(\mathscr{H}, \lambda^*)$ with the smallest error

Train the model $(\mathscr{H}, \lambda^*)$ on the entire training set D to obtain the final hypothesis g^{m*}

Bayesian Decision Theory

Recap: Bayes Theorem

Let X and Y be two random variables

There are two ways to write the joint distribution $P(X, Y)$

$$P(X, Y) = P(X \mid Y)P(Y)$$

Or

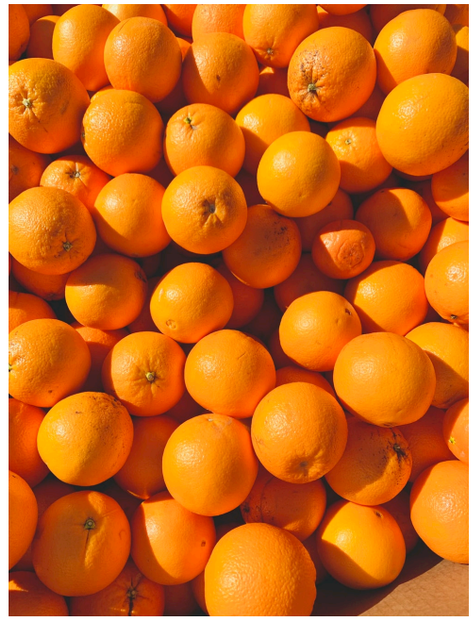
$$P(X, Y) = P(Y \mid X)P(X)$$

Equating the two right hand sides give

$$P(Y \mid X) = \frac{P(X \mid Y)P(Y)}{P(X)}$$

Bayesian Decision Theory

Suppose you want to build a conveyor belt with a robot arm and a camera, that can automatically sort apples from oranges and put them in different piles



Train a model which takes as input an image and tells whether its an apple or an orange

Logistic regression is one possible way to solve this

Collect 100,000 images of apples and 100,000 images of oranges: training set

Use the training images to **train parameters of the logistic regression** to maximize the likelihood of the training data

Use this trained function on the images captured by the camera and guide the robot arm

Bayesian Decision Theory

Suppose you want to build a conveyor belt with a robot arm and a camera, that can automatically sort apples from oranges and put them in different piles

Can we formalize the decision making process?

Train a model which takes as input an image and tells whether its an apple or an orange

Logistic regression is one possible way to solve this

Collect 100,000 images of apples and 100,000 images of oranges: training set

Use the training images to **train parameters of the logistic regression** to maximize the likelihood of the training data

Use this trained function on the images captured by the camera and guide the robot arm

Bayesian Decision Theory

Let ω denote the state of nature

ω_1: apples in the wild

ω_2: oranges in the wild

$P(\omega_1)$ and $P(\omega_2)$ are the **prior probabilities** of apples and oranges

Our belief of the existence of apples and oranges without looking at the data (prior knowledge)

Bayesian Decision Theory

Let ω denote the state of nature

ω_1: apples in the wild

ω_2: oranges in the wild

$P(\omega_1)$ and $P(\omega_2)$ are the **prior probabilities** of apples and oranges

Our belief of the existence of apples and oranges without looking at the data (prior knowledge)

Given an image we observe some measurement (feature) of the object within it

E.g., size of the object and we denote it by x

$P(x \mid \omega_1)$ and $P(x \mid \omega_2)$ are the **class conditional probability density** functions

The distribution of the size of the object given that the object is an apple or an orange

Bayesian Decision Theory

Assume we know $P(\omega_1)$, $P(\omega_2)$ and $P(x \mid \omega_1)$ and $P(x \mid \omega_2)$ for the two classes

Given a new image we observe x

Can we say anything about what object does the image contain?

Bayesian Decision Theory

Assume we know $P(\omega_1)$, $P(\omega_2)$ and $P(x \mid \omega_1)$ and $P(x \mid \omega_2)$ for the two classes

Given a new image we observe x

Can we say anything about what object does the image contain?

We know that the joint probability can be written as

$$P(\omega_j, x) = P(x \mid \omega_j)P(\omega_j) = P(\omega_j \mid x)P(x)$$

Bayes Rule: $$P(\omega_j \mid x) = \frac{P(x \mid \omega_j)P(\omega_j)}{P(x)}$$

Where in this case of two classes we have $P(x) = \sum_{i=1}^{2} P(x \mid \omega_i)P(\omega_i)$

Bayesian Decision Theory

Assume we know $P(\omega_1)$, $P(\omega_2)$ and $P(x \mid \omega_1)$ and $P(x \mid \omega_2)$ for the two classes

Given a new image we observe x

Can we say anything about what object does the image contain?

We know that the joint probability can be written as

$$P(\omega_j, x) = P(x \mid \omega_j)P(\omega_j) = P(\omega_j \mid x)P(x)$$

Bayes Rule:

$$P(\omega_j \mid x) = \frac{P(x \mid \omega_j)P(\omega_j)}{P(x)}$$

Likelihood, Prior, Posterior, Evidence

Where in this case of two classes we have $P(x) = \sum_{i=1}^{2} P(x \mid \omega_i)P(\omega_i)$

Bayesian Decision Theory

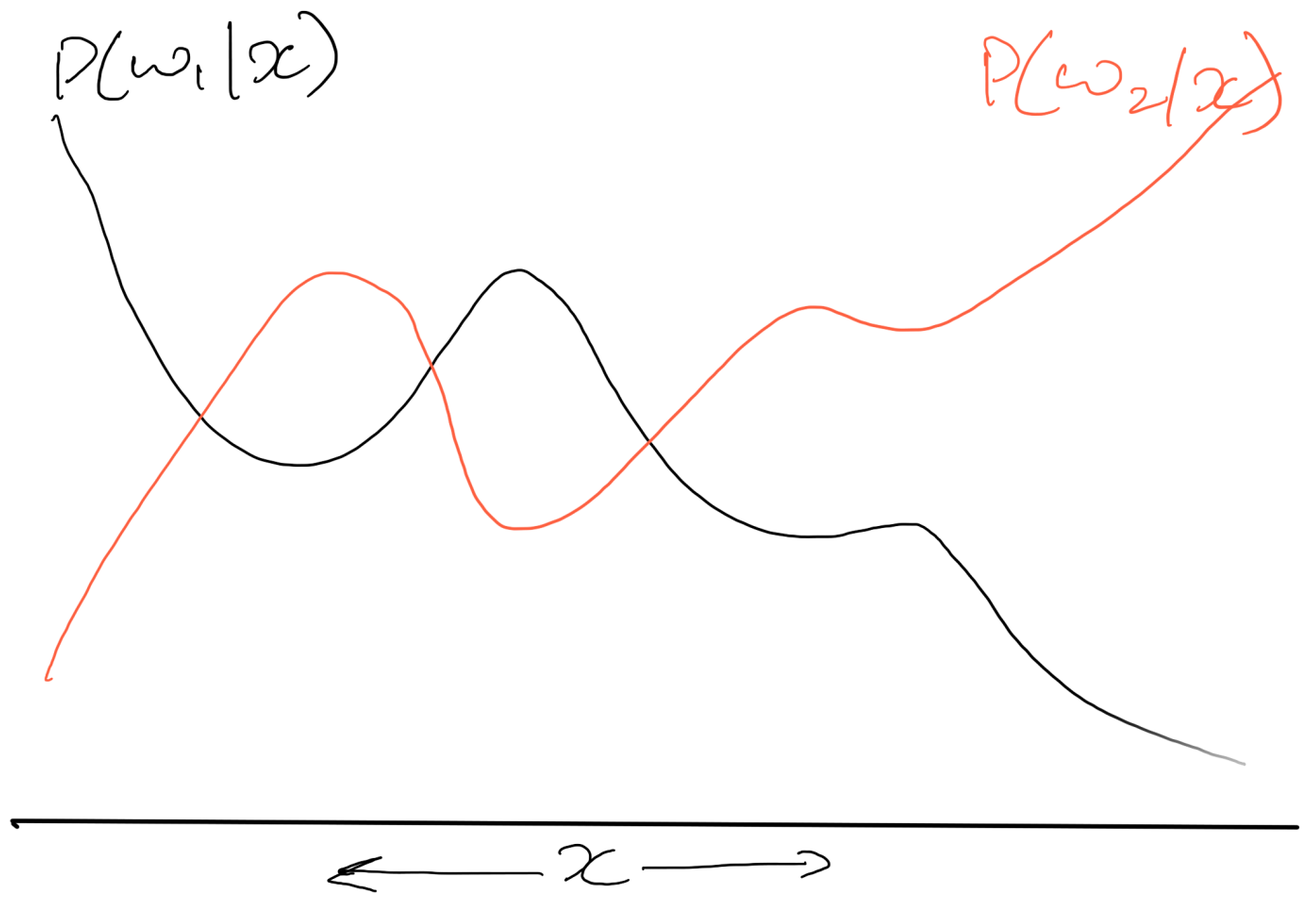
$P(error|x) = P(\omega_1|x)$ if we decide on ω_2

$P(error|x) = P(\omega_2|x)$ if we decide on ω_1

Thus to minimize $P(error|x)$ we have the following rule $P(error|x) = min\left[P(\omega_1|x), P(\omega_2, x)\right]$

Also since evidence $P(x)$ is a scaling factor we have the following decision rule

Pick ω_1 if $P(x|\omega_1)P(\omega_1) > P(x|\omega_2)P(\omega_2)$ otherwise pick ω_2



Bayesian Decision Theory

$\mathbf{x}$: multi-dimensional observations represented as a vector

$\{\omega_1, \omega_2, \ldots, \omega_c\}$: set of finite classes ("states of nature")

$\{\alpha_1, \alpha_2, \ldots, \alpha_a\}$: finite set of actions one can take

$\lambda(\alpha_i \mid \omega_j) = \lambda_{ij}$: loss associated with taking an action α_i when the state of nature is ω_j

$P(error \mid x) = P(\omega_2 \mid x)$ if we decide on ω_1

Bayes Theorem: $$P(\omega_j \mid \mathbf{x}) = \frac{P(\mathbf{x} \mid \omega_j)P(\omega_j)}{P(\mathbf{x})}$$

Evidence: $$P(\mathbf{x}) = \sum_{j=1}^{c} P(\mathbf{x} \mid \omega_j)P(\omega_j)$$

Expected loss associated with taking an action α_i when the true state of nature is ω_j

$$R(\alpha_i \mid \mathbf{x}) = \sum_{j=1}^{c} \lambda(\alpha_i \mid \omega_j)P(\omega_j \mid \mathbf{x})$$

Bayesian Decision Theory

$\mathbf{x}$: multi-dimensional observations represented as a vector

$\{\omega_1, \omega_2, \ldots, \omega_c\}$: set of finite classes ("states of nature")

$\{\alpha_1, \alpha_2, \ldots, \alpha_a\}$: finite set of actions one can take

$\lambda(\alpha_i \mid \omega_j) = \lambda_{ij}$: loss associated with taking an action α_i when the state of nature is ω_j

$P(error \mid x) = P(\omega_2 \mid x)$ if we decide on ω_1

Bayes Theorem: $$P(\omega_j \mid \mathbf{x}) = \frac{P(\mathbf{x} \mid \omega_j)P(\omega_j)}{P(\mathbf{x})}$$

Evidence: $$P(\mathbf{x}) = \sum_{j=1}^{c} P(\mathbf{x} \mid \omega_j)P(\omega_j)$$

Expected loss associated with taking an action α_i when the true state of nature is ω_j

$$R(\alpha_i \mid \mathbf{x}) = \sum_{j=1}^{c} \lambda(\alpha_i \mid \omega_j)P(\omega_j \mid \mathbf{x})$$ ← Conditional Risk

Bayes Decision Rule: in order to minimize the overall risk, compute the conditional risk $R(\alpha_i \mid \mathbf{x})$ for a $i = 1,2,\ldots,a$ and pick the action a_i for which the conditional risk is minimum

Lecture Outline

~~Validation (from previous lecture)~~

~~Bayesian Decision Theory~~

Frequentist vs Bayesian Approach

Bayesian Parameter Estimation

Maximum a Posteriori Estimation (MAP)

Bayesian Linear Regression

Generative vs Discriminative Models

Naive Bayes Classifiers

Bayesian Parameter Estimation

Frequentist vs Bayesian Approaches

Frequentist Point of View

Probability is interpreted as relative frequency of events over the long run

Computed after observing many trials of the same experiment

E.g., flipping a coin N times

Frequentist vs Bayesian Approaches

Frequentist Point of View

Probability is interpreted as relative frequency of events over the long run

Computed after observing many trials of the same experiment

E.g., flipping a coin N times

Bayesian Point of View

Probability if viewed as a measure of the confidence or belief in the occurrence of an event

More room for subjective interpretation and differences in opinions

E.g., making an assumption that errors in a program follow a Poisson distribution!

Differences are more stark when the frequency of the events is low

Frequentist vs Bayesian Approaches

Frequentist Point of View

The dataset $\mathscr{D} = \{(x^1, y^1), \ldots, (x^N, y^N)\}$ is a random sample drawn from some underlying distribution $P(x, y)$

This information is incomplete and there is **uncertainty in the data**

The parameters θ of the model that describes this dataset is **fixed but unknown**

The goal is to find these parameter values that best explains this data

We saw Maximum Likelihood Estimation (MLE) as one way of achieving this

$$\theta^* = \arg\max_\theta \left[\prod_{i=1}^{N} P(y^i \mid x^i; \theta) \right]$$

Frequentist vs Bayesian Approaches

Frequentist Point of View

The dataset $\mathscr{D} = \{(x^1, y^1), \ldots, (x^N, y^N)\}$ is a random sample drawn from some underlying distribution $P(x, y)$

This information is incomplete and there is **uncertainty in the data**

The parameters θ of the model that describes this dataset is **fixed but unknown**

The goal is to find these parameter values that best explains this data

We saw Maximum Likelihood Estimation (MLE) as one way of achieving this

$$\theta^* = \arg\max_\theta \left[\prod_{i=1}^{N} P(y^i \mid x^i; \theta) \right]$$

Bayesian Point of View

The dataset $\mathscr{D} = \{(x^1, y^1), \ldots, (x^N, y^N)\}$ is given and fixed

This information is complete and there is **no uncertainty in the data**

The parameters θ of the model that describes this dataset is a **random variable with unknown distribution**

The goal is to find the distribution over the parameters

This distribution quantifies the uncertainty in your model

Bayesian Parameter Estimation

Start with some prior distribution $P(\theta)$

$P(\theta)$ encodes our knowledge about the parameters before looking at the data

We are also given a dataset $\mathscr{D} = \{(x^1, y^1), (x^2, y^2), \ldots, (x^N, y^N)\}$

The goal is to find the posterior distribution of θ given the dataset $\mathscr{D}$: $(P(\theta \mid \mathscr{D}))$

Bayesian Parameter Estimation

Start with some prior distribution $P(\theta)$

$P(\theta)$ encodes our knowledge about the parameters before looking at the data

We are also given a dataset $\mathscr{D} = \{(x^1, y^1), (x^2, y^2), \ldots, (x^N, y^N)\}$

The goal is to find the posterior distribution of θ given the dataset $\mathscr{D}$: $(P(\theta \mid \mathscr{D}))$

We use the Bayes Theorem

$$
\begin{aligned}
P(\theta \mid \mathscr{D}) &= \frac{P(\mathscr{D} \mid \theta) P(\theta)}{P(\mathscr{D})} \\
&= \frac{\left[\prod_{i=1}^{N} P\left(y^i \mid x^i, \theta\right)\right] P(\theta)}{\int_\theta P(\mathscr{D} \mid \theta) P(\theta) d\theta} \\
&= \frac{\left[\prod_{i=1}^{N} P\left(y^i \mid x^i, \theta\right)\right] P(\theta)}{\int_\theta \left[\prod_{i=1}^{N} P\left(y^i \mid x^i, \theta\right)\right] P(\theta) d\theta}
\end{aligned}
$$

Bayesian Parameter Estimation

For logistic regression the likelihood is given by

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

$$P(y^i \,|\, x^i, \theta) = h_\theta(x^i)^{y^i} \cdot (1 - h_\theta(x^i))^{1-y^i}$$

$$P(\theta \,|\, \mathscr{D}) = \frac{P(\mathscr{D} \,|\, \theta) P(\theta)}{P(\mathscr{D})}$$

$$= \frac{\left[\prod_{i=1}^{N} P\left(y^i \,|\, x^i, \theta\right)\right] P(\theta)}{\int_\theta P(\mathscr{D} \,|\, \theta) P(\theta) d\theta}$$

$$= \frac{\left[\prod_{i=1}^{N} P\left(y^i \,|\, x^i, \theta\right)\right] P(\theta)}{\int_\theta \left[\prod_{i=1}^{N} P\left(y^i \,|\, x^i, \theta\right)\right] P(\theta) d\theta}$$

Prediction with Bayesian Models

Given the training set $\mathscr{D}$ and a new observation x, what is the value of y

What we are looking for is the distribution $P(y \mid x, \mathscr{D})$

$$P(y \mid x, \mathscr{D}) = \int_{\theta} P(y \mid x, \theta) \cdot P(\theta \mid \mathscr{D}) d\theta$$

For logistic regression

$$P(y \mid x, \theta) = h_{\theta}(x) = \frac{1}{1 + e^{-\theta^T x}}$$

$$P(\theta \mid \mathscr{D}) = \frac{\left[\prod_{i=1}^{N} P\left(y^i \mid x^i, \theta\right)\right] P(\theta)}{\int_{\theta} \left[\prod_{i=1}^{N} P\left(y^i \mid x^i, \theta\right)\right] P(\theta) d\theta}$$

So we need to compute the following (hard to compute) integral

$$\int_{\theta} h_{\theta}(x) P(\theta \mid \mathscr{D}) d\theta$$

Bayesian Models in Practice

Option 1

Do away with the full integral to integrate across the entire parameter distribution

Instead first compute a **point estimate** of the parameters

$$\theta_{MAP} = \arg\max_{\theta} \prod_{i=1}^{N} P(y^i \mid x^i, \theta) \cdot P(\theta)$$

Then use this estimate to do compute y for a new x (Inference)

Called Maximum A Posteriori (MAP) estimate

Very similar to Maximum Likelihood Estimate (MLE) but with an addition prior term

Bayesian Models in Practice

Option 1

Do away with the full integral to integrate across the entire parameter distribution

Instead first compute a **point estimate** of the parameters

$$\theta_{MAP} = \arg\max_{\theta} \prod_{i=1}^{N} P(y^i \mid x^i, \theta) \cdot P(\theta)$$

Then use this estimate to do compute y for a new x (Inference)

Called Maximum A Posteriori (MAP) estimate

Very similar to Maximum Likelihood Estimate (MLE) but with an addition prior term

Option 2

Choose the distribution function $P(y \mid x, \theta)$ such that the integral can be computed analytically

Linear Regression: The Bayesian Way

Given the training dataset $\mathscr{D} = \{(x^1, y^1), (x^2, y^2), \ldots, (x^N, y^N)\}$

We assume that $y^i = \theta^T x + \epsilon^i \quad : i = 1,\ldots,N$

ϵ^i are independent noise variables drawn from a normal distribution: $\epsilon^i \sim \mathcal{N}(0,\sigma^2)$

This essentially means that $y^i \sim \mathcal{N}(\theta^T x^i, \sigma^2)$

In other words $P(y^i \,|\, x^i, \theta) = \mathcal{N}(\theta^T x^i, \sigma^2)$

Thus the joint distribution over the entire training set can be written as

$$P(\mathbf{Y} \,|\, \mathbf{X}, \theta) = \mathcal{N}(\theta^T \mathbf{X}, \sigma^2 \mathbf{I})$$

$$\mathbf{X} = \begin{pmatrix} | & \ldots & | \\ x^1 & \ldots & x^N \\ | & \ldots & | \end{pmatrix} \qquad \mathbf{Y} = [y^1, y^2, \ldots, y^N] \qquad \mathbf{I} = \begin{pmatrix} 1 & \ldots & 0 \\ \vdots & 1 & \vdots \\ 0 & \ldots & 1 \end{pmatrix}$$

We can solve the integral analytically with this form

Linear Regression: The Bayesian Way

Parameter Estimation: Computing the Posterior Distribution

Choose a prior distribution over θ: $\theta \sim \mathcal{N}(0, \Sigma = \beta^2 \mathbf{I})$

According to Bayes Theorem we have

$$P(\theta \mid \mathbf{X}, \mathbf{Y}) = \frac{P(\mathbf{Y} \mid \mathbf{X}, \theta) \cdot P(\theta)}{P(\mathbf{Y} \mid \mathbf{X})}$$

Where the denominator (the normalizing constant) is given by

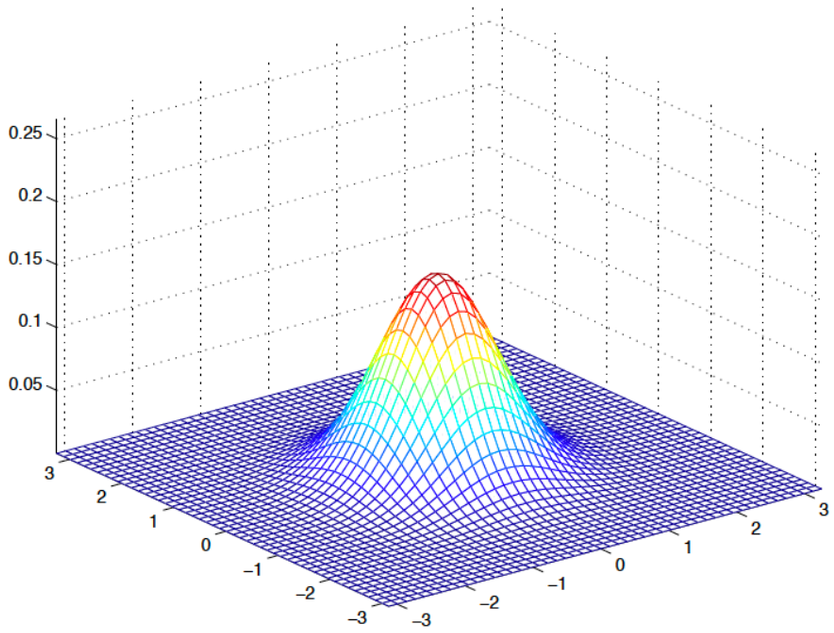
$$\int_{\theta} P(\mathbf{Y} \mid \mathbf{X}, \theta) \cdot P(\theta) d\theta$$

From the previous slide we plug $P(\mathbf{Y} \mid \mathbf{X}, \theta) = \mathcal{N}(\theta^T \mathbf{X}, \sigma^2 \mathbf{I})$

$$P(\theta \mid \mathbf{X}, \mathbf{Y}) \sim \mathcal{N}\left(\frac{1}{\sigma^2} \mathbf{Y} \mathbf{X}^{\mathbf{T}} A^{-1}, A^{-1}\right)$$

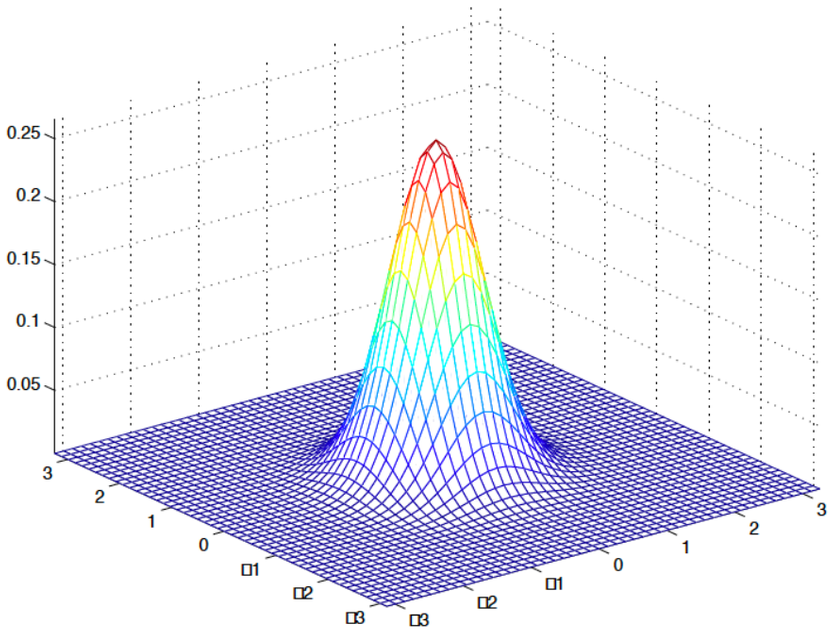
Posterior is also normally distributed

$$A^{-1} = \frac{1}{\sigma^2} \mathbf{X} \mathbf{X}^T + \Sigma_p^{-1}$$



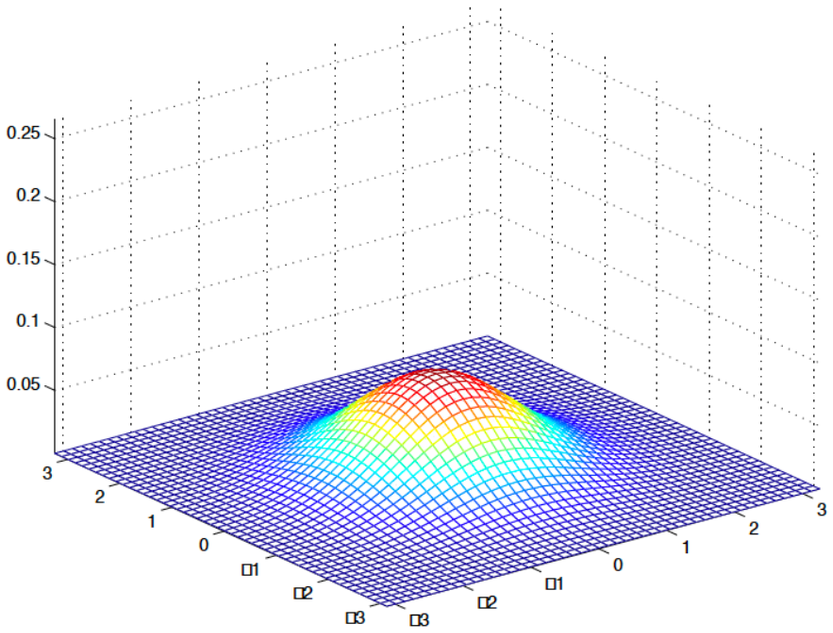
Standard Gaussian

$$\mu = 0, \Sigma = I$$



decrease variance to ``compress''

$$\mu = 0, \Sigma = 0.6I$$



increase variance to ``spread out''

$$\mu = 0, \Sigma = 2I$$

[Figure: Andrew Ng]

Linear Regression: The Bayesian Way

Parameter Estimation: Computing the Posterior Distribution

Choose a prior distribution over θ: $\theta \sim \mathcal{N}(0, \Sigma = \beta^2 \mathbf{I})$

According to Bayes Theorem we have

$$P(\theta \mid \mathbf{X}, \mathbf{Y}) = \frac{P(\mathbf{Y} \mid \mathbf{X}, \theta) \cdot P(\theta)}{P(\mathbf{Y} \mid \mathbf{X})}$$

Where the denominator (the normalizing constant) is given by

$$\int_{\theta} P(\mathbf{Y} \mid \mathbf{X}, \theta) \cdot P(\theta) d\theta$$

From the previous slide we plug $P(\mathbf{Y} \mid \mathbf{X}, \theta) = \mathcal{N}(\theta^T \mathbf{X}, \sigma^2 \mathbf{I})$

$$P(\theta \mid \mathbf{X}, \mathbf{Y}) \sim \mathcal{N}\left(\frac{1}{\sigma^2} \mathbf{Y} \mathbf{X}^{\mathbf{T}} A^{-1}, A^{-1}\right)$$

Posterior is also normally distributed

$$A^{-1} = \frac{1}{\sigma^2} \mathbf{X} \mathbf{X}^T + \Sigma_p^{-1}$$

How to Choose a Prior

Prior should reflect your knowledge about the distribution before looking at the data

Objective Priors: should maximize the impact of data on posterior

Non-Informative Priors are a type of objective priors (e.g., uniform distribution)

Subjective Priors: that capture some domain/external knowledge about the problem

E.g., gaussian priors

Conjugate Priors: a prior is considered conjugate with respect to the likelihood if the resulting posterior distribution is of the same form as the prior distribution — used primarily for simplifying the calculation of the integral

Generative and Discriminative Learning

Methods for Solving Classification Task

Given the training set $\mathscr{D} = \{(x^1, y^1), \ldots, (x^N, y^N)\}$

Let x be the input (images), find y (apples or oranges)?

Method 1

Find the decision boundary (the separating hyper-plane) which separates apples from oranges

Given a new image x find out which side of the hyper-plane is it on

Based on that assign the label y of apple or orange to x

Discriminative Models

Method 2

Look at the data and explicitly model what images of apples and oranges look like

Given a new image x compare against the models of apples and oranges

Pick the class that best explains the input x

Generative Models

Discriminative Learning

Given the training set $\mathscr{D} = \{(x^1, y^1), \ldots, (x^N, y^N)\}$. Let x be the input find y?

Explicitly model the conditional distribution $P(y \mid x, \theta)$

Learning involves maximizing the conditional likelihood given the training data to find the parameters θ

Consider Logistic Regression

$$P(y \mid x; \theta) = \frac{1}{1 + e^{-\theta^T x}}$$

Then maximizing the conditional likelihood (of the training set) to find θ boils down to

$$\max_{\theta} \mathscr{L}(\mathbf{X}, \mathbf{Y}, \theta) = \sum_{i=1}^{N} P(y^i \mid x^i; \theta)$$

Generative Learning

Given the training set $\mathscr{D} = \{(x^1, y^1), \ldots, (x^N, y^N)\}$

Let x be the input find y?

Instead of explicitly modeling the conditional distribution $P(y \mid x, \theta)$ we model the joint distribution $P(x, y; \theta)$

This joint can be expressed as $P(x, y; \theta) = P(x \mid y; \theta)P(y)$

More specifically we explicitly model $P(x \mid y = 0; \theta)$ and $P(x \mid y = 1; \theta)$

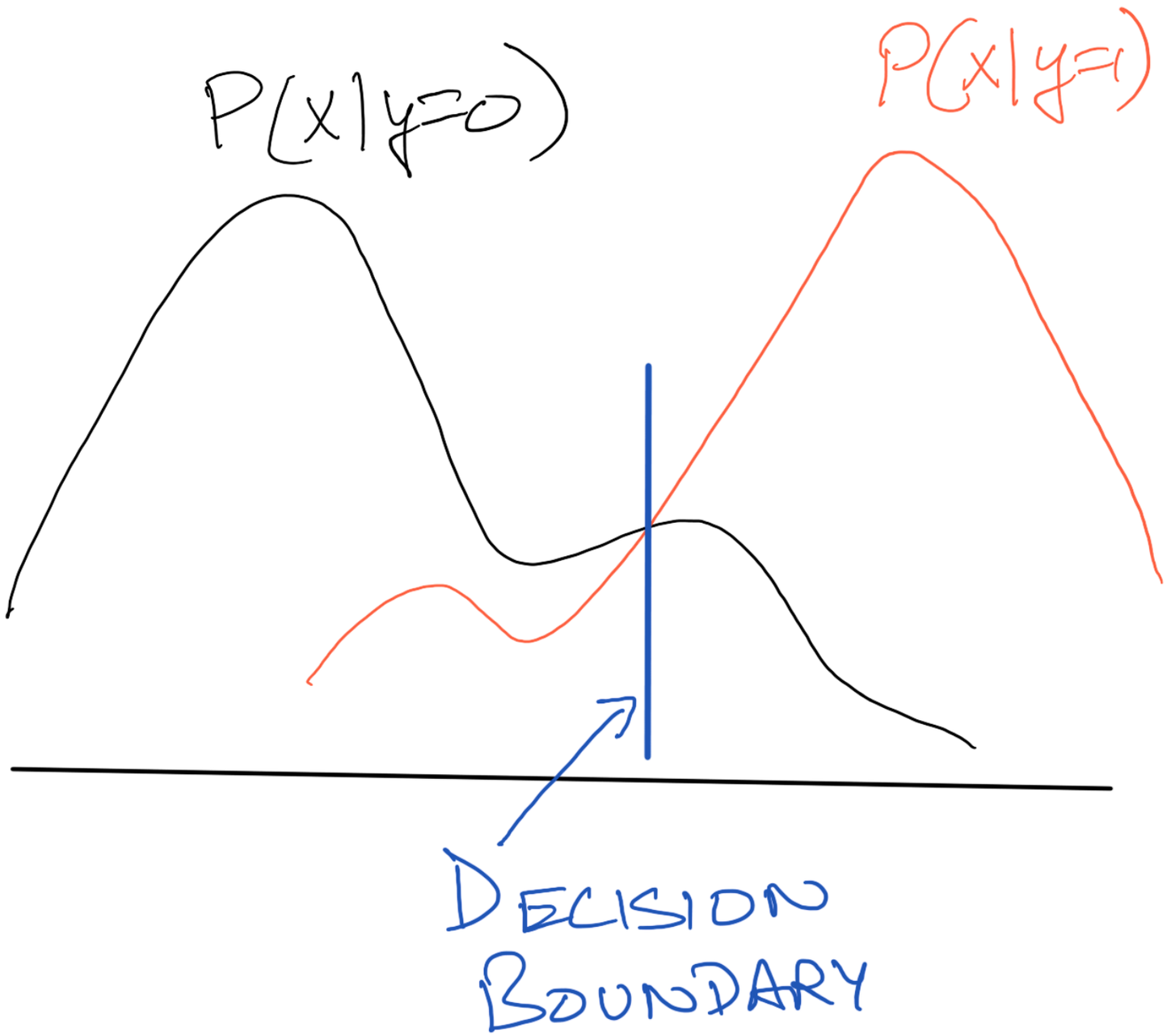
We make a decision based on Bayesian Decision Theory

$$P(y \mid x) = \frac{P(x \mid y; \theta)P(y)}{P(x)}$$

The normalizing constant $P(x)$ can be expressed as

$$P(x) = P(x \mid y = 1; \theta)P(y = 1) + P(x \mid y = 0; \theta)P(y = 0)$$

Discriminative vs Generative Learning



Naive Bayes Classifier

Naive Bayes Classifier

An example of generative models

Under rather strong assumptions

Consider a problem where you have large number of features

$x^i = [x_1^i, x_2^i, \ldots, x_n^i]$ where n is large

The explicitly modeling the joint likelihood $P(\mathbf{x} \mid y)$ of n random variables is hard

Example if $x \in \{0,1\}^n$ and is modeled with a multinomial distribution

Then there are 2^n possible outcomes which is exponential in n

This is generally true: modeling $P(\mathbf{x} \mid y)$ is challenging in most realistic situations

$$P(y \mid x) = \frac{P(x \mid y; \theta) P(y)}{P(x)}$$

Naive Bayes Classifier

Naive Bayes Classifier makes certain simplifying assumptions

It assumes that the features x_i are conditionally independent

$$P(x_i \mid y, x_j) = P(x_i \mid y) \; \forall i \neq j$$

Conditional Independence

Consider the case of $n = 3$

$$\begin{aligned} P(x_1, x_2, x_3 \mid y) &= P(x_1 \mid x_2, x_3, y) P(x_2, x_3 \mid y) \\ &= P(x_1 \mid x_2, x_3, y) P(x_2 \mid x_3, y) P(x_3 \mid y) \\ &= P(x_1 \mid y) P(x_2 \mid y) P(x_3 \mid y) \end{aligned}$$

Naive Bayes Classifier

In general with the conditional independence assumption we have

$$
\begin{aligned}
P(\mathbf{x}\,|\,y) &= P(x_1, x_2, \ldots, x_n\,|\,y) \\
&= P(x_1\,|\,y)P(x_2\,|\,x_1, y)P(x_3\,|\,x_1, x_2, y), \ldots, P(x_n\,|\,x_1, x_2, \ldots, x_{n-1}, y) \\
&= P(x_1\,|\,y)P(x_2\,|\,y)\cdots P(x_n\,|\,y) \\
&= \prod_{i=1}^{n} P(x_i\,|\,y)
\end{aligned}
$$

This means that features are independent given a class

$$P(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} P(x_i\,|\,y)$$

Naive Bayes Classifier

The Naive Bayes assumption reduces the parameters significantly

In the case of our multinomial example we had 2^n parameters

If we make Naive Bayes assumption then each x_i given a class y is now a Bernoulli variable

$P(x_i \mid y = 0) \sim Bernoulli$ with parameters $\omega_{i|y=0}$ and $P(x_i \mid y = 1) \sim Bernoulli$ with parameters $\omega_{i|y=1}$

Thus the total number of parameters is no $2n$

Model the class prior with another Bernoulli distribution with parameters ω_y

$$\mathscr{L}(\omega_y, \omega_{1|y=0}, \omega_{1|y=1}, \ldots, \omega_{n|y=0}, \omega_{n|y=1}) = \prod_{i=1}^{n} P(x^i, y^i)$$

End of Lecture 05