Logistics

- Project 1: Keep going!

- Coming in <2 weeks: Midterm
    - Pen and paper, in class. Bring one sheet of notes

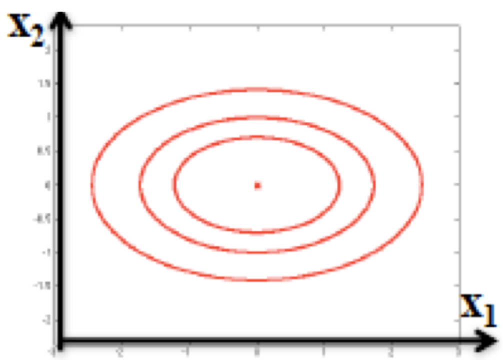
- HW4 out tonight, due in TWO WEEKS

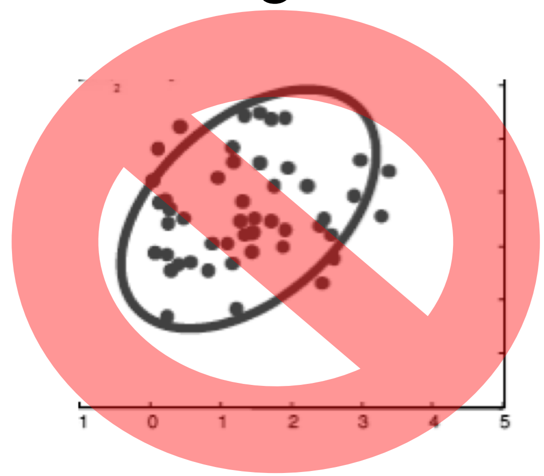
Tufts COMP 135: Introduction to Machine Learning

https://www.cs.tufts.edu/comp/135/2019s/

Classifiers that use Bayes Theorem, especially

Naïve Bayes





Many slides attributable to:
Erik Sudderth (UCI), Emily Fox (UW),
Finale Doshi-Velez (Harvard)
James, Witten, Hastie, Tibshirani (ISL/ESL books)

Prof. Mike Hughes

Objectives Today:
Bayes Theorem & Classification

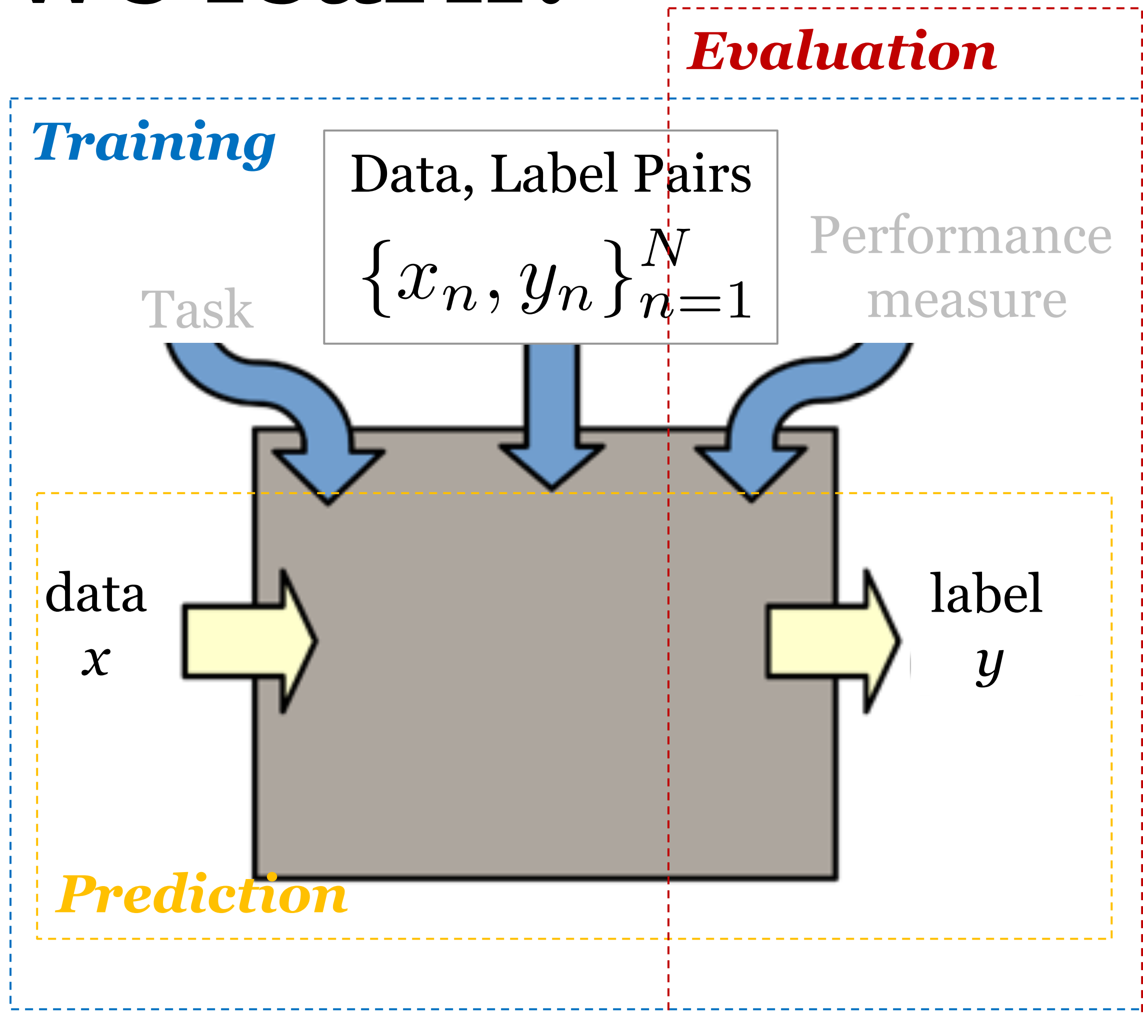
- Review: **Neural Nets**
- Two kinds of classifiers
  - Discriminative
  - Generative
- Bayes Theorem
- Using Bayes Theorem for Classification
  - Naïve Bayes: Each feature is independent
  - “Joint” Bayes: Capture class-specific correlations

What will we learn?

Supervised Learning

Unsupervised Learning

Reinforcement Learning



Task: Binary Classification

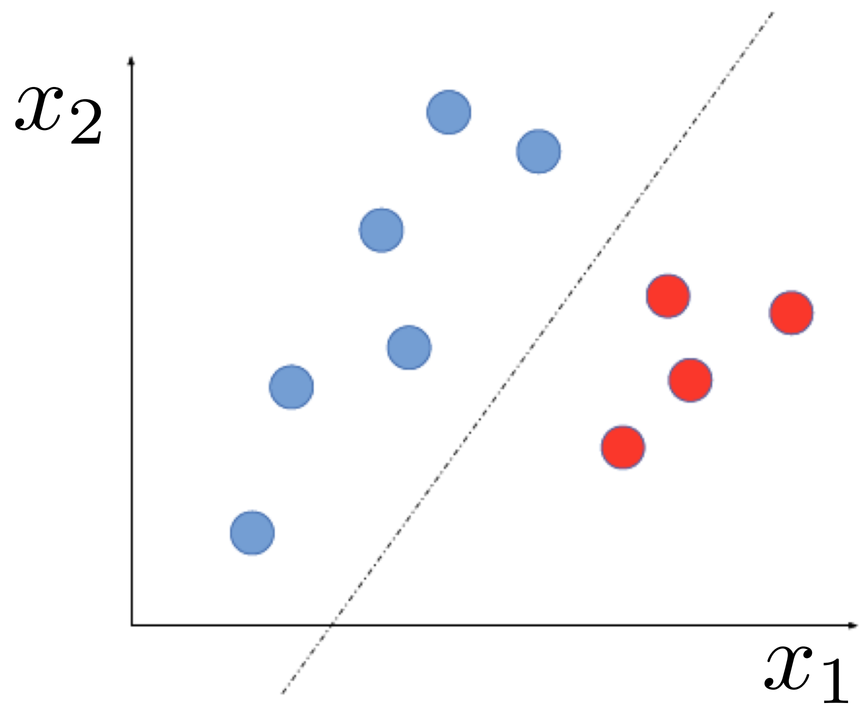
Supervised Learning

binary classification

Unsupervised Learning

Reinforcement Learning

y is a binary variable (red or blue)



Representing multi-class labels

$$y_n \in \{0, 1, 2, \ldots C - 1\}$$

Encode as length-C ***one hot binary*** vector

$$\bar{y}_n = \begin{bmatrix} \bar{y}_{n1} & \bar{y}_{n2} & \ldots & \bar{y}_{nc} & \ldots & \bar{y}_{nC} \end{bmatrix}$$

```
Examples (assume C=4 labels)

    class 0:   [1 0 0 0]
    class 1:   [0 1 0 0]
    class 2:   [0 0 1 0]
    class 3:   [0 0 0 1]
```

From Vector of Reals to Vector of Probabilities

$$z_i = \begin{bmatrix} z_{i1} & z_{i2} & \dots & z_{ic} & \dots & z_{iC} \end{bmatrix}$$

$$\hat{p}_i = \left[\frac{e^{z_{i1}}}{\sum_{c=1}^{C} e^{z_{ic}}} \quad \frac{e^{z_{i2}}}{\sum_{c=1}^{C} e^{z_{ic}}} \quad \dots \quad \dots \quad \frac{e^{z_{iC}}}{\sum_{c=1}^{C} e^{z_{ic}}} \right]$$

called the "softmax" function

MLP: Multi-Layer Perceptron
1 or more hidden layers
followed by 1 output layer

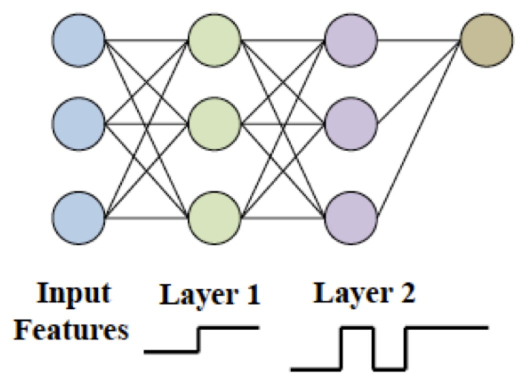
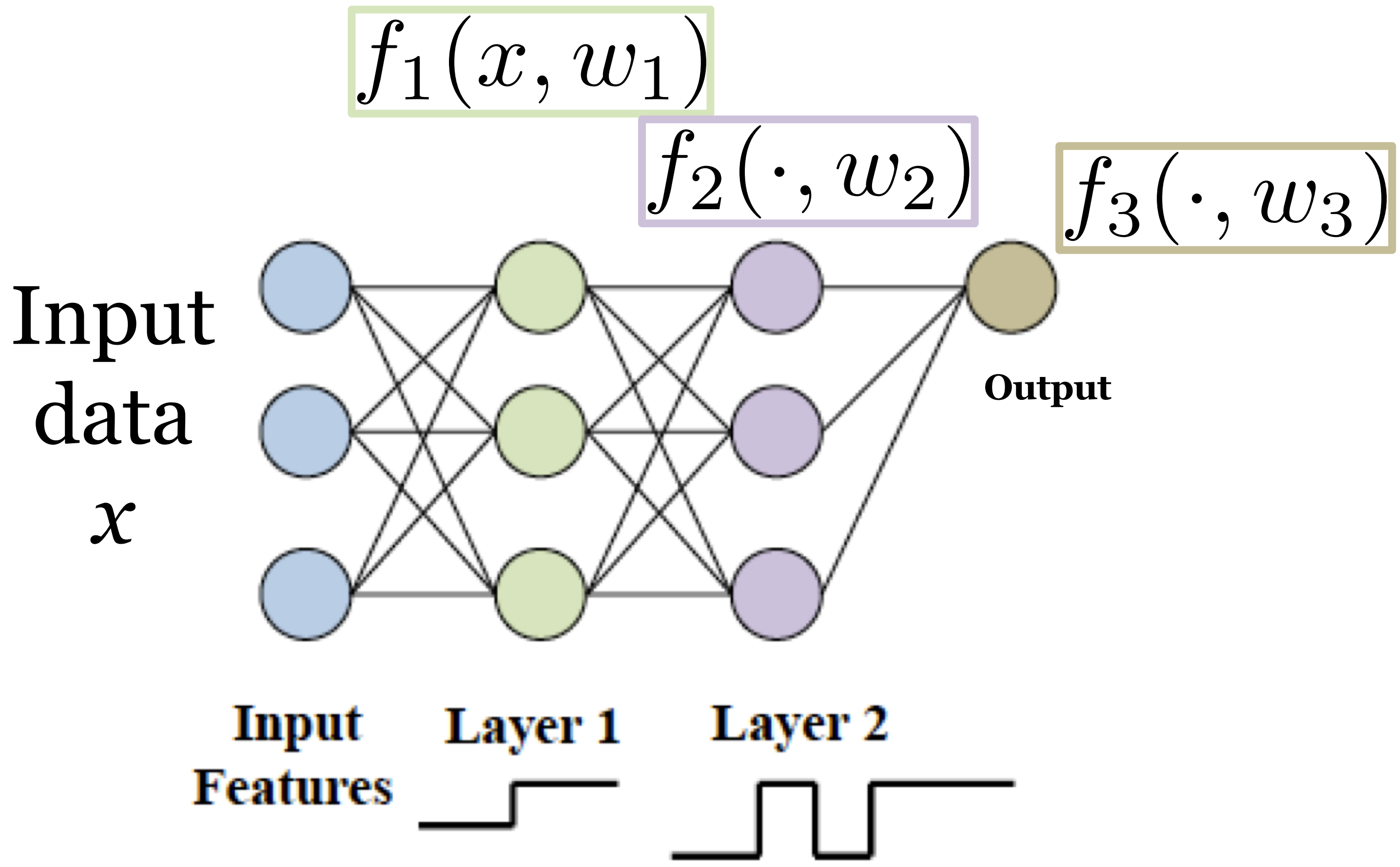
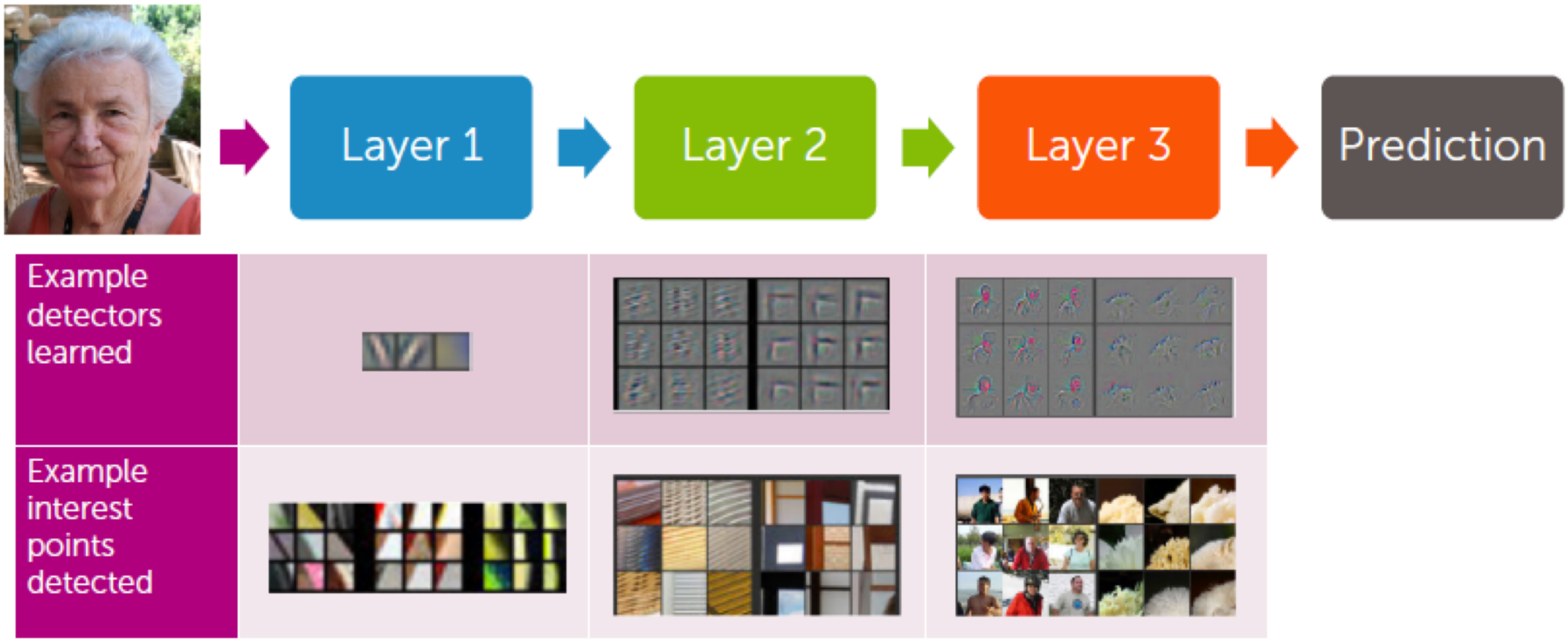


Diagram of an MLP



Each Layer Extracts “Higher Level” Features



How to train Neural Nets?

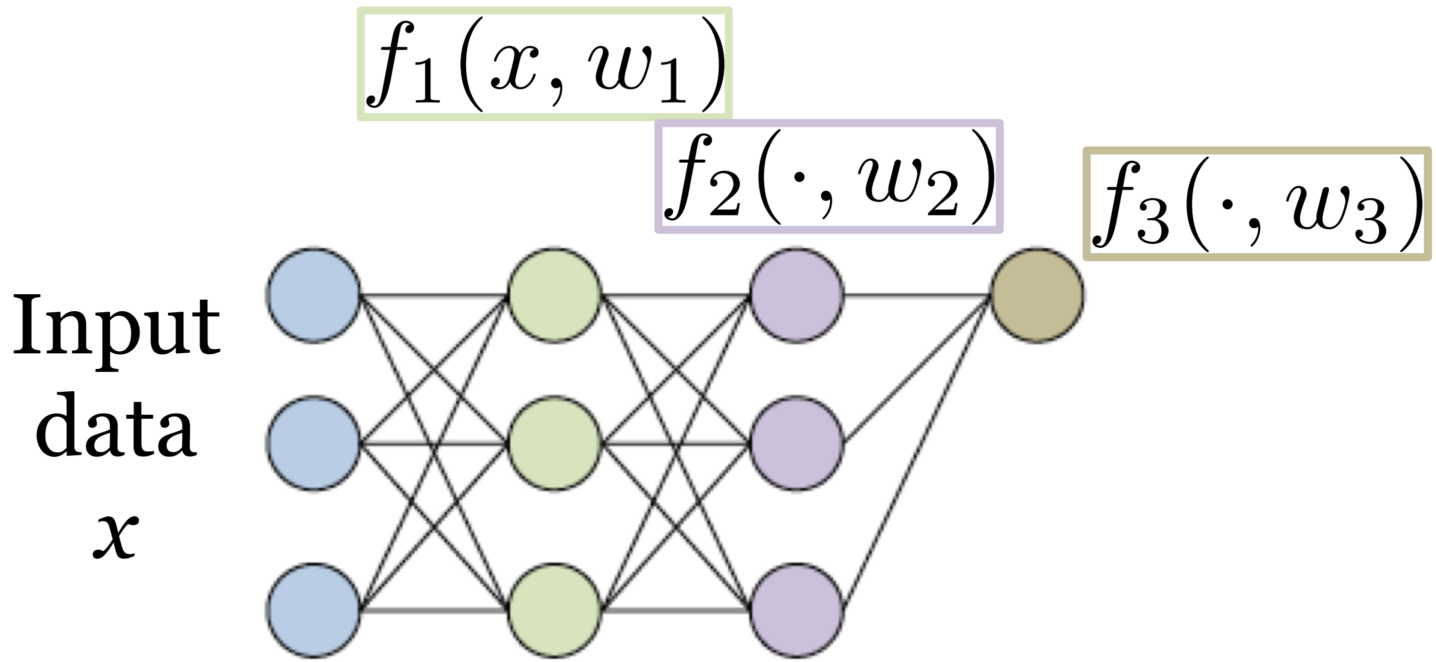
Just like logistic regression

Set up a loss function

Apply Gradient Descent!

Output as function of weights

$$f_3(f_2(f_1(x, w_1), w_2), w_3)$$



Minimizing loss for composable functions

$$\min_{w_1, w_2, w_3} \sum_{n=1}^{N} \mathrm{loss}(y_n, f_3(f_2(f_1(x_n, w_1), w_2), w_3))$$

Loss can be:

- Squared error for regression problems
- Log loss for multi-way classification problems
- … many others possible!

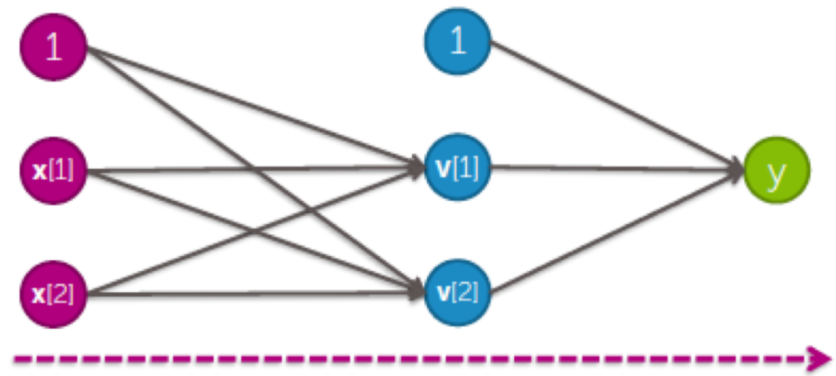
Compute loss via Forward Propagation

For fixed weights, forming predictions is easy!

Compute values left to right

1. Inputs: **x**[1],...,**x**[d]
2. Hidden: **v**[1],...,**v**[d]
3. Output: y

$w^{(1)}\ b^{(1)} \qquad w^{(2)}\ b^{(2)}$



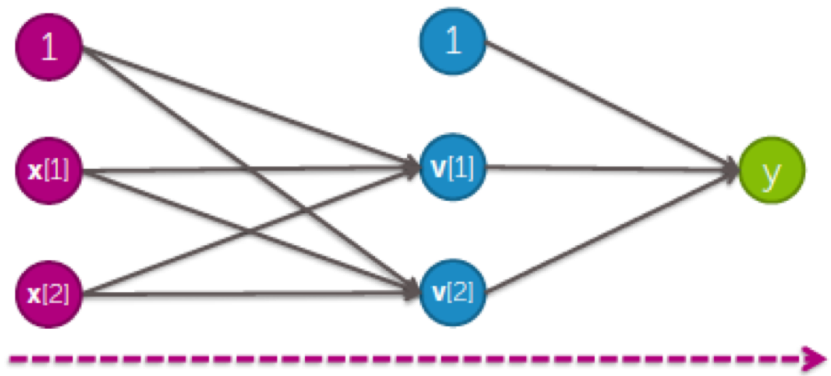
Compute loss via Forward Propagation

For fixed weights, forming predictions is easy!

Compute values left to right

1. Inputs: **x**[1],...,**x**[d]
2. Hidden: **v**[1],...,**v**[d]
3. Output: y

$w^{(1)}\ b^{(1)} \qquad w^{(2)}\ b^{(2)}$



Step 2:

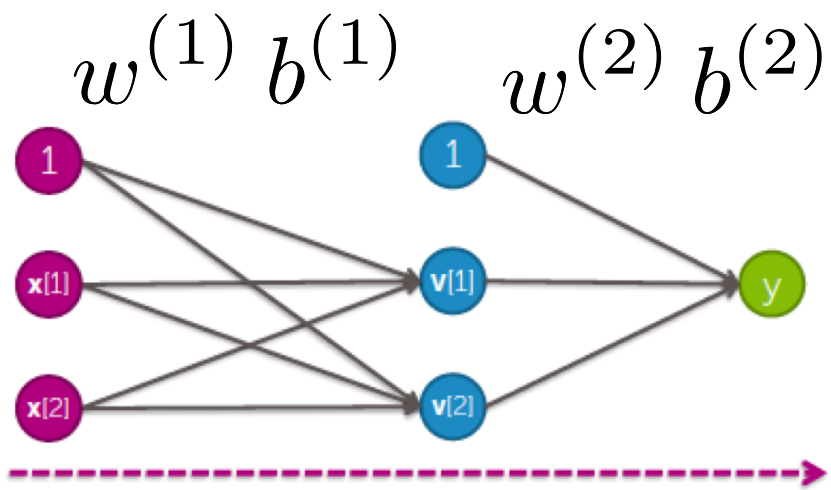
```
v = activation(np.dot(w1, x) + b1)
```

Compute loss
via Forward Propagation

For fixed weights, forming predictions is easy!

Compute values left to right

1. Inputs: **x**[1],…,**x**[d]
2. Hidden: **v**[1],…,**v**[d]
3. Output: y



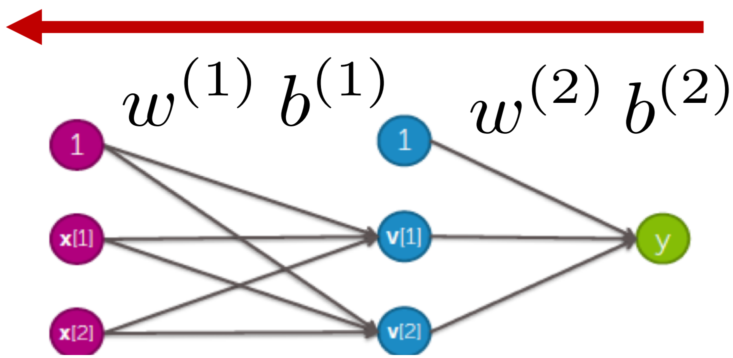
Step 2:

```
v = activation(np.dot(w1, x) + b1)
```

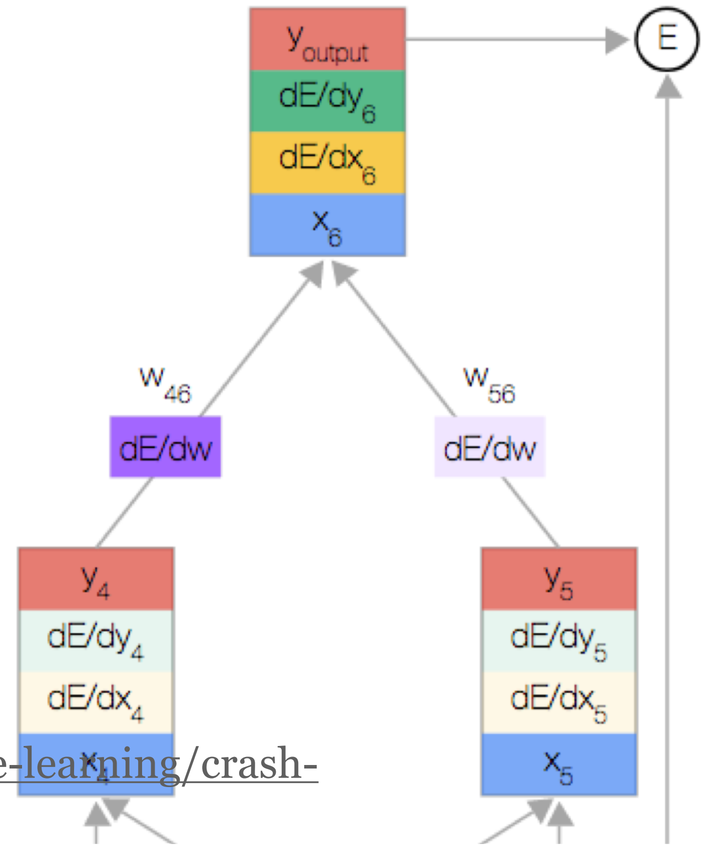
Step 3:

```
yhat = np.dot(w2, v) + b2
```

Compute gradient via **Back Propagation**

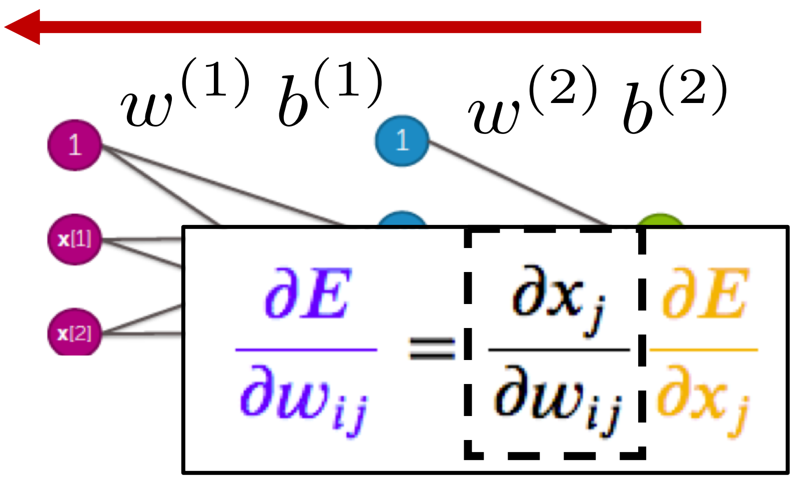


Goal: Compute gradient wrt weights



Visual Demo:
https://google-developers.appspot.com/machine-learning/crash-course/backprop-scroll/

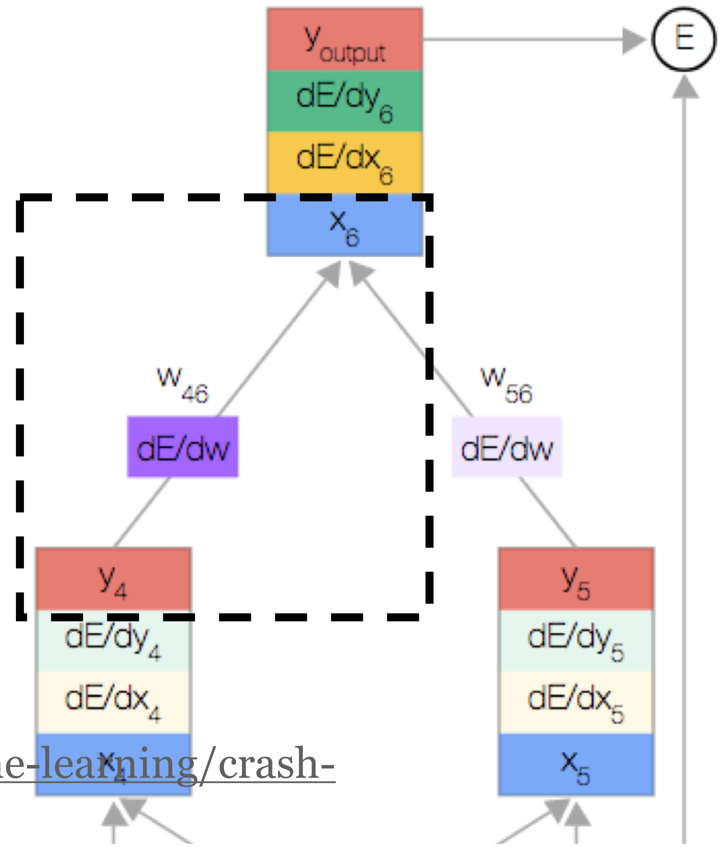
Compute gradient via **Back Propagation**



$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial x_j}{\partial w_{ij}} \frac{\partial E}{\partial x_j}$$

Compute use input y_4 from forward pass

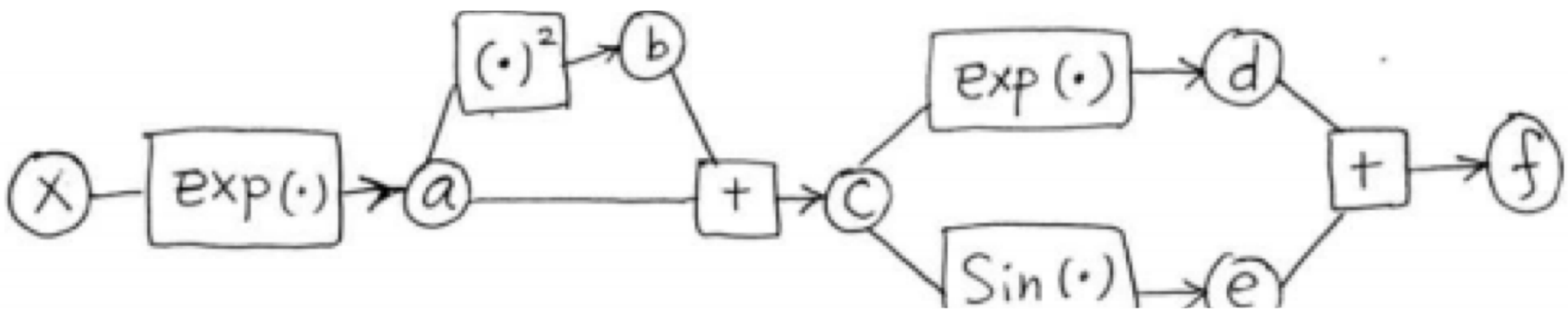
Look up from previous computation



Visual Demo:
https://google-developers.appspot.com/machine-learning/crash-course/backprop-scroll/

Automatic Differentiation can be done via Backprop!

$$f = \exp\left(\exp(x) + \exp(x)^2\right) + \sin\left(\exp(x) + \exp(x)^2\right)$$



Back Propagation

(Do forward propagation)

$\frac{df}{dx_N} \leftarrow 1$

For $i = N-1,\ N-2,\ \ldots 1$:

$$\frac{df}{dx_i} \leftarrow \sum_{j:i\in\mathrm{Pa}(j)} \frac{df}{dx_j}\frac{dg_j}{dx_i}.$$

Credit: Justin Domke (UMass)
https://people.cs.umass.edu/~domke/courses/sml/09autodiff_nnets.pdf

Objectives Today: Bayes Theorem & Classification

- Review: **Neural Nets**
- Two kinds of classifiers
  - Discriminative
  - Generative
- Bayes Theorem
- Using Bayes Theorem for Classification
  - Naïve Bayes: Each feature is independent
  - "Smarter" Bayes: Capture class-specific correlations
    - Quadratic Discriminant Analysis

Recall: Rules of Probability

X

Y

	Candidate A	Candidate B
Young voters	0.28	0.42
Senior voters	0.24	0.06

sum rule $$p(X) = \sum_Y p(X, Y)$$

product rule $$p(X, Y) = p(Y|X)p(X)$$
$$= p(X|Y)p(Y)$$

Kinds of Probabilistic Classifiers

- Discriminative
  - Directly learn parameters that define the label given data distribution

$$p(Y = y|X = x)$$

Examples: logistic regression, NNs

Kinds of Probabilistic Classifiers

Discriminative

- Directly learn parameters that define the label given data distribution $p(Y = y|X = x)$

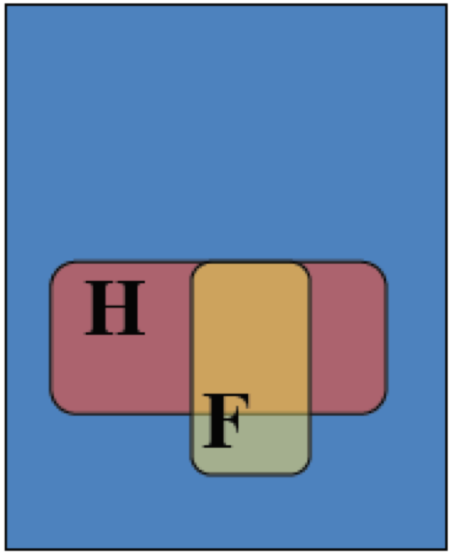
Examples: logistic regression, NNs

Generative

- Learn parameters for two distributions
  - Probability of label $p(Y = y)$
  - Probability of data given label $p(X = x|Y = y)$
- Combine via Bayes theorem to make predictions

Probabilistic Reasoning

- Two events: headache, flu
- p(H) = 1/10
- p(F) = 1/40
- p(H|F) = 1/2

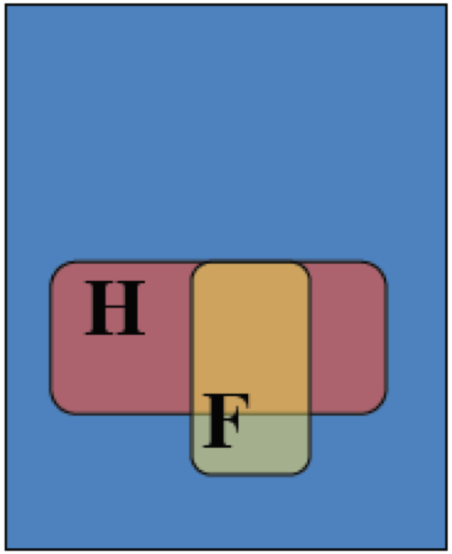


You wake up with a headache.
What is chance that you have flu?
How to write this is a probability?

Credit: E. Sudderth

Probabilistic Reasoning

- Two events: headache, flu
- p(H) = 1/10
- p(F) = 1/40
- p(H|F) = 1/2

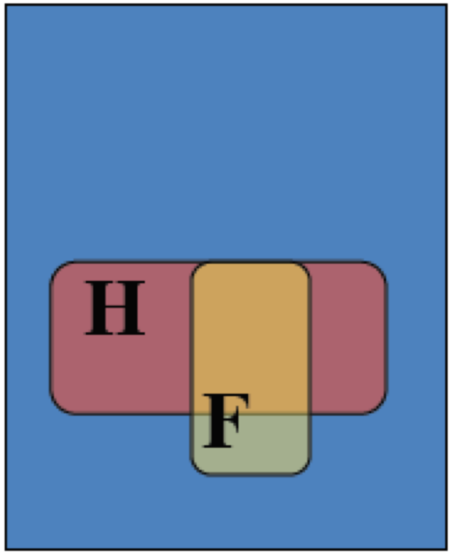


You wake up with a headache.
What is chance that you have flu?
Goal: P(F | H)

Credit: E. Sudderth

Probabilistic Reasoning

- Two events: headache, flu
- p(H) = 1/10
- p(F) = 1/40
- p(H|F) = 1/2



You wake up with a headache.
What is chance that you have flu?
Goal: P(F | H)

Credit: E. Sudderth

Probabilistic Reasoning

- Two events: headache, flu
- $p(H) = 1/10$
- $p(F) = 1/40$
- $p(H|F) = 1/2$

- $P(H \,\&\, F) = ?$
- $P(F|H) = ?$

You wake up with a headache.
What is chance that you have flu?
Goal: $P(F \mid H)$, but **first step: $P(H \,\&\, F)$**

Credit: E. Sudderth

Probabilistic Reasoning

- Two events: headache, flu
- $p(H) = 1/10$
- $p(F) = 1/40$
- $p(H|F) = 1/2$

Product rule!

- $P(H \& F) = p(F)\, p(H|F)$
  $= (1/2) * (1/40) = 1/80$
- $P(F|H) = ?$

You wake up with a headache.
What is chance that you have flu?
Goal: $P(F \mid H)$

Credit: E. Sudderth

Probabilistic Reasoning

- Two events: headache, flu
- p(H) = 1/10
- p(F) = 1/40
- p(H|F) = 1/2

P(H & F) = p(F) p(H|F)
= (1/2) * (1/40) = 1/80
P(F|H) = p(H & F) / p(H)
= (1/80) / (1/10) = 1/8

Product rule again!

You wake up with a headache.
What is chance that you have flu?
Goal: P(F | H) = 1/8

Credit: E. Sudderth

Probabilistic Reasoning

- Two events: headache, flu
- $p(H) = 1/10$
- $p(F) = 1/40$
- $p(H|F) = 1/2$

$$
\boxed{\begin{aligned}
P(H \,\&\, F) &= p(F)\, p(H|F) \\
&= (1/2) * (1/40) = 1/80 \\
P(F|H) &= p(H \,\&\, F) / p(H) \\
&= (1/80) / (1/10) = 1/8
\end{aligned}}
$$

Product rule again!

You wake up with a headache.
What is chance that you have flu?
Goal: $P(F \mid H) = 1/8$

Credit: E. Sudderth

Bayes Theorem:

$$p(Y = y | X = x) = \frac{p(X = x | Y = y) p(Y = y)}{p(X = x)}$$

Bayes Theorem:

$$p(Y = y | X = x) = \frac{p(X = x | Y = y) p(Y = y)}{p(X = x)}$$

$$p(Y = y | X = x) = \frac{p(X = x | Y = y) p(Y = y)}{\sum_{y'} p(X = x, Y = y')}$$

Use sum rule to rewrite the denominator

Bayes Classifier: Prediction

Given: $\boxed{p(Y = y)}$

$\boxed{p(X = x | Y = y)}$

Prediction: *just plug into Bayes Rule and compute!*

$$p(Y = y | X = x) = \frac{\boxed{p(X = x | Y = y)}\,\boxed{p(Y = y)}}{\sum_{y'} \boxed{p(X = x | Y = y')}\,\boxed{p(Y = y')}}$$

Bayes Classifiers: Training

1) Estimate the label probability $\boxed{p(Y = y)}$

How: Just measure empirical frequencies!

Y

Features	# bad	# good
X=0	42	15
X=1	338	287
X=2	3	5

p(y)	**383/690**	**307/690**

Bayes Classifiers: Training

1) Estimate the label probability

How: Just measure empirical frequencies!

2) Estimate the data-given-label probability $\boxed{p(X = x|Y = y)}$

2a) Separate features into label-specific datasets

$$D_c = \{ x^{(j)} : y^{(j)} = c \}$$

2b) Estimate a density from the label-specific data

pmf (if x discrete) or pdf (if x continuous)

Bayes Classifiers: Training

2) Estimate the data-given-label probability $\boxed{p(X = x|Y = y)}$

2a) Separate features into label-specific datasets

$$D_c = \{ x^{(j)} : y^{(j)} = c \}$$

2b) Estimate a density from the label-specific data

pmf (if x discrete) or pdf (if x continuous)

Features	# bad	# good
X=0	42	15
X=1	338	287
X=2	3	5

p(y)	383/690	307/690

$\Longrightarrow$

p(x | y=0)	p(x | y=1)
42 / 383	15 / 307
338 / 383	287 / 307
3 / 383	5 / 307

Bayes Classifiers: Training

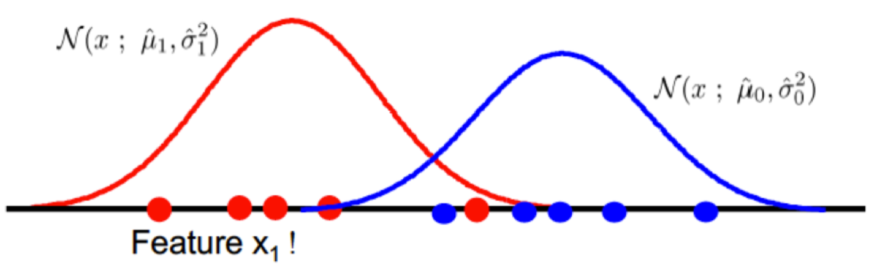
2) Estimate the data-given-label probability

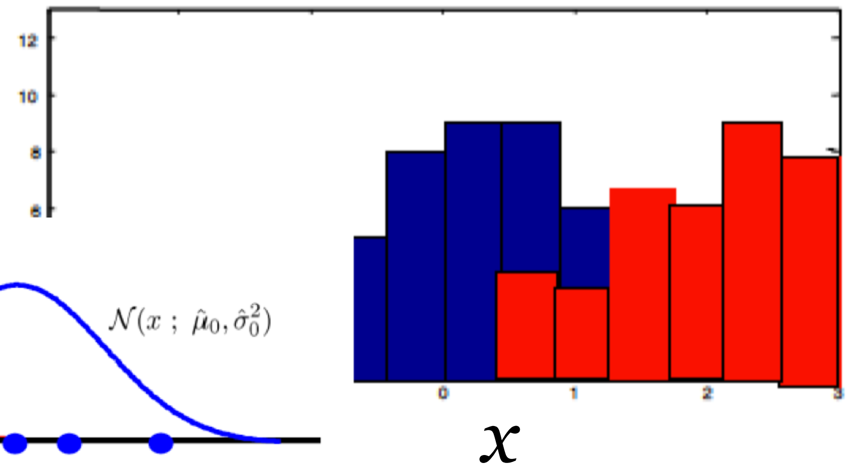
2a) Separate features into label-specific datasets

$$D_c = \{ x^{(j)} : y^{(j)} = c \}$$

2b) Estimate a density from the label-specific data

pmf (if x discrete) or **pdf (if x continuous)**





When x has many features

Feature vector x has 3 binary features, A, B, & C

Enumerate all possible values of x:

A	B	C
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

When x has many features

Feature vector x has 3 binary features, A, B, & C

Enumerate all possible values of x, ***then assign each value a class-specific probability***

$$\boxed{p(X = x | Y = y)}$$

A	B	C	p(A,B,C | y=1)
0	0	0	0.50
0	0	1	0.05
0	1	0	0.01
0	1	1	0.10
1	0	0	0.04
1	0	1	0.15
1	1	0	0.05
1	1	1	0.10

When x has many features

Feature vector x has 3 binary features, A, B, & C

Enumerate all possible values of x, ***then assign each value a class-specific probability***

$\boxed{p(X = x | Y = y)}$

A	B	C	p(A,B,C | y=1)
0	0	0	0.50
0	0	1	0.05
0	1	0	0.01
0	1	1	0.10
1	0	0	0.04
1	0	1	0.15
1	1	0	0.05
1	1	1	0.10

How many values needed for M binary features?
How many for M features that each take K possible values?

When x has many features

Feature vector x has 3 binary features, A, B, & C

$\boxed{p(X = x|Y = y)}$

A	B	C	p(A,B,C | y=1)
0	0	0	0.50
0	0	1	0.05
0	1	0	0.01
0	1	1	0.10
1	0	0	0.04
1	0	1	0.15
1	1	0	0.05
1	1	1	0.10

Enumerate all possible values of x, then assign each value a class-specific probability

How many values needed for M binary features? 2^M

How many for M features that each take K possible values? K^M

Rare features

- Suppose in our training data of size 500, one possible feature vector [0 0 1] never occurs with label 1, and occurs once with label 0.

- What will be the estimated probabilities

  - $P(X = [0\ 0\ 1] \mid Y = 1)$?
  - $P(X = [0\ 0\ 1] \mid Y = 0)$?

Rare features

- Suppose in our training data of size 500, one possible feature vector [0 0 1] never occurs with label 1, and occurs once with label 0.

- What will be the estimated probabilities?

  - $P(X = [0\ 0\ 1] \mid Y = 1)$? 0
  - $P(X = [0\ 0\ 1] \mid Y = 0)$? small
    - *(can't say unless we know how often y=0 occurs)*

Strategy to prevent overfitting: Reduce model complexity

- Model 1:
  - Assume nothing about p(X | Y)
  - Define joint proba table for all 2^M feature vectors
  - Need 2^M numbers for each class y

- Model 2:
  - Assume each feature occurs independently

$$p(X = [x_1, x_2, x_3]|Y = y) = \boxed{p(X_1 = x_1|Y = y)}\boxed{p(X_2 = x_2|Y = y)}\boxed{p(X_3 = x_3|Y = y)}$$

  - How many numbers needed for each class y? _____

Credit: E. Sudderth

Strategy to prevent overfitting: Reduce model complexity

- Model 1:
  - Assume nothing about p(X | Y)
  - Define joint proba table for all 2^M feature vectors
  - Need 2^M numbers for each class y

- Model 2:
  - Assume each feature is independent given label

$$p(X = [x_1, x_2, x_3] | Y = y) = \boxed{p(X_1 = x_1 | Y = y)}\boxed{p(X_2 = x_2 | Y = y)}\boxed{p(X_3 = x_3 | Y = y)}$$

  - How many numbers needed for each class y? 2 M

Credit: E. Sudderth

Naïve Bayes:

Assume independence to make many features tractable

- Model 1: "Joint Bayes"

A	B	C	p(A,B,C | y=1)
0	0	0	0.50
0	0	1	0.05
0	1	0	0.01
0	1	1	0.10
1	0	0	0.04
1	0	1	0.15
1	1	0	0.05
1	1	1	0.10

- Model 2: "Naïve" Bayes
  - Assume each feature occurs independently given label

A	p(A|y=1)
0	.4
1	.6

B	p(B |y=1)
0	.7
1	.3

C	p(C |y=1)
0	.1
1	.9

Credit: E. Sudderth

Naïve Bayes:

Assume independence to make many features tractable

- Model 1: "Joint Bayes"

A	B	C	p(A,B,C | y=1)
0	0	0	0.50
0	0	1	0.05
0	1	0	0.01
0	1	1	0.10
1	0	0	0.04
1	0	1	0.15
1	1	0	0.05
1	1	1	0.10

- Model 2: "Naïve" Bayes
  - Assume each feature occurs independently given label

A	p(A|y=1)
0	.4
1	.6

B	p(B |y=1)
0	.7
1	.3

C	p(C |y=1)
0	.1
1	.9

Example: Spam Email Classifier

y $\in$ {spam, not spam}

X = observed words in email

- Ex: [“the” … “probabilistic” … “lottery”…]
- “1” if word appears; “0” if not

1000’s of possible words: 2^{1000s} parameters? *if we did full joint model*

# of atoms in the universe: $\approx 2^{270}$…

Model words ***given*** email type as independent

Some words more likely for spam (“lottery”)

Some more likely for non-spam (“probabilistic”)

Only 1000’s of parameters now…

What about real-valued x?

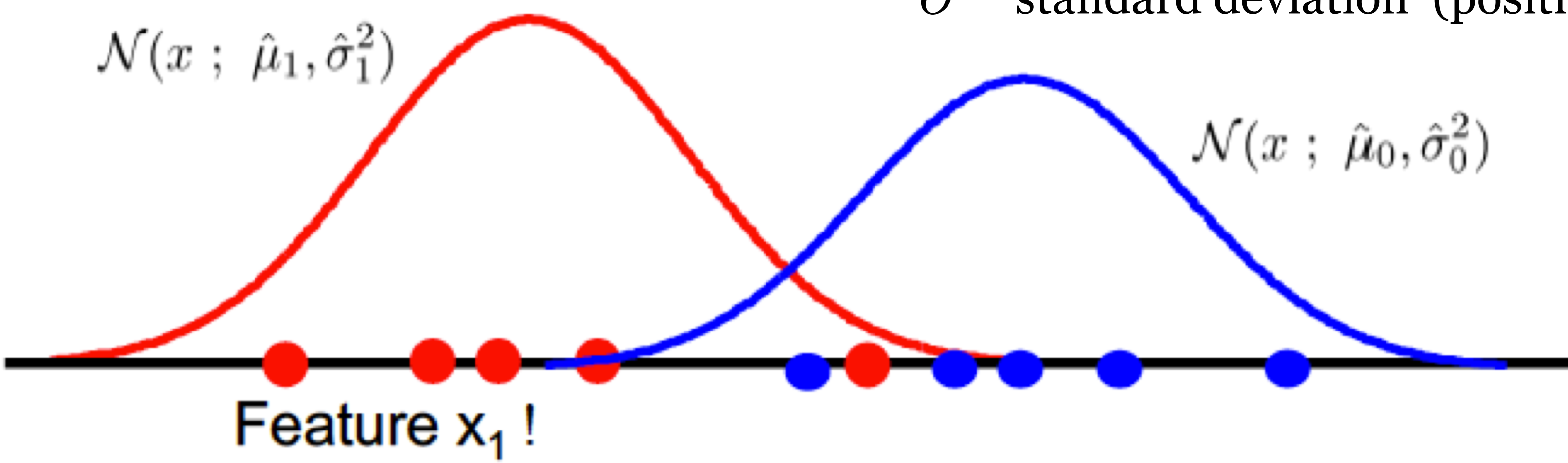
Real-valued x: Gaussian Model

Probability density function:

$$p(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi}}\frac{1}{\sigma^2}e^{-\frac{1}{2}\frac{1}{\sigma^2}(x-\mu)^2}$$

μ mean (any real value)

σ standard deviation (positive)



Easy to estimate class-specific mean and stddev from data

Credit: E. Sudderth

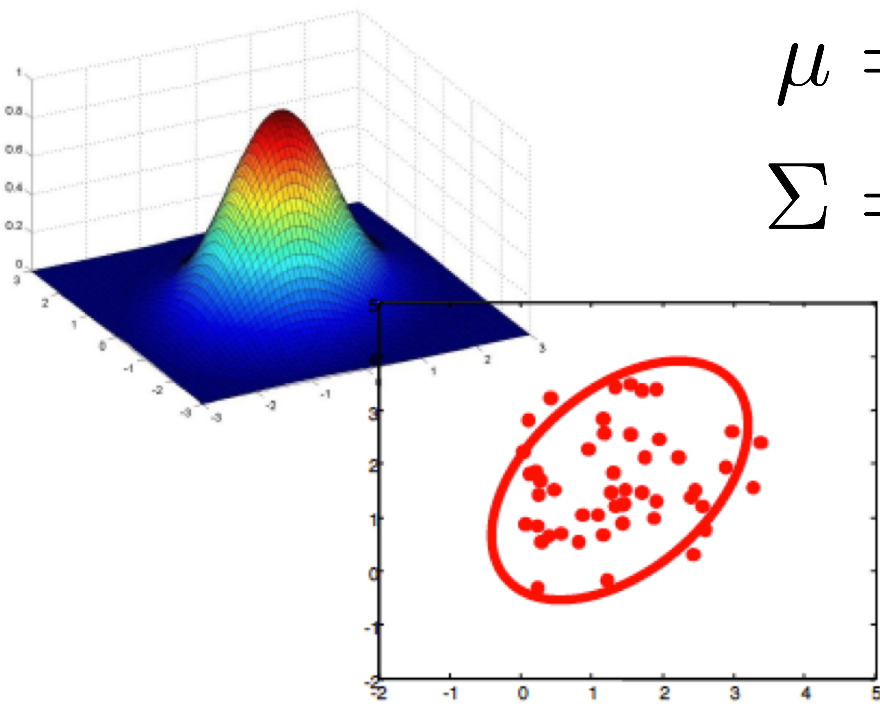
Vector x: Multivariate Gaussian

Probability density function:

$$\mathcal{N}(\underline{x}\ ;\ \underline{\mu}, \Sigma) = \frac{1}{(2\pi)^{F/2}} |\Sigma|^{-1/2} \exp\left\{ -\frac{1}{2} (\underline{x} - \underline{\mu})^T \Sigma^{-1} (\underline{x} - \underline{\mu}) \right\}$$

$\mu = F \times 1$ mean vector

$\Sigma = F \times F$ covariance matrix



Credit: E. Sudderth

Naïve Bayes for Vectors *x*

Assume each feature dimension is **independent** of others

Probability density functions:

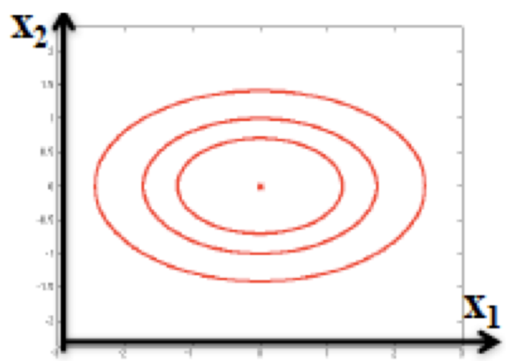
$$p(x_1, x_2) = p(x_1)p(x_2)$$

$$p(x_1) = \frac{1}{Z} \exp\left\{-\frac{1}{2\sigma_1^2}(x_1 - \mu_1)^2\right\} \qquad p(x_2) = \frac{1}{Z_2} \exp\left\{-\frac{1}{2\sigma_2^2}(x_2 - \mu_2)^2\right\}$$

Equivalent to multivariate Gaussians
With diagonal covariance:

$$\mu = [\mu_1 \ \ \mu_2]$$

$$\Sigma = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}$$



Credit: E. Sudderth

Naïve Bayes for Vectors *x*

Assume each feature dimension is **independent** of others

Probability density functions:

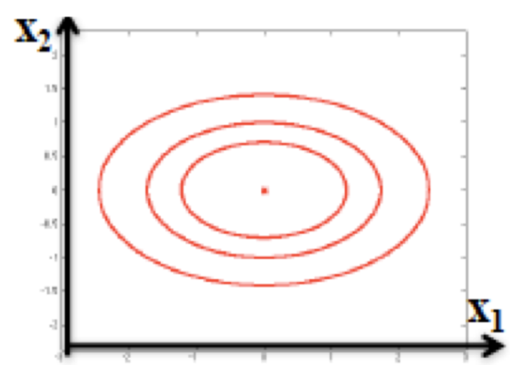
$$p(x_1, x_2) = p(x_1)p(x_2)$$

$$p(x_1) = \frac{1}{Z} \exp\left\{ -\frac{1}{2\sigma_1^2}(x_1 - \mu_1)^2 \right\} \qquad p(x_2) = \frac{1}{Z_2} \exp\left\{ -\frac{1}{2\sigma_2^2}(x_2 - \mu_2)^2 \right\}$$

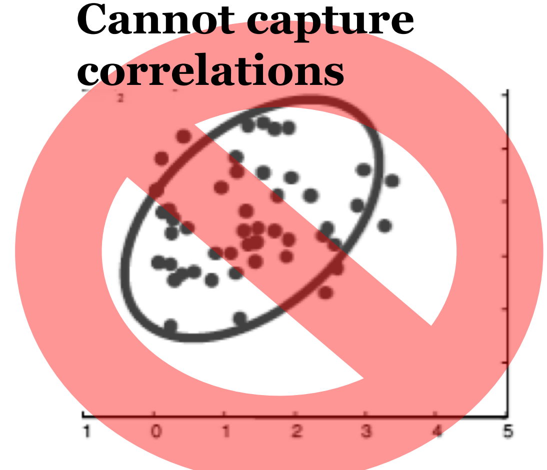
Equivalent to multivariate Gaussians
With diagonal covariance:

$$\mu = [\mu_1 \ \ \mu_2]$$

$$\Sigma = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}$$



Cannot capture correlations



Credit: E. Sudderth

Reducing complexity

Given feature vector with F dimensions

- Full-covariance Gaussian ("Joint Bayes")

$$\mu = F \times 1 \quad \text{mean vector}$$

$$\Sigma = F \times F \quad \text{covariance matrix}$$

- Diagonal-covariance Gaussian ("Naïve Bayes")

How many mean parameters?

How many covariance parameters?

Reducing complexity

Given feature vector with F dimensions

- Full-covariance Gaussian ("Joint Bayes")

$$\mu = F \times 1 \quad \text{mean vector}$$

$$\Sigma = F \times F \quad \text{covariance matrix}$$

- Diagonal-covariance Gaussian ("Naïve Bayes")

How many mean parameters? F

How many covariance parameters? F

Naïve Bayes Classifier: Advantages

- Fast to train
  - Just counting with discrete data
- Fast to do prediction at test time
- Easy to interpret parameters
- Few (if any) hyperparameters to tune

- Works well with ***small data***

Naïve Bayes Classifier: Disadvantages

- Assumptions rarely ever justified!

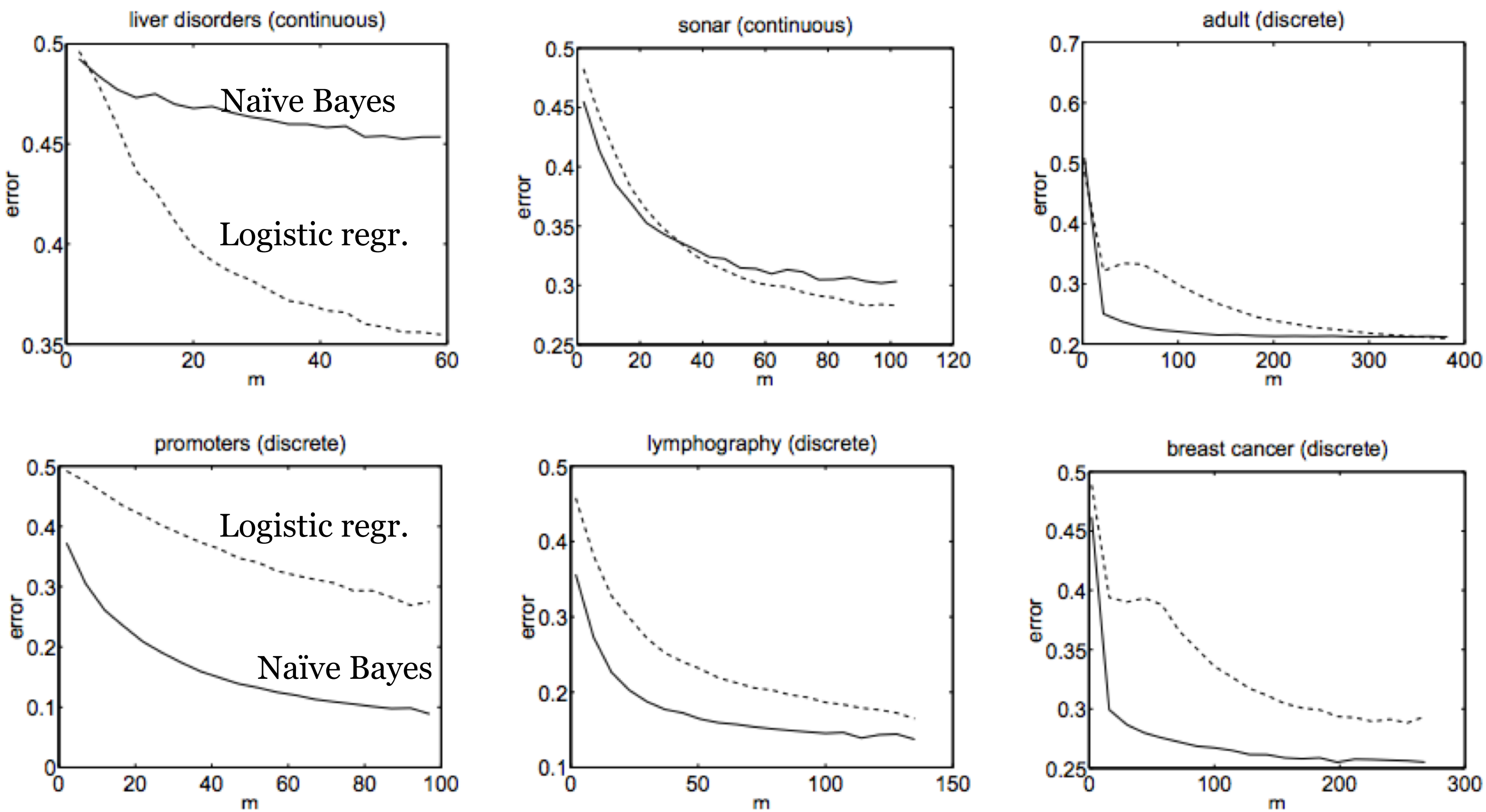
- Not very flexible model

On Discriminative vs. Generative classifiers: A comparison of logistic regression and naive Bayes

Andrew Y. Ng
Computer Science Division
University of California, Berkeley
Berkeley, CA 94720

Michael I. Jordan
C.S. Div. & Dept. of Stat.
University of California, Berkeley
Berkeley, CA 94720

Generalization Error vs Training Set Size



Comparisons in ISL Ch. 4

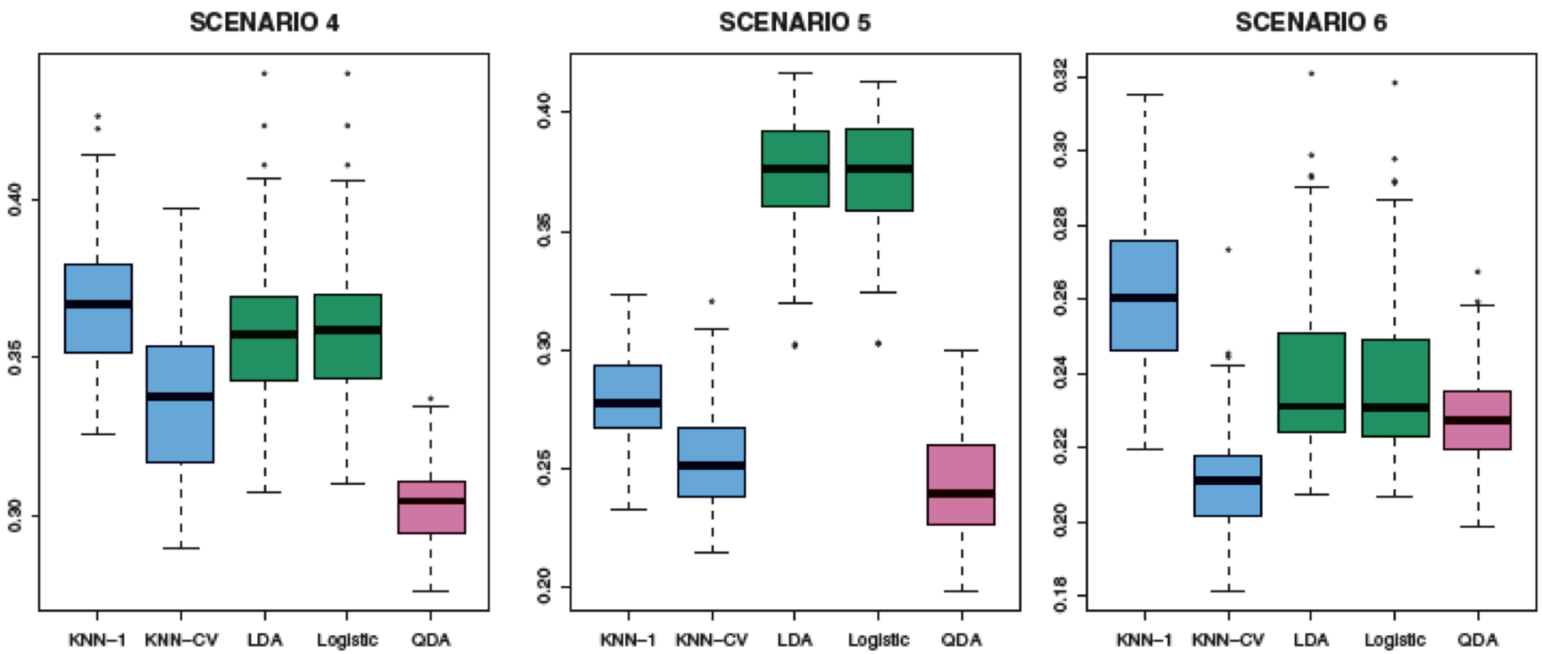


FIGURE 4.11. *Boxplots of the test error rates for each of the non-linear scenarios described in the main text.*

Objectives Today: Bayes Theorem & Classification

What have we learned?

- Two kinds of classifiers
  - Discriminative
  - Generative
- Bayes Theorem
- Using Bayes Theorem for Classification
  - Naïve Bayes: Each feature is independent
  - “Joint” Bayes: Capture class-specific correlations
    - With full-covariance Gaussians, called Quadratic Discriminant Analysis