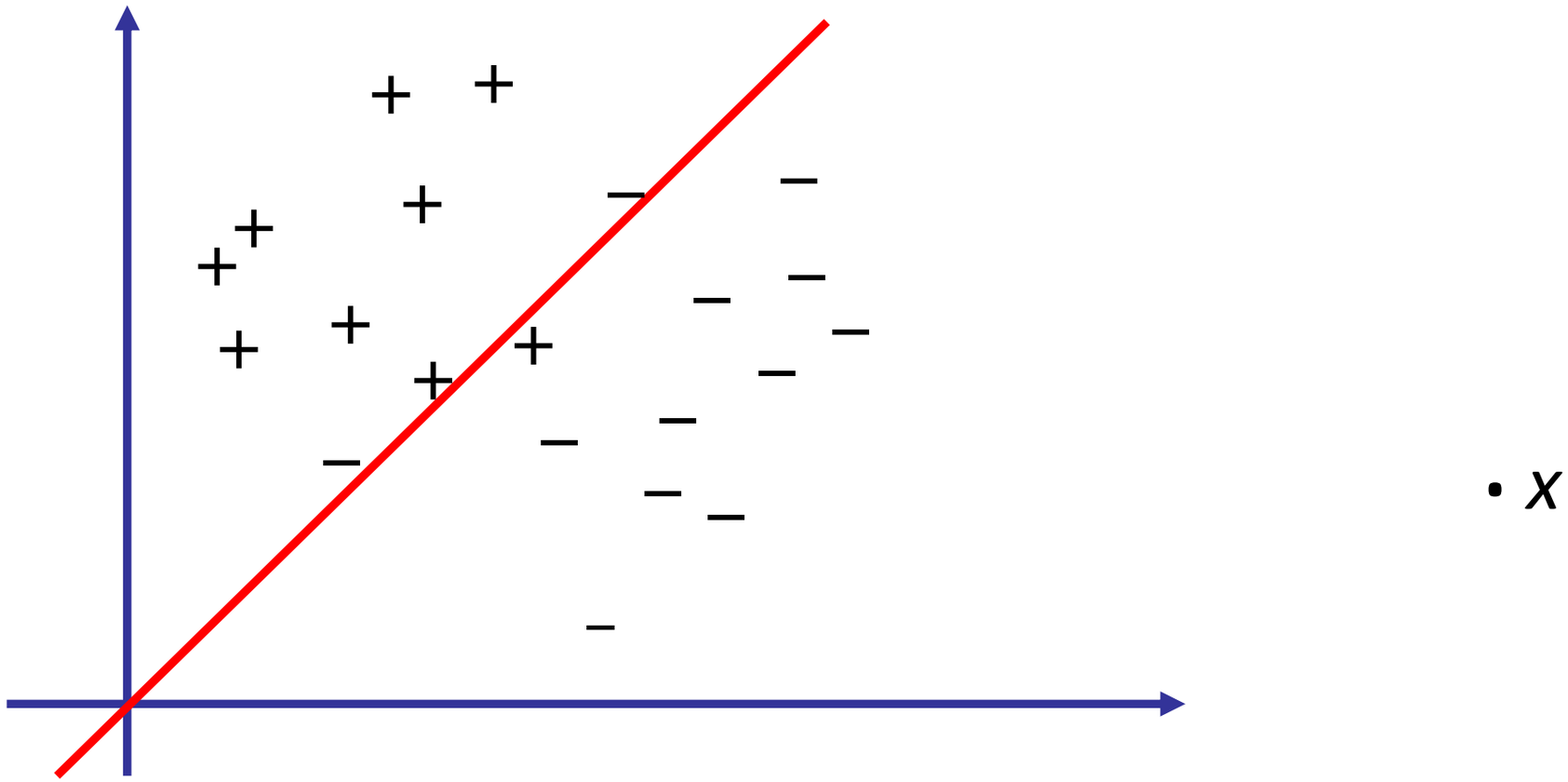
Generative Models for Classification

STAT 37710 / CMSC 35300
Rebecca Willett and Yuxin Chen

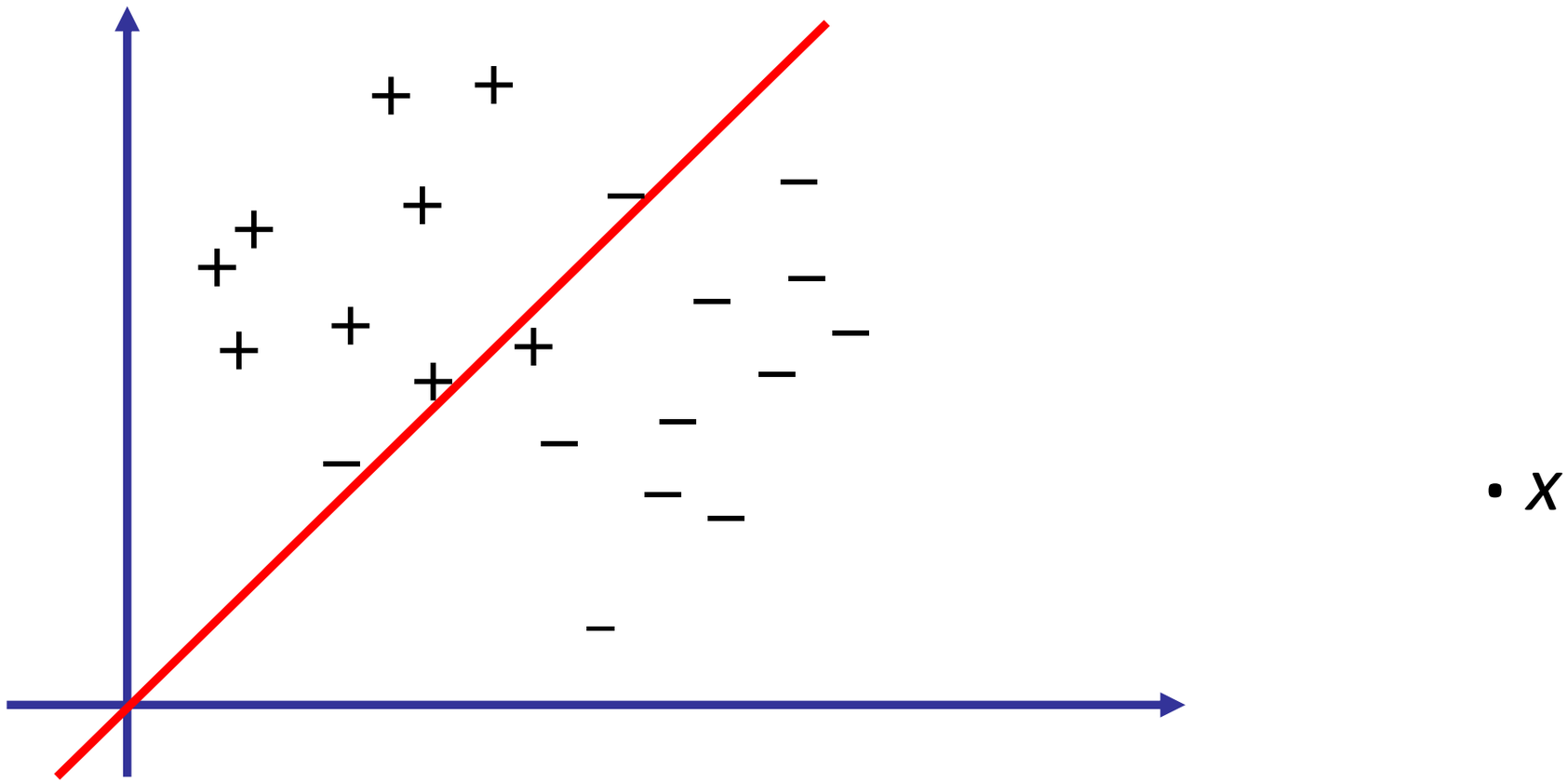
Motivating example

What will logistic regression predict for data point x?



Motivating example

What will logistic regression predict for data point x?



Logistic regression can be overconfident about labels for outliers

Discriminative vs generative modeling

Discriminative models aim to estimate conditional distribution

$$P(y \mid x)$$

Generative models aim to estimate joint distribution

$$P(y, x)$$

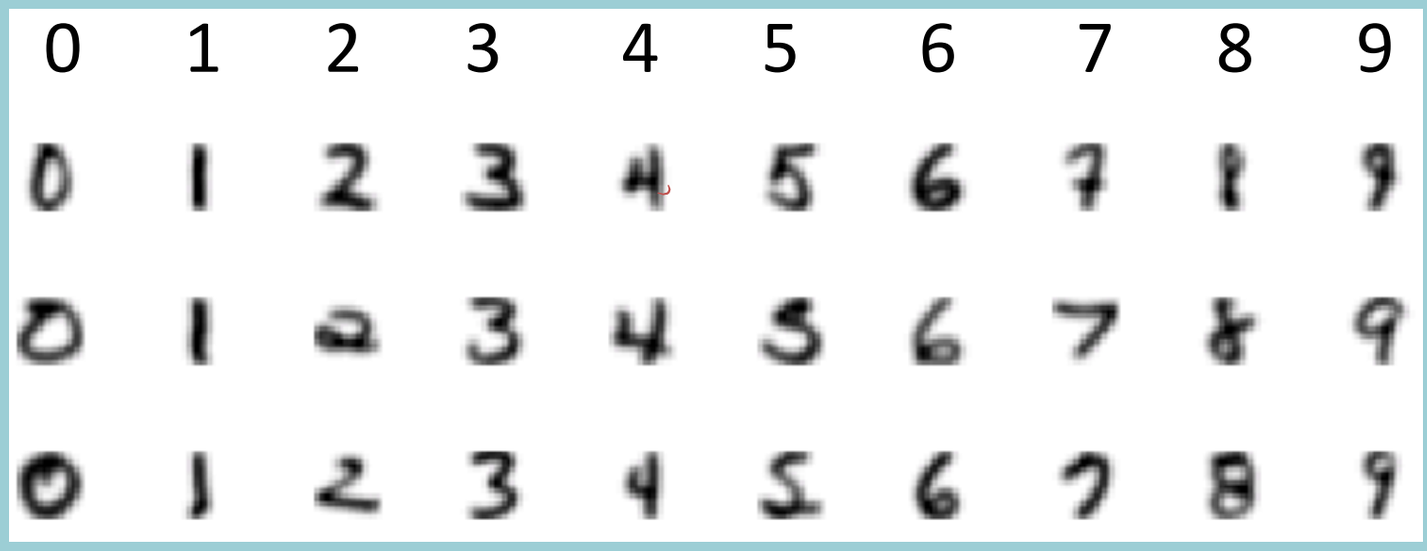
Can derive conditional from joint distribution, but not vice versa.

Typical approaches to generative modeling

1. Estimate prior on labels $P(y)$
2. Estimate conditional distribution $P(x \mid y)$ for each class y
3. Obtain predictive distribution using Bayes' rule:

$$P(y \mid x) = \frac{1}{Z} P(y) P(x \mid y)$$

Example: Handwritten digits



Naive Bayes model

Model class label as generated from categorical variable

$$P(Y = y) = p_y,\ y \in \mathcal{Y} = \{1, \ldots, c\}$$

Model features as conditionally independent given Y

$$P(X_{[1]}, \ldots, X_{[d]} \mid Y) = \prod_{i=1}^{d} P(X_{[i]} \mid Y)$$

- given class label, each feature is generated independently of the other features
- need to specify feature distribution $P(X_{[i]} \mid Y)$

Gaussian Naive Bayes classifiers (GNB)

Model class label as generated from categorical variable

$$P(Y = y) = p_y,\ y \in \mathcal{Y} = \{1, \ldots, c\}$$

Model features as conditionally independent Gaussians

$$P(X_{[1]}, \ldots, X_{[d]} \mid Y) = \prod_{i=1}^{d} P(X_{[i]} \mid Y)$$

$$P(x_{[i]} \mid y) = \mathcal{N}(x_{[i]} \mid \mu_{y,[i]}, \sigma^2_{y,[i]})$$

How do we estimate the parameters?

Maximum likelihood estimation for P(y)

$$\mathcal{Y} = \{-1, +1\} \qquad P(Y = +1) = p \qquad D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$$

Estimate $P(y)$ using D via MLE:

$$\max_p P(D \mid p) = \prod_{i=1}^{n} p^{[y_i=1]}(1-p)^{[y_i=-1]} = p^{n_+}(1-p)^{n_-}$$

where n_+ (resp. n_-) corresponds to the number of positive (resp. negative) instances in D. Therefore, the log-likelihood

$$\log P(D \mid p) = n_+ \log p + n_- \log(1-p)$$

Taking the gradient and set to 0, we get MLE for label distribution

$$\hat{p}_{\text{MLE}} = \frac{n_+}{n_+ + n_-} = \frac{\text{Count}(Y = +1)}{n}$$

Maximum likelihood estimation for P(x|y)

$$P(x_{[i]} \mid y) = \mathcal{N}(x_{[i]}; \mu_{y,[i]}, \sigma^2_{y,[i]}) \qquad D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$$

MLE for feature distribution:

$$\hat{\mu}_{y,[i]} = \frac{1}{\mathsf{Count}(Y = y)} \sum_{j:y_j=y} x_{j[i]}$$

$$\sigma^2_{y,[i]} = \frac{1}{\mathsf{Count}(Y = y)} \sum_{j:y_j=y} \left(x_{j[i]} - \hat{\mu}_{y,[i]}\right)^2$$

Decision rules

We have estimated $\hat{P}(y)$ and $\hat{P}(x \mid y)$. In order to predict label y for a new data point x, use the Bayes' rule

$$P(y \mid x) = \frac{1}{Z} P(y) P(x \mid y), \quad \text{where } Z = \sum_y P(y) P(x \mid y)$$

To minimize misclassification error, predict

$$y = \arg\max_{y_j} \hat{P}(y_j \mid x) = \arg\max_{y_j} \hat{P}(y_j) \prod_{i=1}^{d} \hat{P}(x_{[i]} \mid y_j)$$

Gaussian Naive Bayes classifiers

Learning given data $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$

- MLE for class label distribution

$$\hat{P}(Y = y) = \hat{p}_y = \frac{\text{Count}(Y = y)}{n}$$

- MLE for feature distribution:

$$\hat{P}(x_{[i]} \mid y) = \mathcal{N}(x_{[i]}; \hat{\mu}_{y[i]}, \sigma^2_{y[i]})$$

$$\hat{\mu}_{y[i]} = \frac{1}{\text{Count}(Y = y)} \sum_{j:y_j=y} x_{j[i]}$$

$$\hat{\sigma}^2_{y[i]} = \frac{1}{\text{Count}(Y = y)} \sum_{j:y_j=y} \left(x_{j[i]} - \hat{\mu}_{y[i]}\right)^2$$

Prediction given new point x:

$$y = \arg\max_{y_j} \hat{P}(y_j \mid x) = \arg\max_{y_j} \hat{P}(y_j) \prod_{i=1}^{d} \hat{P}(x_{[i]} \mid y_j)$$

Decision boundary (1D)

Example: Decision boundary (1D)

Assume $d = 1, x = x_{[1]}, \mathcal{Y} = \{-1, +1\}$, and $P(Y = +1) = 0.5$.
The decision boundary for a new point x is

$$y = \arg\max_{y_j} P(y_j \mid x) = \arg\max_{y_j} \hat{P}(y_j)\hat{P}(x \mid y_j)$$

$$= \arg\max_{y_j} P(x \mid y_j)$$

Decision rules for binary classification

We want to predict

$$y = \arg\max_{y_j} \hat{P}(y_j \mid x) = \arg\max_{y_j} \hat{P}(y_j) \prod_{i=1}^{d} \hat{P}(x_{[i]} \mid y_j)$$

For binary tasks (i.e. $c = 2, y \in \{-1, +1\}$), this is equivalent to

$$y = \text{sign} \underbrace{\left(\log \frac{P(Y = +1 \mid x)}{P(Y = -1 \mid x)} \right)}_{f(x)}$$

Discriminant function

The function $f(x) = \log \frac{P(Y=+1|x)}{P(Y=-1|x)}$ is called discriminant function.

Special case: GNB, c=2, class-invariant variance

Example: GNB ($c = 2$, class-invariant variance)

Assume

- binary classes: $\mathcal{Y} = \{-1, +1\}$
- class independent variance: $P(x \mid y) = \prod_i \mathcal{N}(x_{[i]}; \mu_{y,[i]}, \sigma^2_{[i]})$.

Then,

$$f(x) = \log \frac{P(Y = +1 \mid x)}{P(Y = -1 \mid x)} = w^\top x + b$$

where $w_{[i]} = \frac{\hat{\mu}_{+,[i]} - \hat{\mu}_{-,[i]}}{\hat{\sigma}^2_{[i]}}$, and $b = \log \frac{\hat{p}_+}{1-\hat{p}_+} + \sum_{i=1}^{d} \frac{\mu^2_{-,[i]} - \mu^2_{+,[i]}}{2\hat{\sigma}^2_{[i]}}$

How?

$$
\begin{aligned}
f(x) &= \log \frac{P(Y=+1 \mid x)}{P(Y=-1 \mid x)} = \log \frac{P(Y=+1)\prod_{i=1}^{d} P(x_{[i]} \mid Y=+1)/P(x)}{P(Y=-1)\prod_{i=1}^{d} P(x_{[i]} \mid Y=-1)/P(x)} \\
&= \log \frac{\hat{p}_+}{1-\hat{p}_+} + \log \prod_{i=1}^{d} \frac{P(x_{[i]} \mid Y=+1)}{P(x_{[i]} \mid Y=-1)} \\
&= \log \frac{\hat{p}_+}{1-\hat{p}_+} + \log \prod_{i=1}^{d} \frac{\frac{1}{\sqrt{2\pi}\sigma_{[i]}} \exp\left(-\frac{1}{2\sigma_{[i]}^2}\left(x_{[i]}-\mu_{+1,[i]}\right)^2\right)}{\frac{1}{\sqrt{2\pi}\sigma_{[i]}} \exp\left(-\frac{1}{2\sigma_{[i]}^2}\left(x_{[i]}-\mu_{-1,[i]}\right)^2\right)} \\
&= \log \frac{\hat{p}_+}{1-\hat{p}_+} + \sum_{i=1}^{d} \left(-\frac{1}{2\sigma_{[i]}^2}\left(x_{[i]}-\mu_{+1,[i]}\right)^2 + \frac{1}{2\sigma_{[i]}^2}\left(x_{[i]}-\mu_{-1,[i]}\right)^2\right) \\
&= \sum_{i=1}^{d} \underbrace{\left(\frac{\hat{\mu}_{+,[i]}-\hat{\mu}_{-,[i]}}{\hat{\sigma}_{[i]}^2}\right)}_{w_{[i]}} x_{[i]} + \underbrace{\log \frac{\hat{p}_+}{1-\hat{p}_+} + \sum_{i=1}^{d} \frac{\mu_{-,[i]}^2 - \mu_{+,[i]}^2}{2\hat{\sigma}_{[i]}^2}}_{b}
\end{aligned}
$$

Gaussian NB (c=2): f vs. class probability

$$f(x) = \log \frac{P(Y = +1 \mid x)}{P(Y = -1 \mid x)}$$

$$\Leftrightarrow \;\; P(Y = +1 \mid x) = \frac{1}{1 + \exp\left(-f(x)\right)} = \sigma(f(x))$$

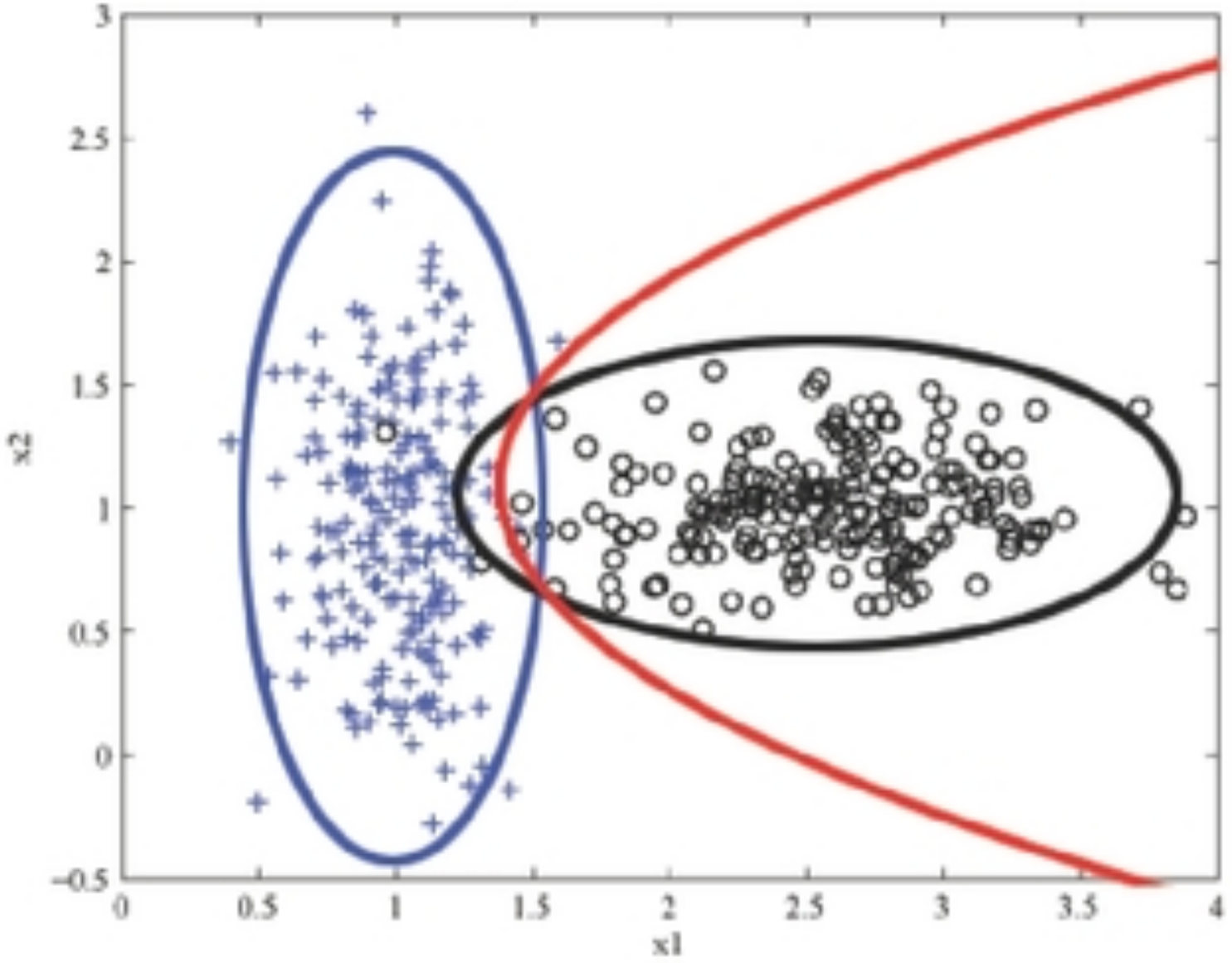
Therefore, the class probability of 2-class GNB with class independent variance is

$$P(Y = +1 \mid x) = \sigma(w^\top x + b)$$

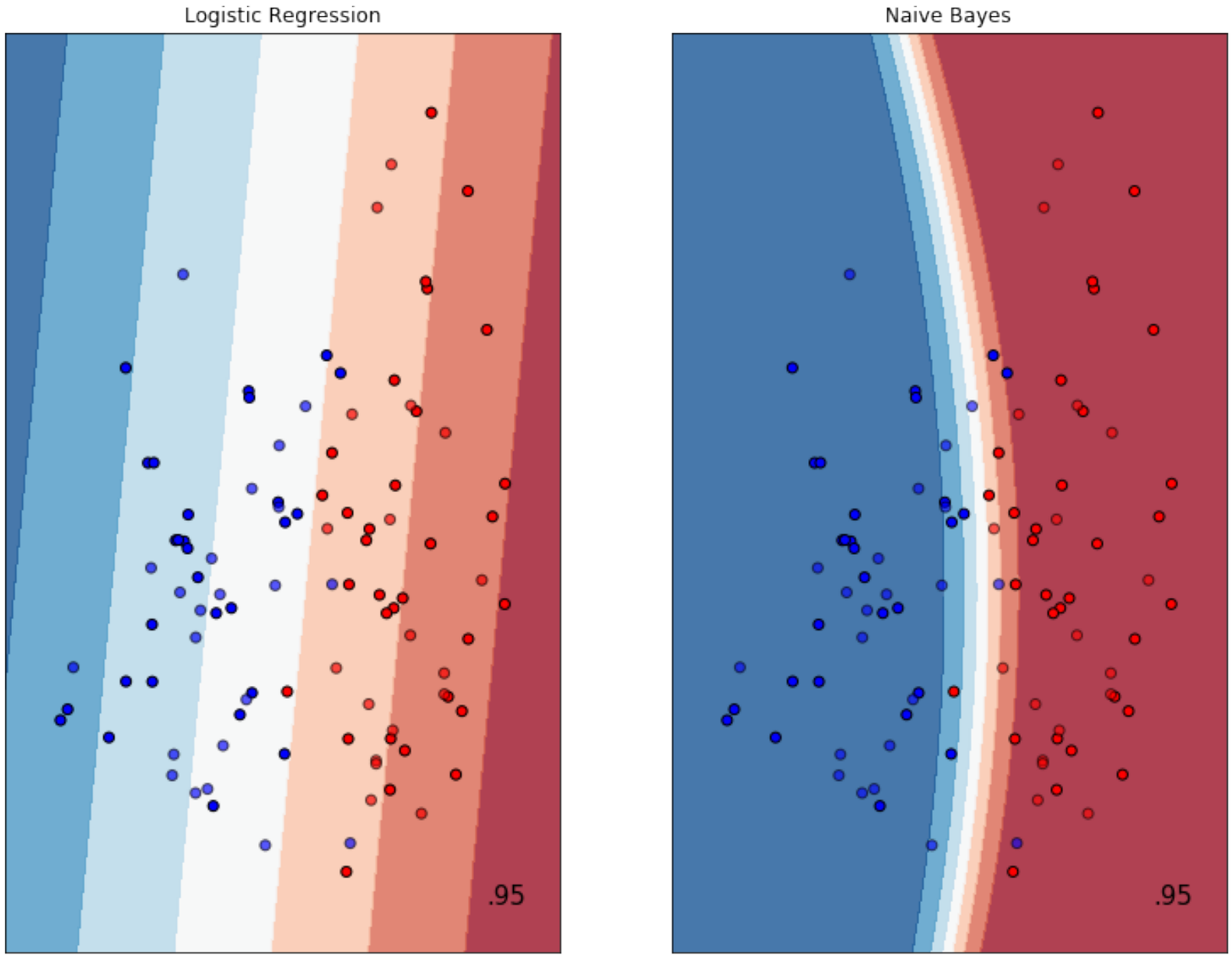
This is of the same form as logistic regression.

If model assumptions are met, GNB will make same predictions as Logistic Regression!

Gaussian NB (c=2): decision boundary

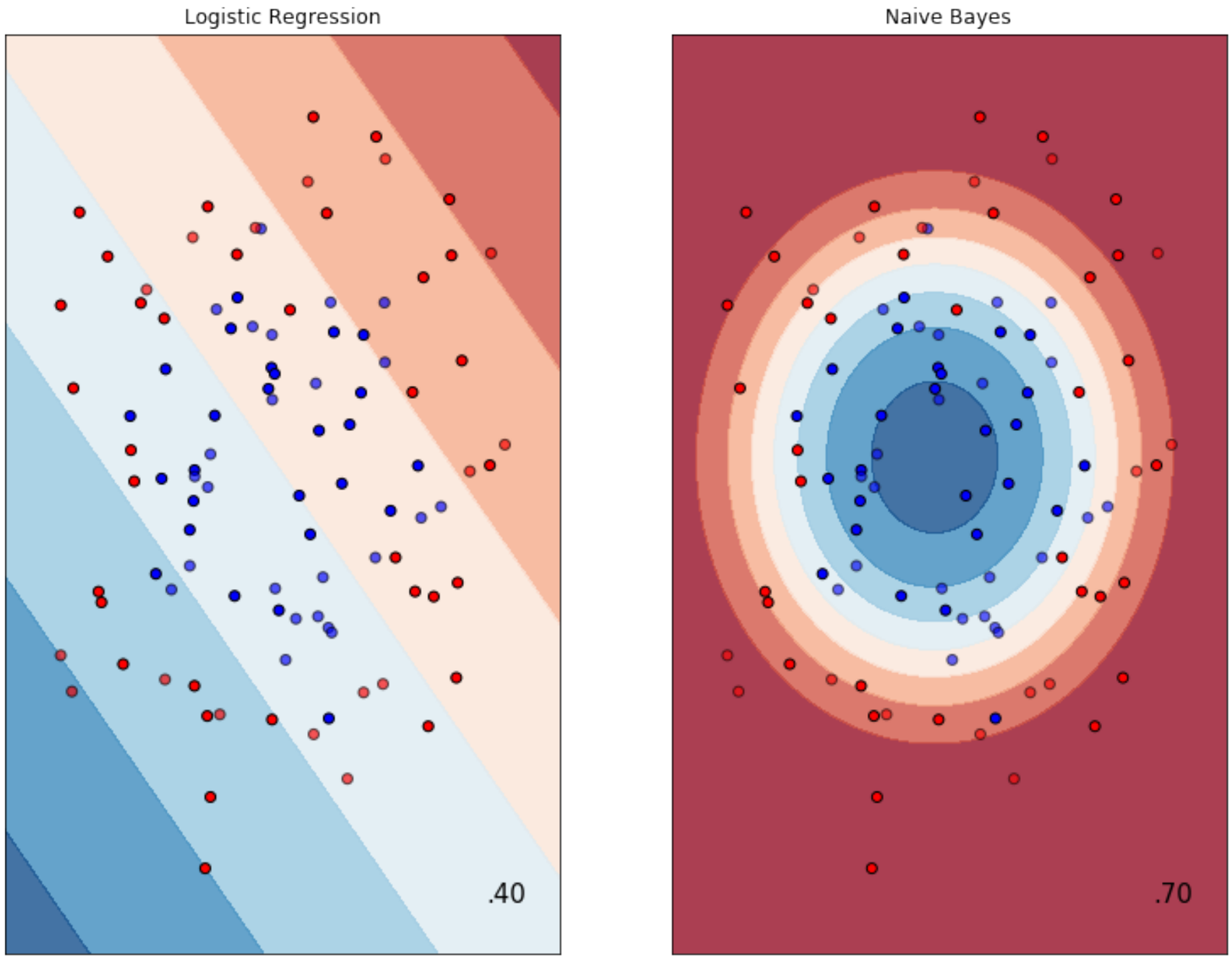


Demo: Gaussian NB vs LR (linear)



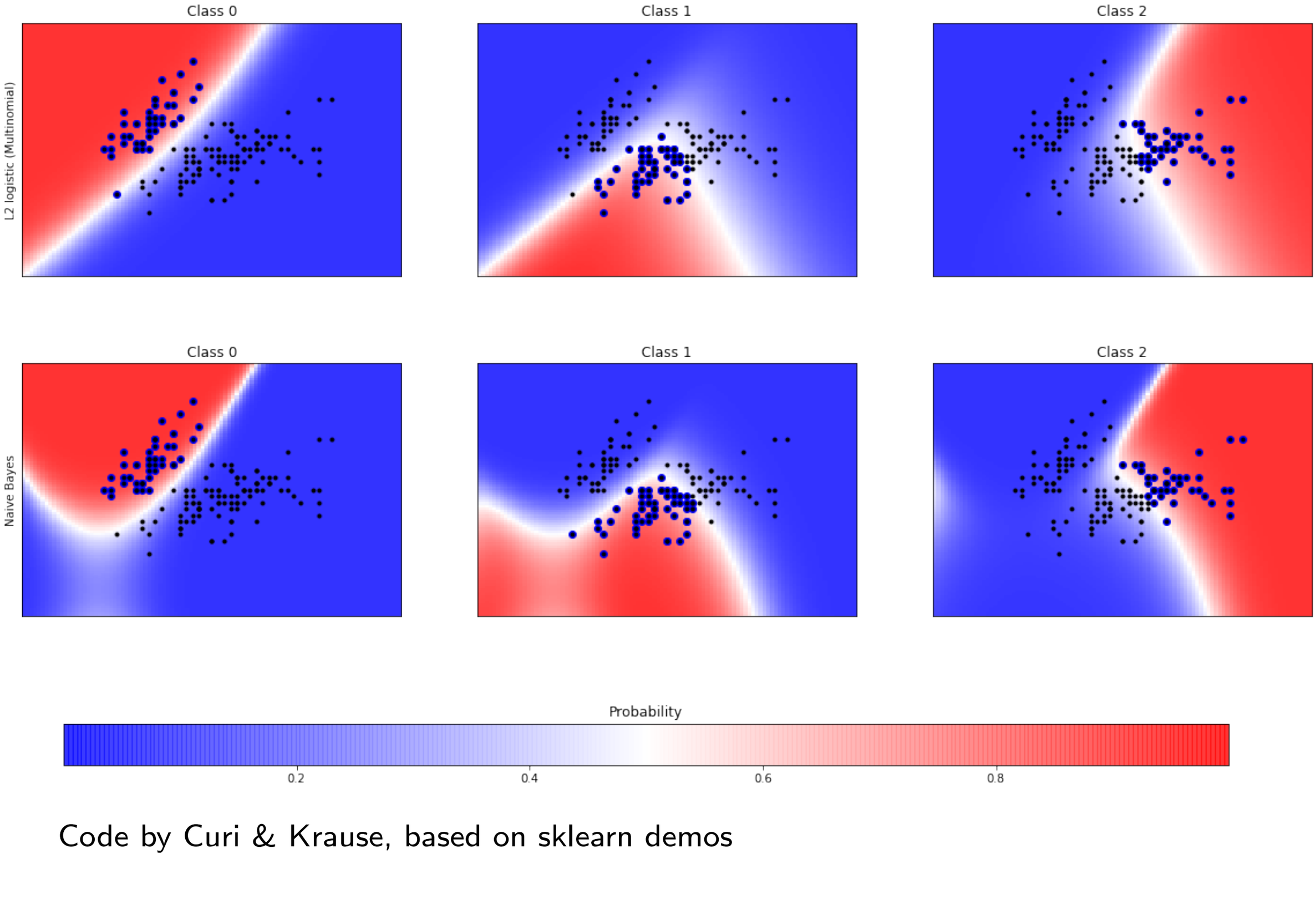
Code by Curi & Krause, based on sklearn demos

Demo: Gaussian NB vs LR (circle)



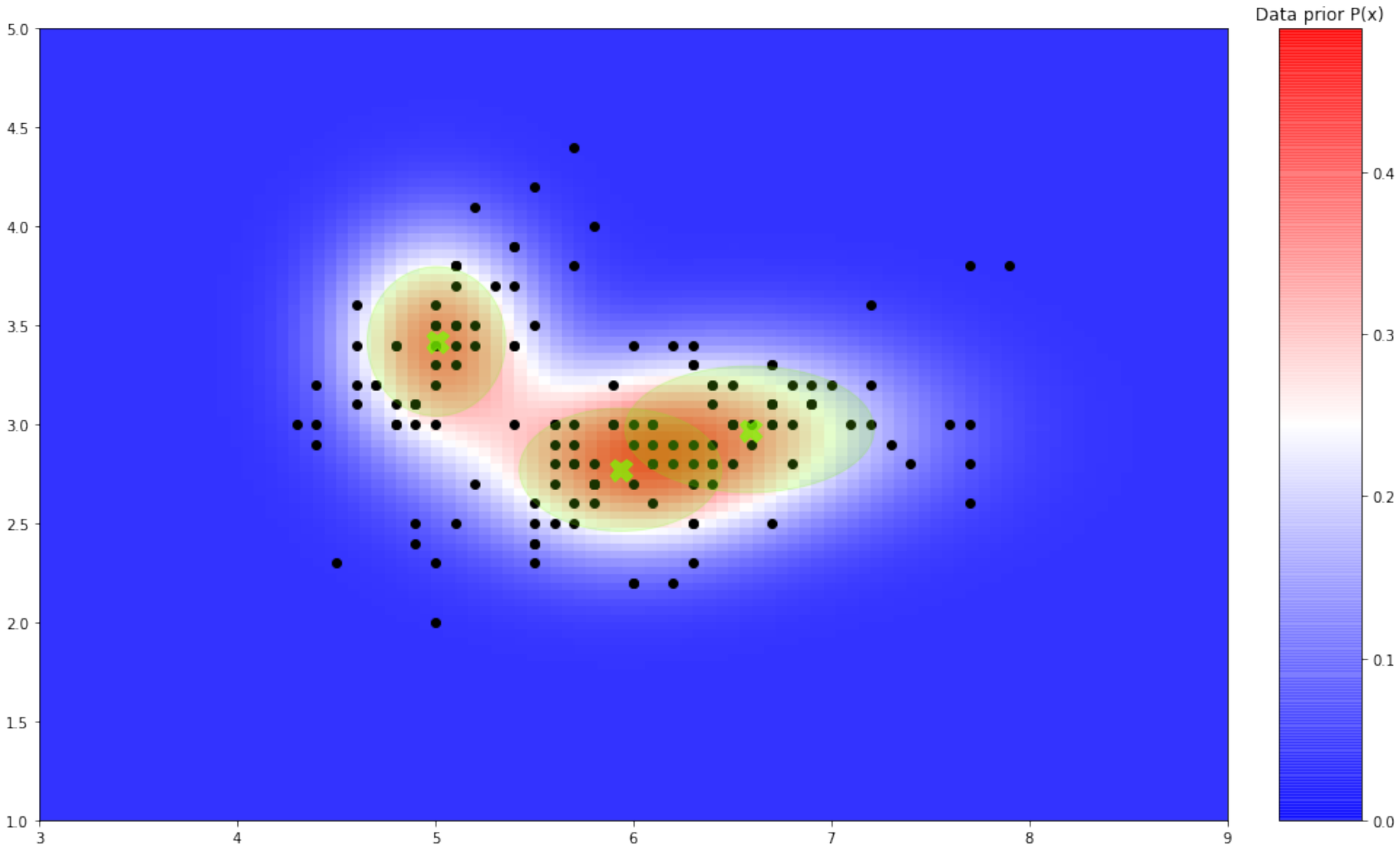
Code by Curi & Krause, based on sklearn demos

Demo: Gaussian NB vs LR (multi-class)



Code by Curi & Krause, based on sklearn demos

Demo: Gaussian NB (data prior)



Code by Curi & Krause, based on sklearn demos

Limitation of Naive Bayes models

Example: dependent features

Suppose $\mathcal{Y} = \{-1, +1\}$, and $x_{[1]} = \cdots = x_{[d]}, \forall x \in \mathcal{X}$. Assume $P(Y = +1) = 0.5$ and $P(X_{[i]} = x \mid y) = \mathcal{N}(x \mid \mu_y, 1)$. We consider the discriminant function for two GNB variants:

1. For GNB that only uses $X_{[1]}$: $f_1(x) = \log \frac{P(Y=+1|X_{[1]}=x)}{P(Y=-1|X_{[1]}=x)}$
2. For GNB that uses $X_{[1]}, \ldots, X_{[d]}$:

$$
\begin{aligned}
f_2(x) &= \log \frac{P(Y = +1 \mid X_{[1]} = x, \ldots, X_{[d]} = x)}{P(Y = -1 \mid X_{[1]} = x, \ldots, X_{[d]} = x))} \\
&= \log \frac{P(X_{[1]} = x, \ldots, X_{[d]} = x \mid Y = +1)}{P(X_{[1]} = x, \ldots, X_{[d]} = x \mid Y = -1))} \\
&= \log \prod_{i=1}^{d} \frac{P(X_{[i]} = x \mid Y = +1)}{P(X_{[i]} = x, \mid Y = -1))} = d \cdot f_1(x)
\end{aligned}
$$

Due to cond. ind. assumption, predictions can be overconfident

Gaussian Bayes classifiers (GBC)

Model class label as generated from categorical variable

$$P(Y = y) = p_y, \; y \in \mathcal{Y} = \{1, \ldots, c\}$$

Model features as multivariate Gaussians

$$P(x \mid y) = \mathcal{N}(x; \mu_y, \Sigma_y)$$

Example: Gaussian Naive Bayes

GNB is special case, with $\Sigma_y = \text{diag}\left(\sigma^2_{y,[1]}, \ldots, \sigma^2_{y,[d]}\right)$

How do we estimate the parameters?

MLE for Gaussian Bayes classifier

Given data $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$

- MLE for class label distribution

$$\hat{P}(Y = y) = \hat{p}_y = \frac{\text{Count}(Y = y)}{n}$$

- MLE for feature distribution:

$$\hat{P}(x \mid y) = \mathcal{N}(x; \hat{\mu}_y, \hat{\Sigma}_y^2)$$

$$\hat{\mu}_y = \frac{1}{\text{Count}(Y = y)} \sum_{j:y_j=y} x_j$$

$$\hat{\Sigma} y = \frac{1}{\text{Count}(Y = y)} \sum_{j:y_j=y} (x_j - \hat{\mu}_y)(x_j - \hat{\mu}_y)^\top$$

Discriminant functions for GBC

Given $P(Y = +1) = p_+$ and $P(x \mid y) = \mathcal{N}(x; \mu_y, \Sigma_y)$, the discriminant function for GBC is given by

$$f(x) = \log \frac{P(Y = +1 \mid x)}{P(Y = -1 \mid x)}$$

$$= \log \frac{p_+}{1 - p_+} + \frac{1}{2} \log \frac{\left|\hat{\Sigma}_-\right|}{\left|\hat{\Sigma}_+\right|} +$$

$$\frac{1}{2} \left[\left((x - \hat{\mu}_-)^\top \hat{\Sigma}_-^{-1} (x - \hat{\mu}_-) \right) - \left((x - \hat{\mu}_+)^\top \hat{\Sigma}_+^{-1} (x - \hat{\mu}_+) \right) \right]$$

Example: GNB case (with shared variance)

$$f(x) = \log \frac{p_+}{1 - p_+} + \sum_{i=1}^{d} \left(-\frac{1}{2\sigma_{[i]}^2} \left(x_{[i]} - \mu_{+1,[i]} \right)^2 + \frac{1}{2\sigma_{[i]}^2} \left(x_{[i]} - \mu_{-1,[i]} \right)^2 \right)$$

Fisher's linear discriminant analysis (LDA), c=2

Suppose we fix $p_+ = 0.5$.

Further, assume covariances are equal: $\Sigma_+ = \Sigma_- = \Sigma$.

Then the discriminant function for GBC could be simplified as

$$f(x) = \log \frac{p_+}{1-p_+} + \frac{1}{2}\left[\log \frac{\left|\hat{\Sigma}_-\right|}{\left|\hat{\Sigma}_+\right|} + \left((x-\hat{\mu}_-)^\top \hat{\Sigma}_-^{-1}(x-\hat{\mu}_-)\right) - \left((x-\hat{\mu}_+)^\top \hat{\Sigma}_+^{-1}(x-\hat{\mu}_+)\right)\right]$$

$$= \frac{1}{2}\left[\left((x-\hat{\mu}_-)^\top \hat{\Sigma}^{-1}(x-\hat{\mu}_-)\right) - \left((x-\hat{\mu}_+)^\top \hat{\Sigma}^{-1}(x-\hat{\mu}_+)\right)\right]$$

$$= x^\top \hat{\Sigma}^{-1}(\hat{\mu}_+ - \hat{\mu}_-) + \frac{1}{2}\left(\hat{\mu}_-^\top \hat{\Sigma}^{-1}\hat{\mu}_- - \hat{\mu}_+^\top \hat{\Sigma}^{-1}\hat{\mu}_+\right)$$

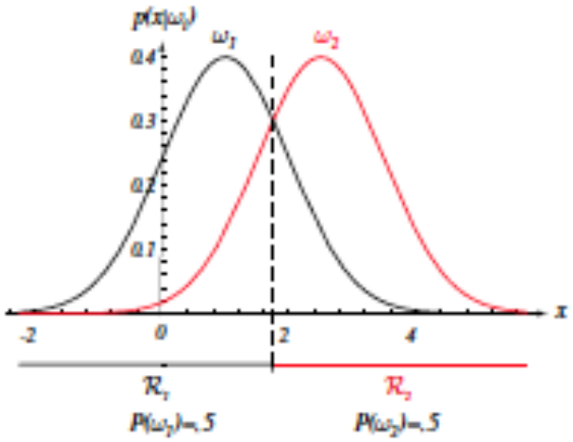
Fisher's linear discriminant analysis (Fisher's LDA)

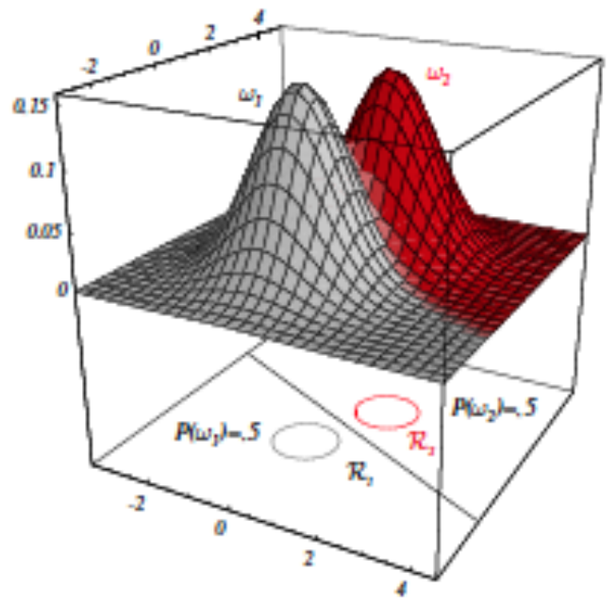
Under the above assumptions, Fisher's LDA predicts

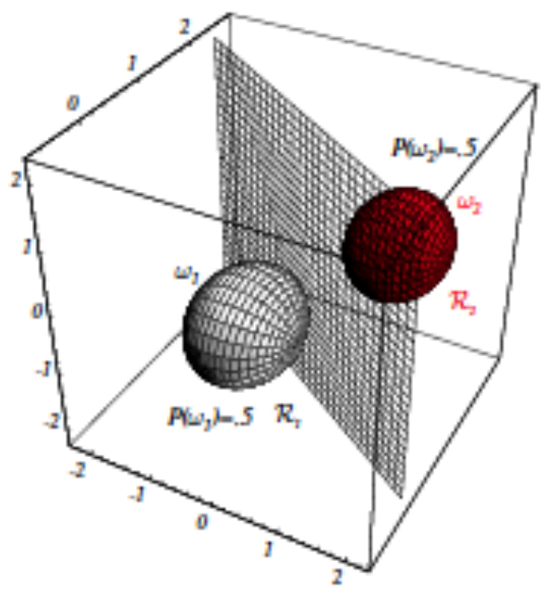
$$y = \text{sign}(f(x)) = \text{sign}(w^\top x + b)$$

where $w = \hat{\Sigma}^{-1}(\hat{\mu}_+ - \hat{\mu}_-)$, and $b = \frac{1}{2}\left(\hat{\mu}_-^\top \hat{\Sigma}^{-1}\hat{\mu}_- - \hat{\mu}_+^\top \hat{\Sigma}^{-1}\hat{\mu}_+\right)$

LDA Example


p(x|ω_j)
ω_1
ω_2
0.4
0.3
0.2
0.1
-2
0
2
4
x
R_1
P(ω_1)=.5
R_2
P(ω_2)=.5


ω_1
ω_2
0.15
0.1
0.05
0
-2
0
2
4
P(ω_1)=.5
P(ω_2)=.5
R_1
R_2


P(ω_2)=.5
ω_2
ω_1
R_2
P(ω_1)=.5
R_1
2
1
0
-1
-2

LDA vs. logistic regression

Fisher's LDA uses the discriminant function

$$f(x) = \log \frac{P(Y = +1 \mid x)}{P(Y = -1 \mid x)} := w^\top x + b$$

$$\Leftrightarrow \;\; P(Y = +1 \mid x) = \frac{1}{1 + \exp\left(-f(x)\right)} = \sigma(f(x))$$

Therefore, the class probability of LDA is

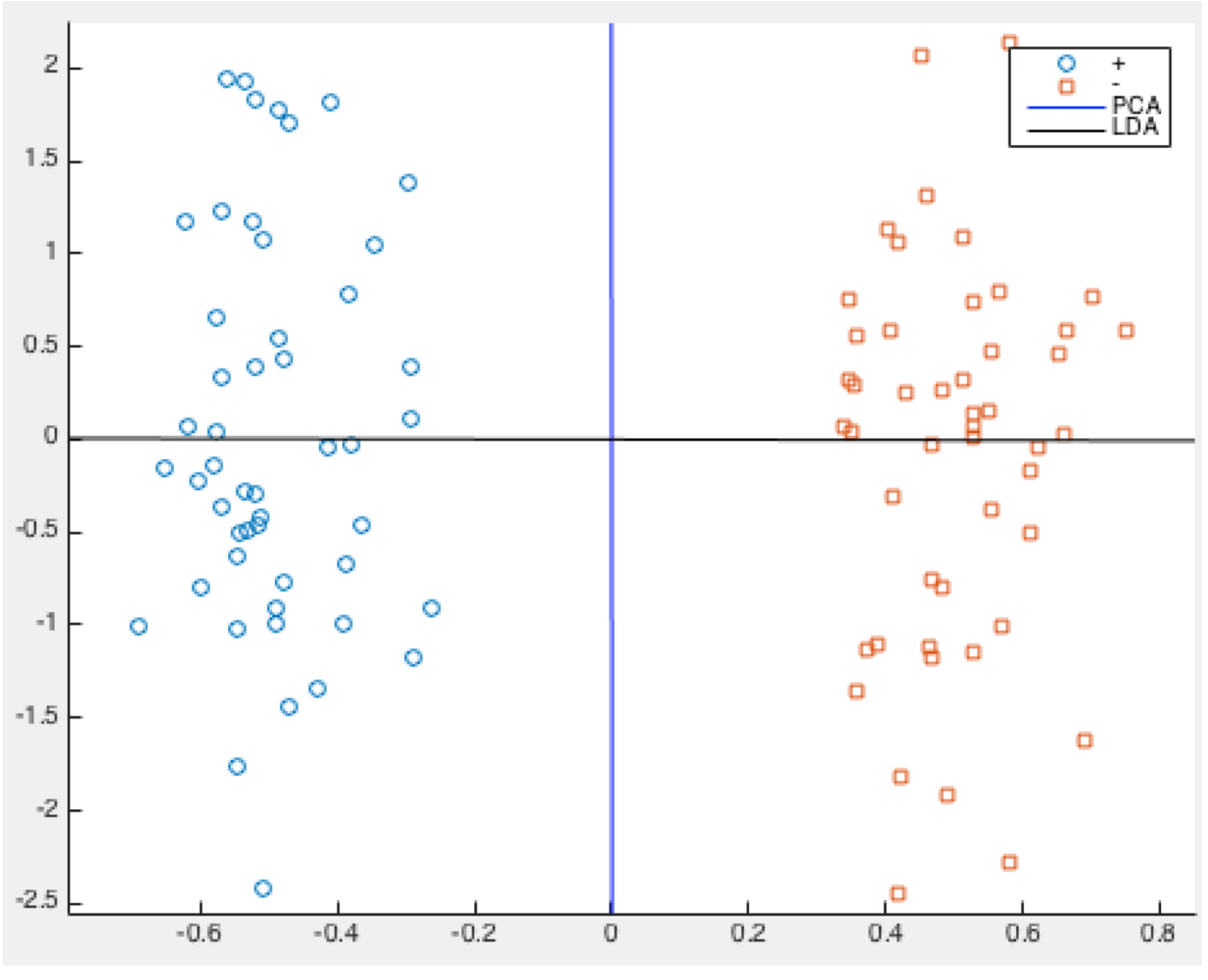
$$P(Y = +1 \mid x) = \sigma(w^\top x + b)$$

This is of the same form as logistic regression.

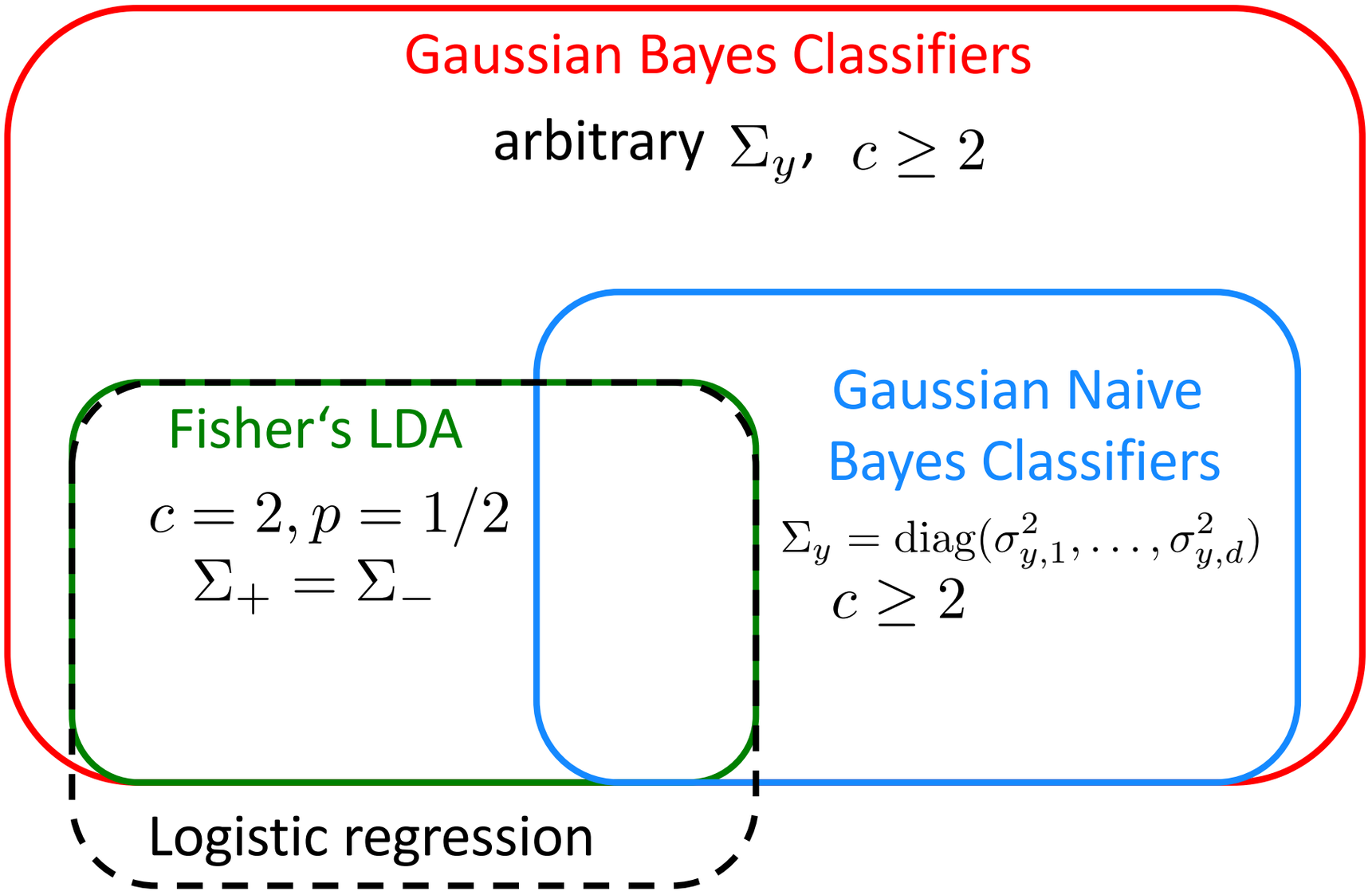
If model assumptions are met, LDA will make same predictions as Logistic Regression!

LDA vs. PCA

- LDA attempts to find a 1-dim subspace that maximizes class separability, i.e. ratio of between and within class variances.
- PCA (1 component) maximizes the variance of the resulting 1-dim projection.



Gaussian Bayes classifiers: the big picture



Fisher's LDA vs logistic regression

Fisher's LDA

- Generative model, i.e., models $P(X, Y)$
- Can be used to detect outliers: $P(X) < t$
- Assumes normality of X
- not very robust against violation of this assumption

Logistic regression

- Discriminative model, i.e., models $P(Y \mid X)$ only
- Cannot detect outliers
- Makes no assumptions on X
- More robust

GNB vs GBC

Gaussian Naive Bayes models

- Conditional independence assumption may lead to overconfidence
- Predictions might still be useful
- #parameters $= O(cd)$
- Complexity (memory + inference) linear in d

General Gaussian Bayes models

- Captures correlations among features
- Avoids overconfidence
- #parameters $= O(cd^2)$
- Complexity quadratic in d

Avoid overfitting

Maximum Likelihood Estimation is prone to overfitting. We can avoid over fitting by

- Restricting model class (e.g., assumptions on covariance structure, e.g., Gaussian Naive Bayes). This leads to fewer parameters
- Using priors, which often leads to "smaller" parameters

Prior over parameters (c=2)

- As prior for our class probabilities, have assumed $P(Y = 1) = \theta$
- Maximum likelihood estimate: $\hat{\theta} = \frac{\text{Count}(Y=1)}{n}$
- What happens in the extreme case $n = 1$?

Prior over parameters (c=2)

- As prior for our class probabilities, have assumed $P(Y = 1) = \theta$
- Maximum likelihood estimate: $\hat{\theta} = \frac{\text{Count}(Y=1)}{n}$
- What happens in the extreme case $n = 1$?
- May want to put prior distribution $P(\theta)$ and compute posterior distribution $P(\theta \mid y_1, \ldots, y_n)$.

Prior over parameters (c=2)

- As prior for our class probabilities, have assumed $P(Y = 1) = \theta$
- Maximum likelihood estimate: $\hat{\theta} = \frac{\text{Count}(Y=1)}{n}$
- What happens in the extreme case $n = 1$?
- May want to put prior distribution $P(\theta)$ and compute posterior distribution $P(\theta \mid y_1, \ldots, y_n)$.

Example: Beta prior over parameters

$$\text{Beta}(\theta; \alpha_+, \alpha_-) = \frac{1}{B(\alpha_+, \alpha_-)} \theta^{\alpha_+ - 1} (1 - \theta)^{\alpha_- - 1}$$

Recall: conjugate distributions

A pair of prior distributions and likelihood functions is called conjugate if the posterior distribution remains in the same family as the prior.

Example: Beta priors and Binomial likelihood

- Prior: Beta$(\theta; \alpha_+, \alpha_-)$
- Observations: supose we have n_+ positive and n_- negative labels
- Posterior: Beta$(\theta; \alpha_+ + n_+, \alpha_- + n_-)$

Therefore α_+ and α_- act as pseudo-counts. The MAP estimate is

$$\hat{\theta} = \arg\max_\theta P(\theta \mid y_1, \ldots, y_n; \alpha_+, \alpha_-) = \frac{\alpha_+ + n_+ - 1}{\alpha_+ + n_+ + \alpha_- + n_- - 2}$$

Generative vs discriminative modeling

Discriminative models

- model $P(y|x)$. Do not attempt to model $P(x)$
- Cannot detect outliers (property of $P(x)$)
- are typically more robust, since accurately modeling x may be difficult

Generative models

- model joint distribution $P(x, y)$
- can be more powerful (e.g., detect outliers) if model assumptions are met
- are typically less robust against outliers

Summary

What you should be able to do

- Understand the connection between discriminative and generative classification
- Apply Naive Bayes classifiers
- Relate different Gaussion Bayes classifier (GNB, LDA, GBC)
- Use priors as regularizers
- Compute distributions over features, and use them for outlier detection

Acknowledgement Slides for this lecture are built upon the Generative Modeling lecture notes by Prof. Andreas Krause (ETH Zurich) as part of the Intro to ML course. Demo (Python code) accessible at `https://bit.ly/2Sd9iL1`