SUPERVISED LEARNING

CLASSIFICATION

Keith Burghardt

USC Information Sciences Institute

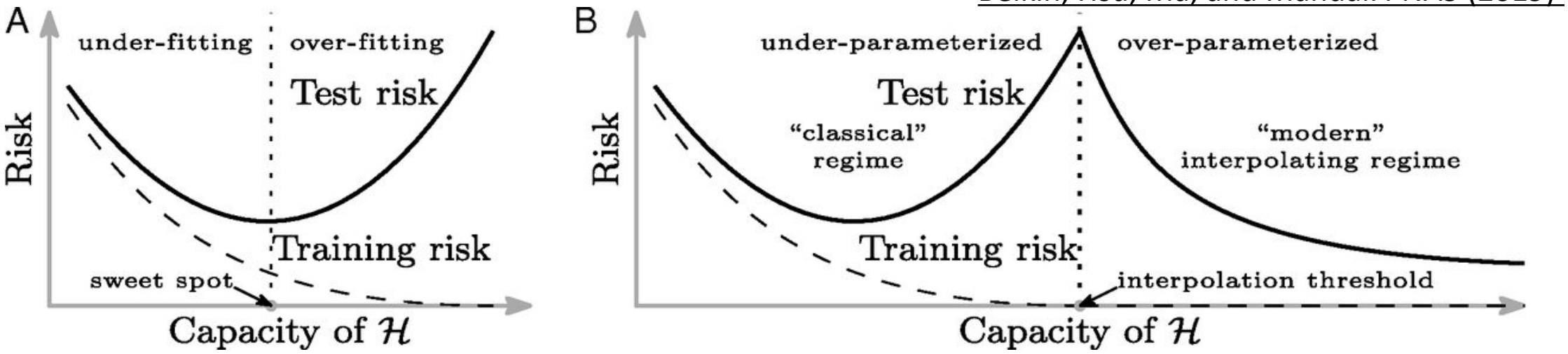
DSCI 552 – Spring 2021

Topics this week

- **Reminder:**
  - Quiz 2 due Wednesday
  - Quiz 3 will be posted Wednesday
  - **Literature review and Quiz 3 are due February 10 at 10 a.m. PT (next Wednesday)**
- Supervised classification
  - Bayesian inference
  - Parametric models (ex: logistic regression)
  - K-Nearest Neighbo(u)rs
- Performance measures

Bias-Variance Curve Redux

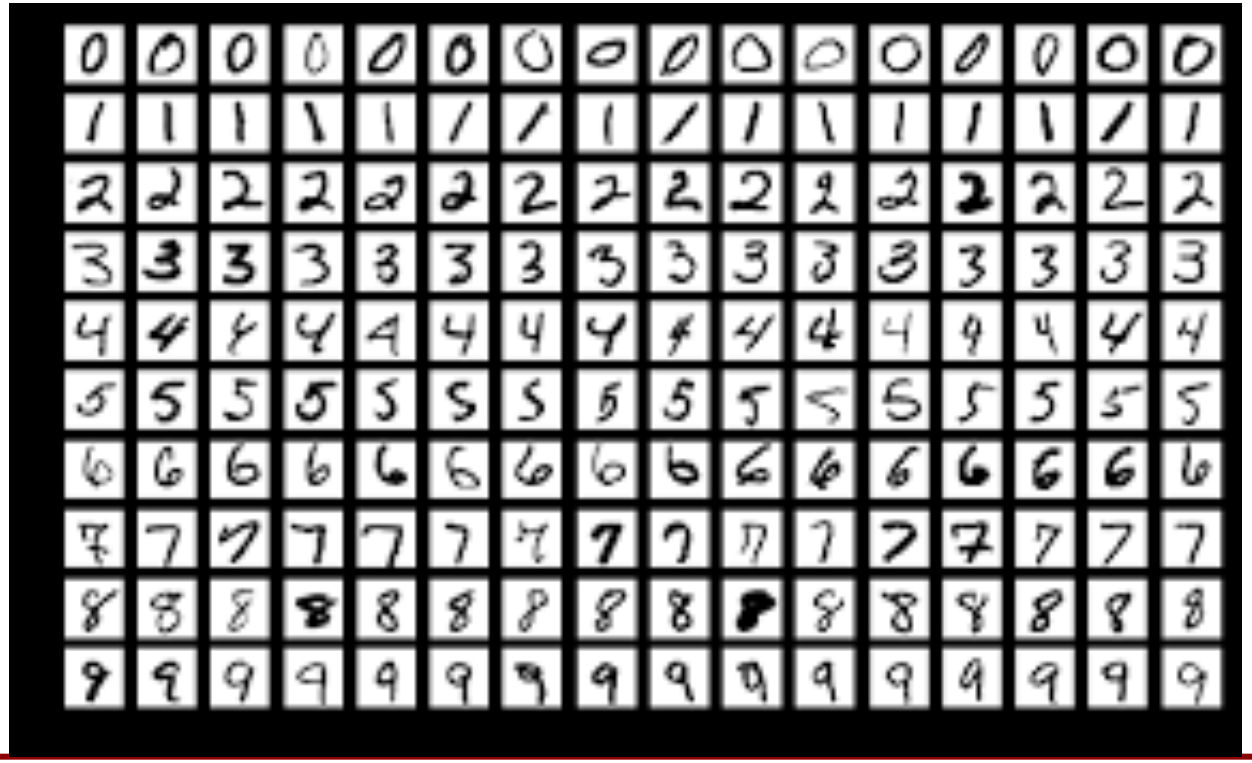
Belkin, Hsu, Ma, and Mandal. *PNAS* (2019)



- Until 2019, the left-hand description was standard
- RQ: Why do neural networks *work*?
  - Old models = Dozens of parameters
  - NNs: thousands to TRILLIONS of parameters
- Empirically found that non-linear methods *improve* with parameters

Example: MNIST

- Train on 48,000 images; test on 28,000 images
- 32 x 32 pixels
- We can exactly fit every pixel with 43M parameters



VGG19

Accuracy: 99.5%

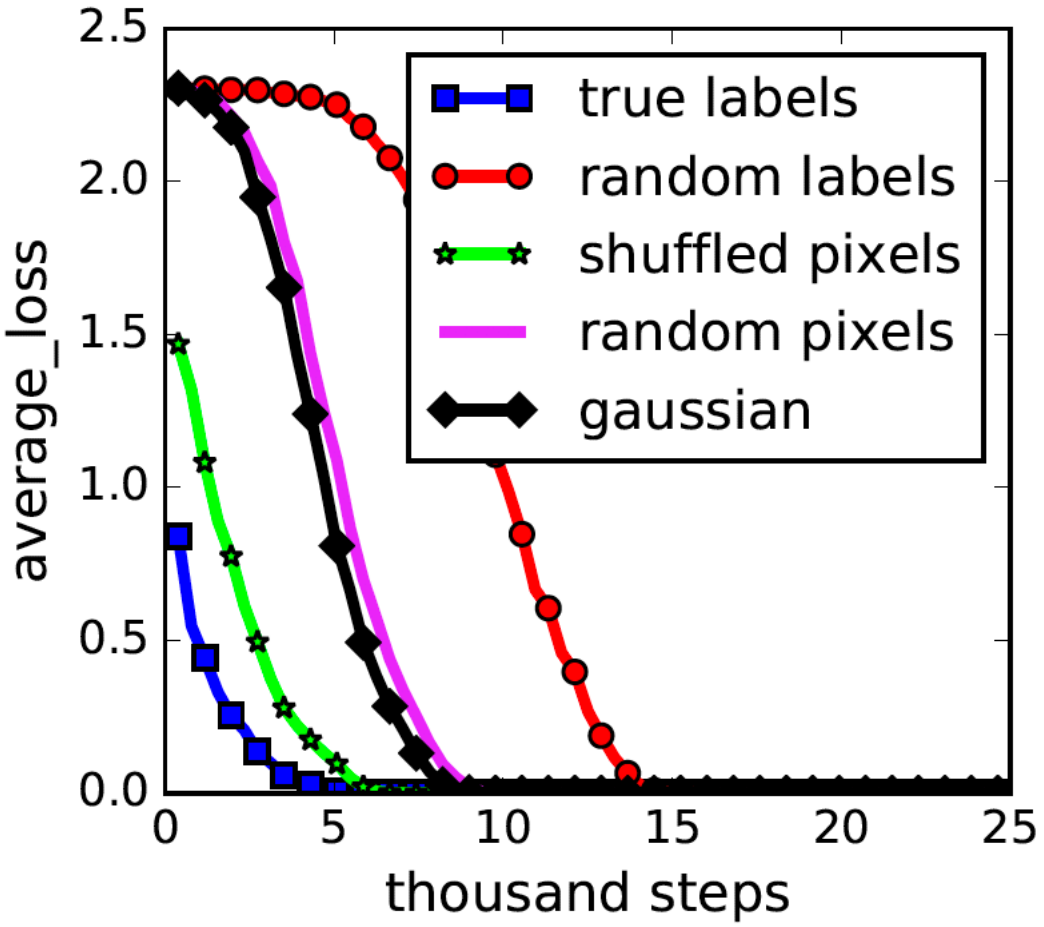
Parameters: 20M

Only 2 pixels/parameter(!)

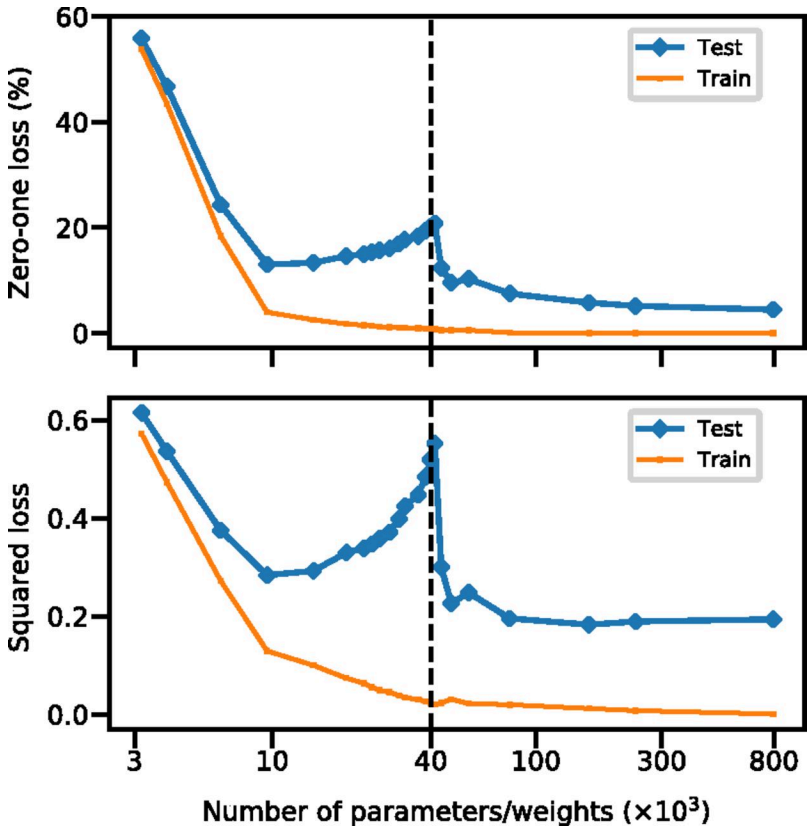
https://www.kaggle.com/muerbingsha/mnist-vgg19

Training on Random Labels?

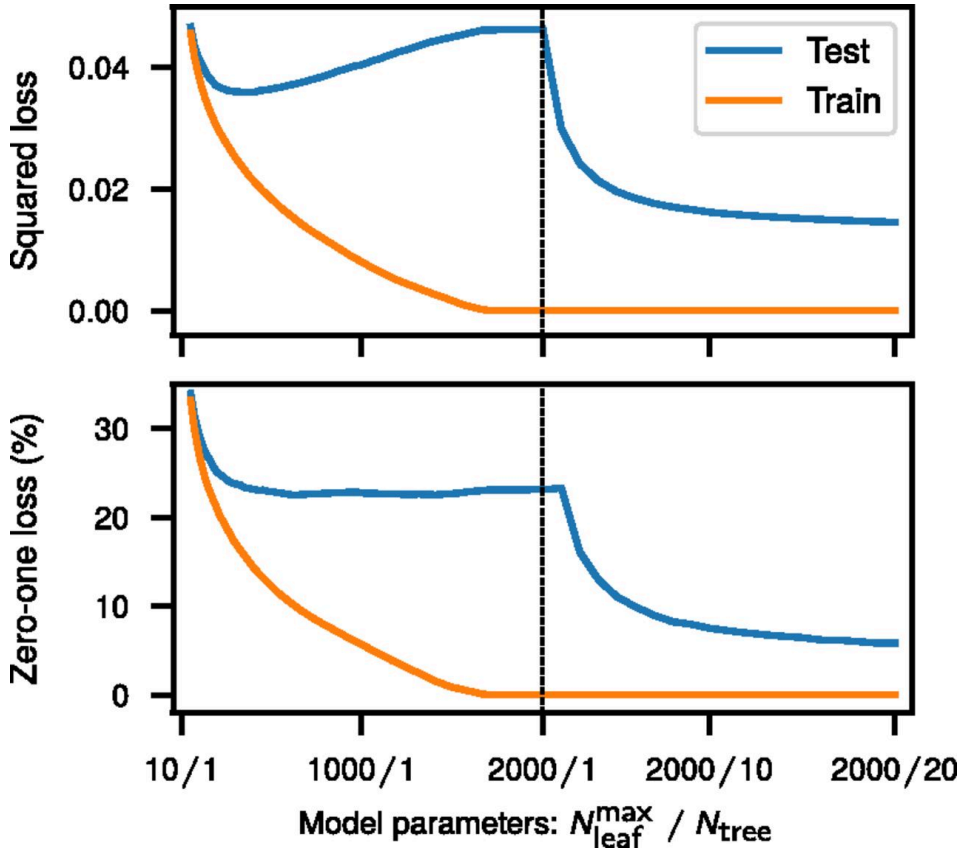
- Zhang et al., (2017) asked: what if we train on random labels?
- Neural networks can effectively memorize training data!
- They did not know why a neural network could “overfit” yet work so well



Examples on “MNIST”

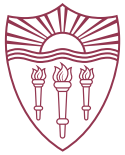


Neural network



Random Forest

More Parameters = Better?



- Yes and no
- For many models/datasets, yes
- Imagine we have 3 datapoints, should we fit a 1000-parameter model on the data? No!
- Clearly, there is a limit to when this is correct
- Nonetheless, keep in mind: don't reduce the number of parameters just for the sake of it
- Use validation/testing to infer the best model for the job

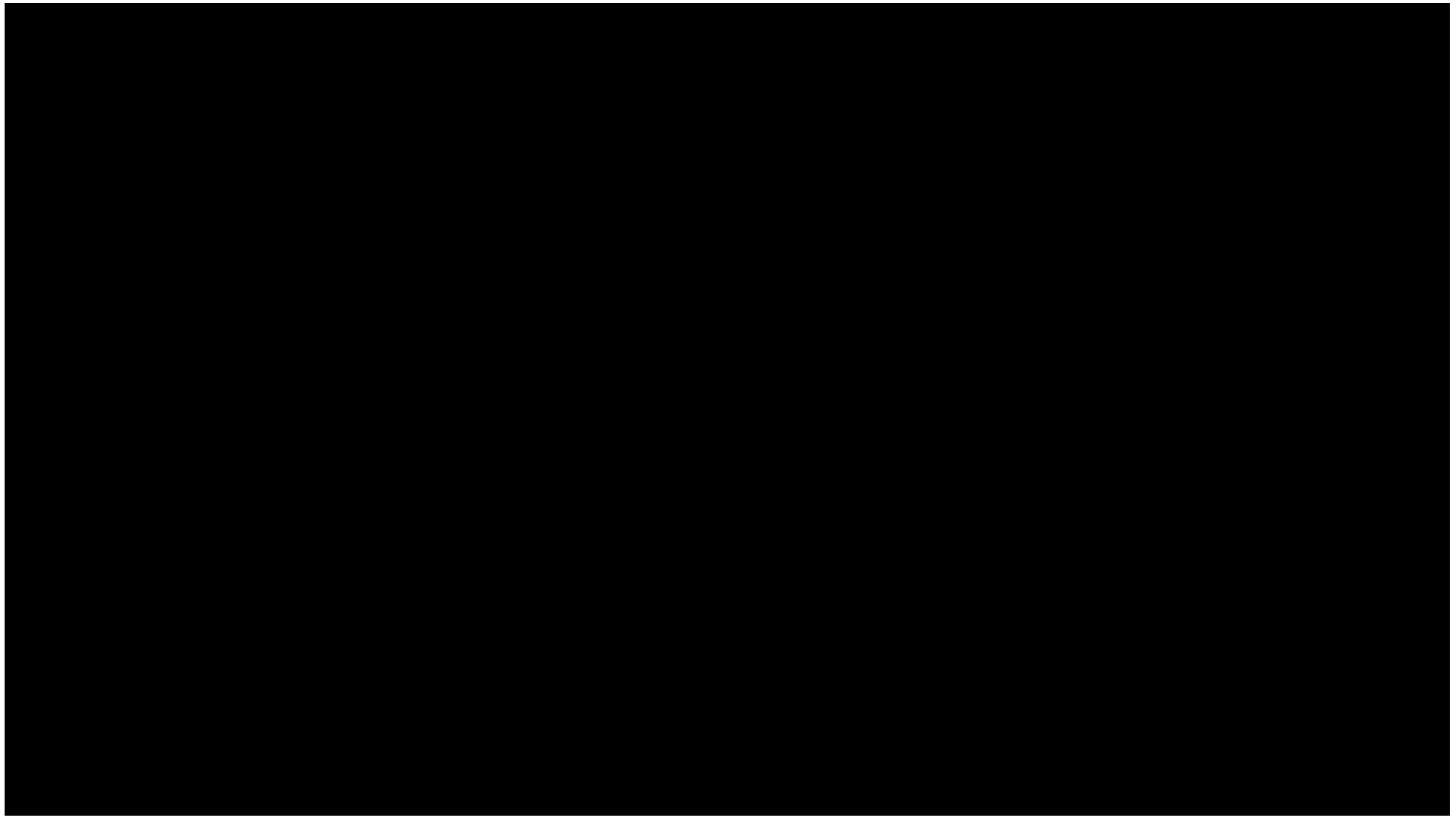
CLASSIFICATION MODELS

What we mean by classification

- The prior paper has real life examples, e.g., online check deposits:
  - Apps allow you to deposit a check by just taking a picture
  - Why can't you tell bank the check was for "$1900" instead of "$10.00"?
- Deposit checks at scale: banks need neural network to
  - distinguish "9" from "0", and
  - "." from smudge
- Classification: sort data into classes, e.g., 0,1,2,…
  - Supervised (learn what "0" means)
  - Unsupervised (find distinct clusters)
- Today, we will discuss how supervised classification works

BAYESIAN INFERENCE

Let’s simplify the problem: coin toss

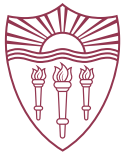


2:40

“Call it”: What would you predict?

- Scenario:
  - You take a break from ML, and decide to run a gas station in Texas
  - Anton Chigurh arrives and asks you to guess the coin toss
  - You (correctly) suspect that losing the coin toss is not in your best interest
- What do you predict?
- If you never saw the coin before, you may assume it is “fair”
  - We call this the “*prior*”
- Given you saw the coin tossed a few times: {H,T,H,H,H,H,T}
  - We call this the “*evidence*”
- You calculate the *likelihood* of 2T and 5H given your estimated probability of heads, p

This is Bayesian Statistics!

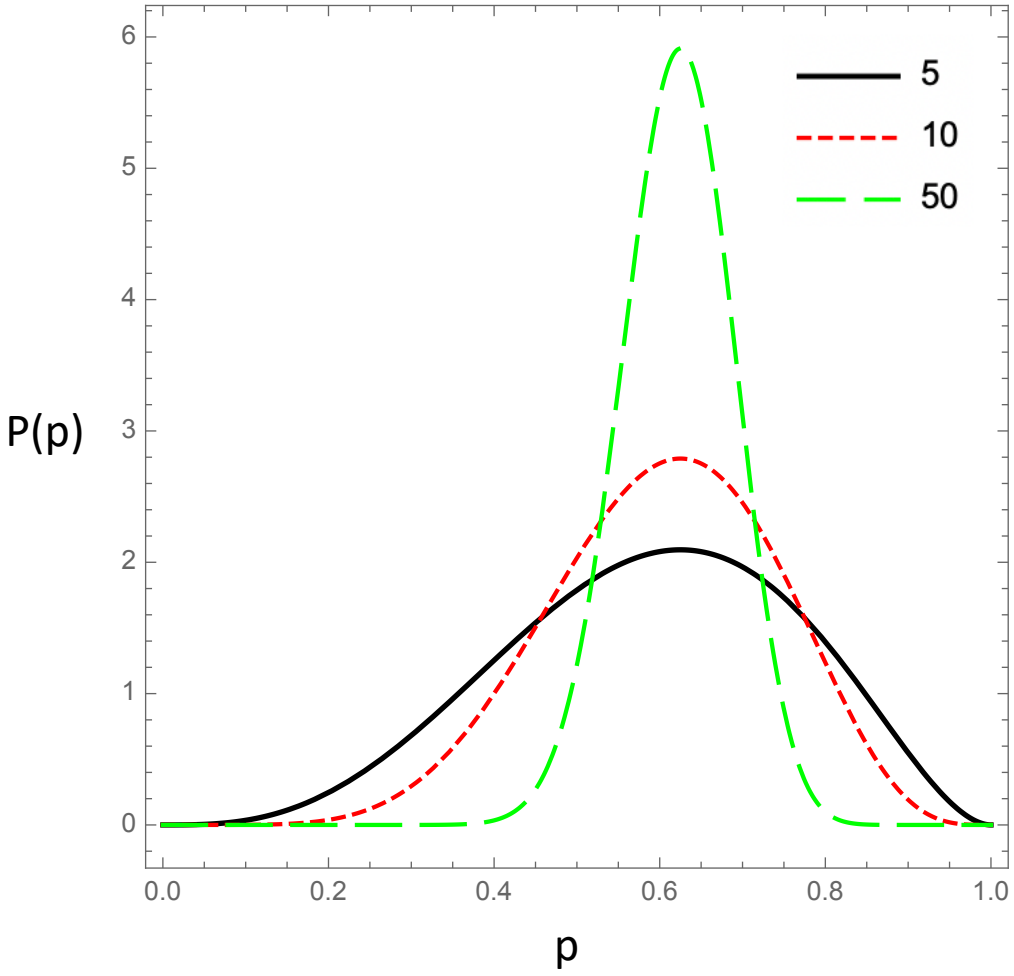


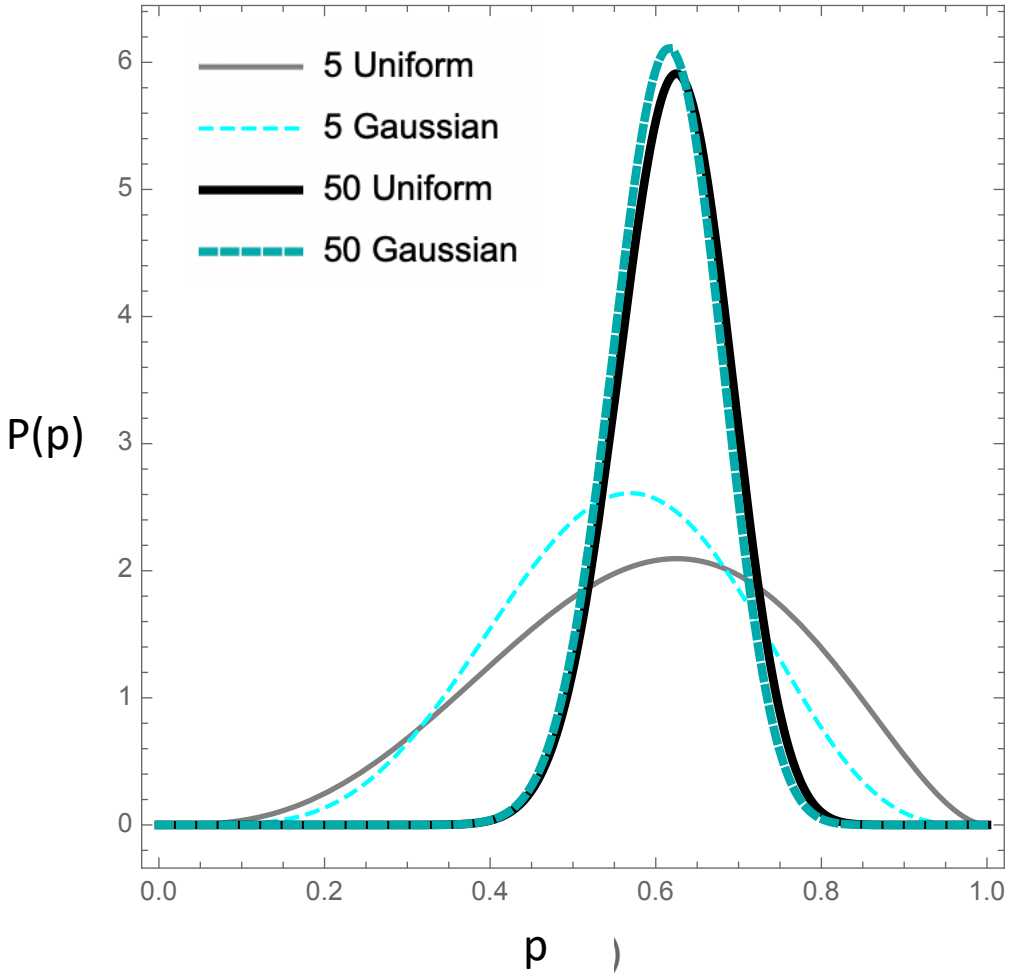
- Let “p” be the probability of heads
  - This is a continuous variable
  - We are not looking at Bayesian inference with classes just yet
- What could “p” be?
- Let x be the evidence (#H, #T)
- Bayes rule:
  - $Pr(p|x) = Pr(p)\ Pr(x|p)/Pr(x)$
- The possible values of p depends on
  - Prior: Pr(p) (overall likelihood next flip is heads)
  - Likelihood: Pr(#H,#T|p) (#H,#T if probability of heads is p)
  - Probability of evidence: Pr(#H,#T) (=1 in our case)

Bayes Rule

- How does Bayes' Rule come about?
- Conditional probability definition: $P(\theta|x) = P(\theta,x)/p(x)$
- Similarly, $P(x|\theta) = P(\theta,x)/p(\theta)$
- $P(\theta,x) = P(\theta)*P(x|\theta) = p(x)*P(\theta|x)$
- Therefore, divide by p(x), and $P(\theta|x) = P(\theta,x)/p(x)$
- H/T example:
  - $P(x=H|p) = p$; $P(x=T|p) = 1-p$
    - $P(X|p) = p^x(1-p)^{1-x}$
  - Likelihood: $P(\{x1,x2,x3,\ldots\}|p) = p^{\#H}\ (1-p)^{\#T}$
  - Prior, e.g., uniform = ½, or biased, e.g., $p^2\ (1-p)^2$
  - If uniform prior: $P(p|x) \sim p^{\#H}\ (1-p)^{\#T}$

Effect of N, Prior





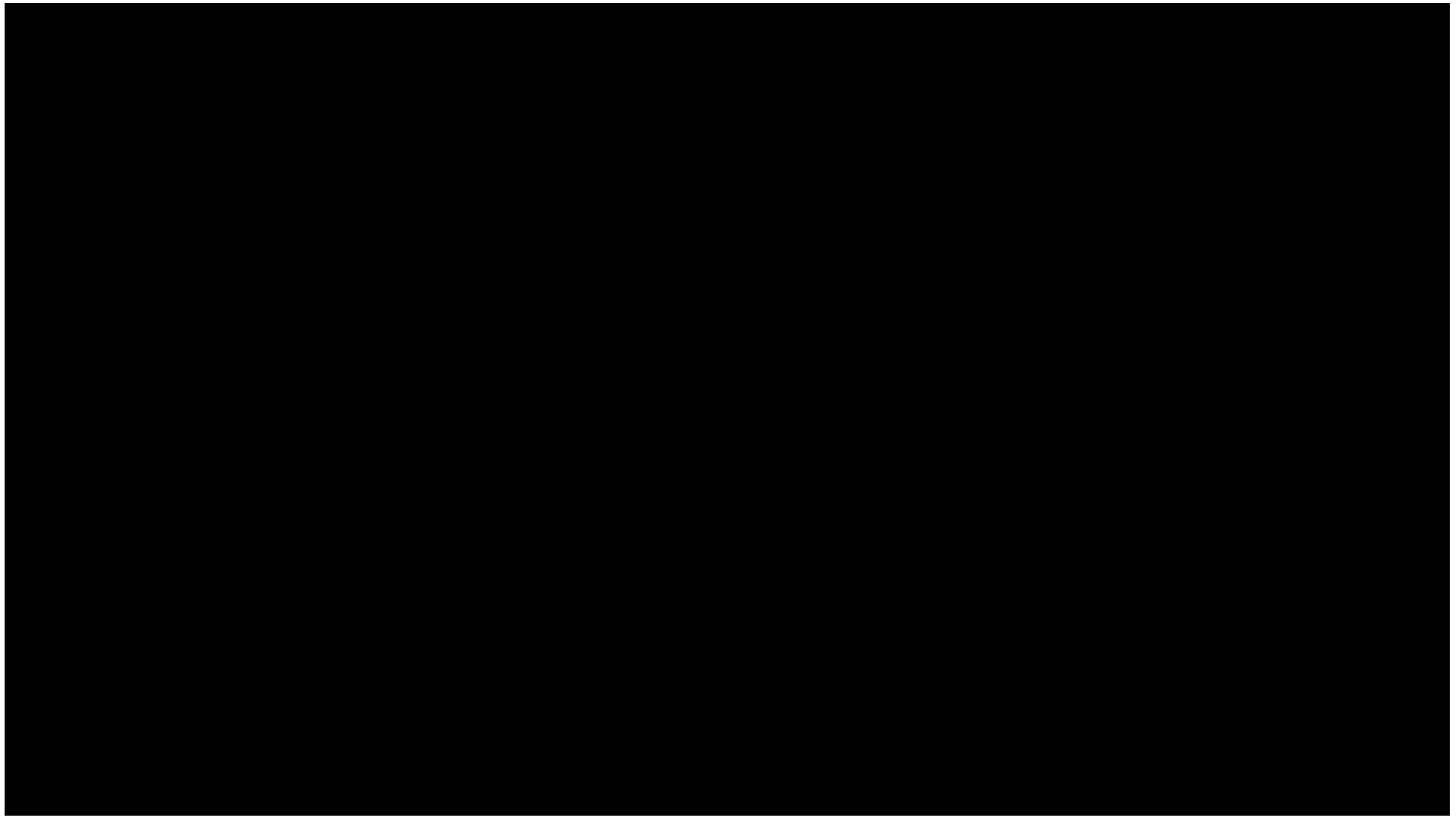
What we notice

- Maximum likelihood = most likely posterior if uniform prior
- Priors matter when data is small!
  - Posteriors show error reduces as more evidence is gathered
  - As more data is gathered, posteriors approximate true
- Useful in, e.g., neural networks to estimate classification error
- Real world example:
  - You or I could have really different views of the world
  - But if we agree on the facts, then as evidence is gathered we will approach the truth
  - E.g., Greek or Christian myths on how earth formed turned into our modern understanding of earth's formation as we gathered more evidence

“Call it.”

- What would you choose: heads or tails?
- We notice that evidence does not rule out a fair coin
- MLE: p = 5/7, so predict heads

Well done!



3:44

Bayes' Rule: *K* Classes

$$P(C_i \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid C_i)P(C_i)}{p(\mathbf{x})}$$

$$= \frac{p(\mathbf{x} \mid C_i)P(C_i)}{\sum_{k=1}^{K} p(\mathbf{x} \mid C_k)P(C_k)}$$

$$P(C_i) \geq 0 \text{ and } \sum_{i=1}^{K} P(C_i) = 1$$

$$\text{choose } C_i \text{ if } P(C_i \mid \mathbf{x}) = \max_k P(C_k \mid \mathbf{x})$$

Bayes' Estimator for Classes

- Until now, I showed coin tossing
  - This is an example of estimating a continuous variable, p
- We can also estimate the probability of discrete classes
- Same rules as before:
  - Treat ϑ as a random var with prior $p(\vartheta)$
  - Bayes' rule: $p(\vartheta|\mathcal{X}) = p(\mathcal{X}|\vartheta)\, p(\vartheta) / p(\mathcal{X})$
  - Full: $p(x|\mathcal{X}) = \int p(x|\vartheta)\, p(\vartheta|\mathcal{X})\, d\vartheta$
  - Maximum a Posteriori (MAP):

    $$\vartheta_{MAP} = \text{argmax}_{\vartheta}\, p(\vartheta|\mathcal{X})$$
  - Maximum Likelihood (ML): $\vartheta_{ML} = \text{argmax}_{\vartheta}\, p(\mathcal{X}|\vartheta)$
  - Bayes': $\vartheta_{Bayes'} = E[\vartheta|\mathcal{X}] = \int \vartheta\, p(\vartheta|\mathcal{X})\, d\vartheta$

Bayes' Estimator: Example

- $x^t \sim \mathcal{N}(\vartheta, \sigma_o^2)$ and $\vartheta \sim \mathcal{N}(\mu, \sigma^2)$
- $\vartheta_{ML} = m$
- $\vartheta_{MAP} = \vartheta_{Bayes'} =$

$$E[\theta \mid \mathcal{X}] = \frac{N/\sigma_0^2}{N/\sigma_0^2 + 1/\sigma^2} m + \frac{1/\sigma^2}{N/\sigma_0^2 + 1/\sigma^2} \mu$$

Note: this is O(1/N) different than MLE

Parametric Classification

$$g_i(x) = p(x \mid C_i) P(C_i)$$

or

$$g_i(x) = \log p(x \mid C_i) + \log P(C_i)$$

$$p(x \mid C_i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left[-\frac{(x-\mu_i)^2}{2\sigma_i^2} \right]$$

$$g_i(x) = -\frac{1}{2}\log 2\pi - \log \sigma_i - \frac{(x-\mu_i)^2}{2\sigma_i^2} + \log P(C_i)$$

- Given the sample $\mathcal{X} = \{x^t, r^t\}_{t=1}^N$

$$x \in \Re \qquad r_i^t = \begin{cases} 1 \text{ if } x^t \in C_i \\ 0 \text{ if } x^t \in C_j, j \neq i \end{cases}$$

- ML estimates are

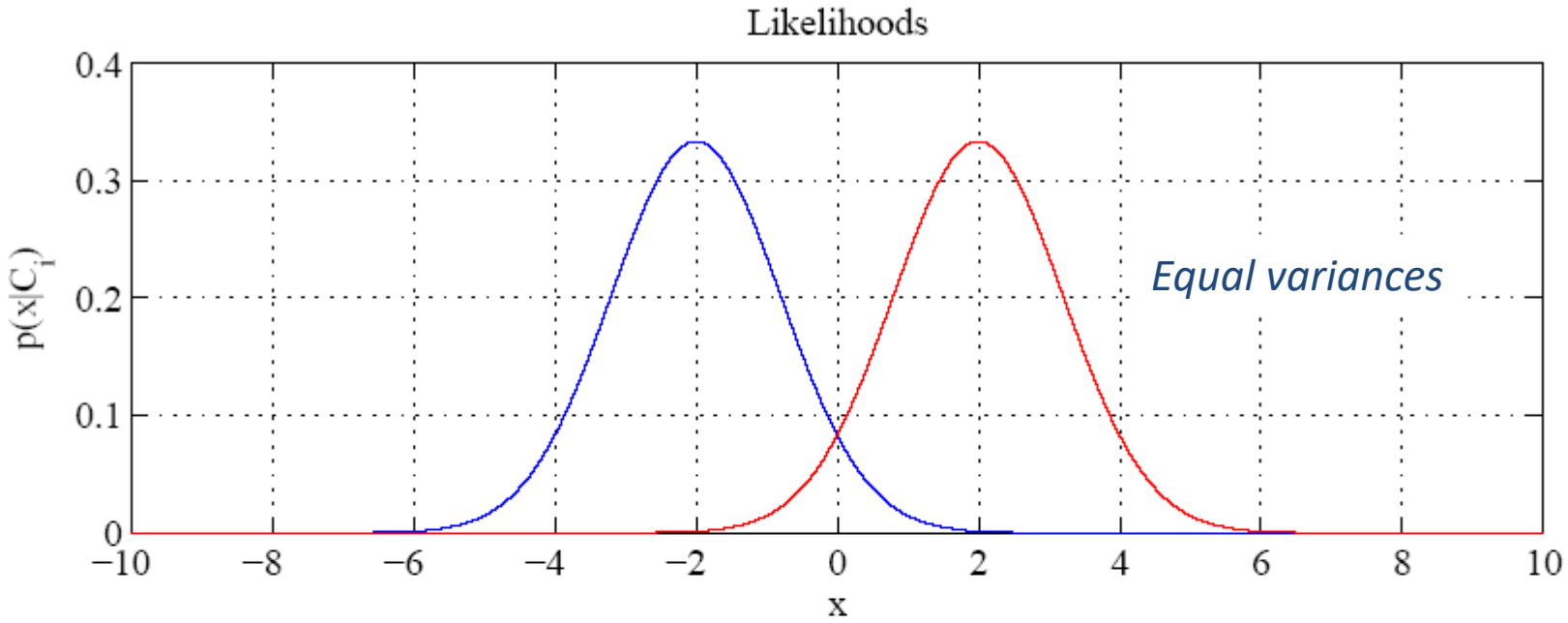
$$\hat{P}(C_i) = \frac{\sum_t r_i^t}{N} \quad m_i = \frac{\sum_t x^t r_i^t}{\sum_t r_i^t} \quad s_i^2 = \frac{\sum_t (x^t - m_i)^2 r_i^t}{\sum_t r_i^t}$$

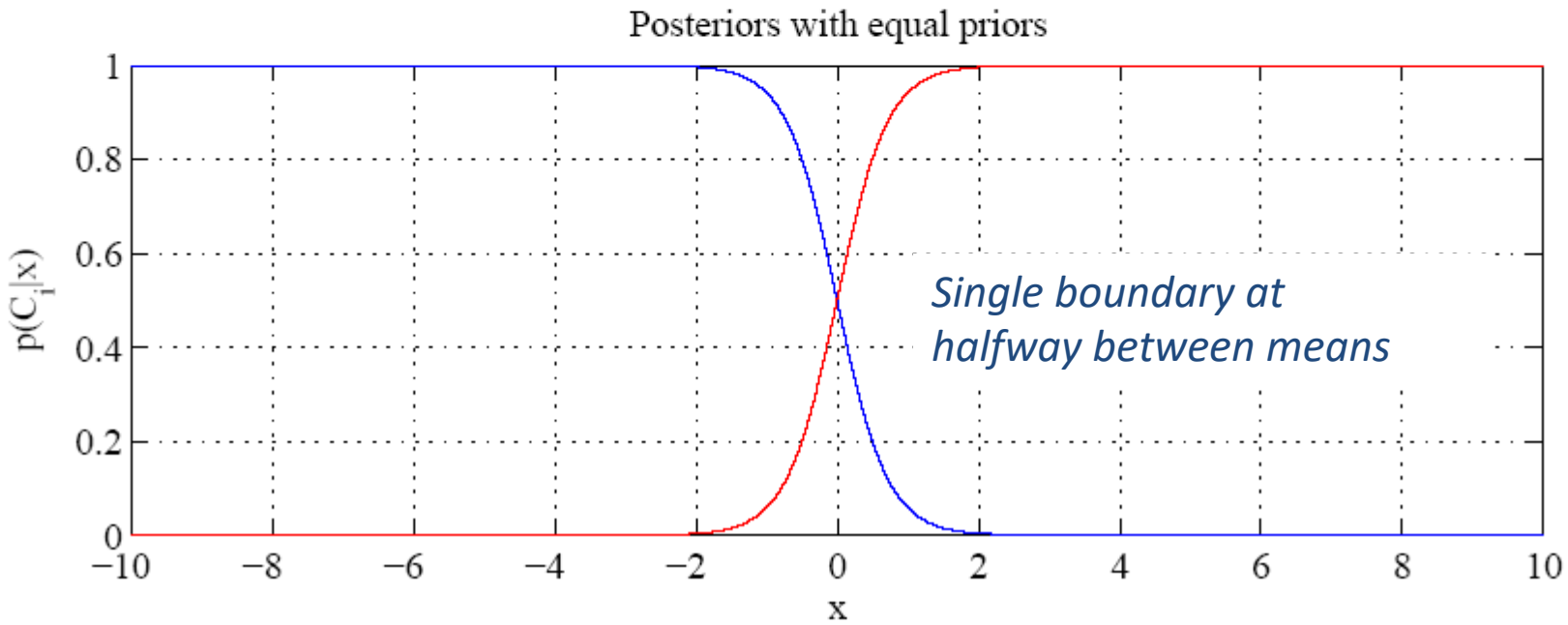
- Discriminant

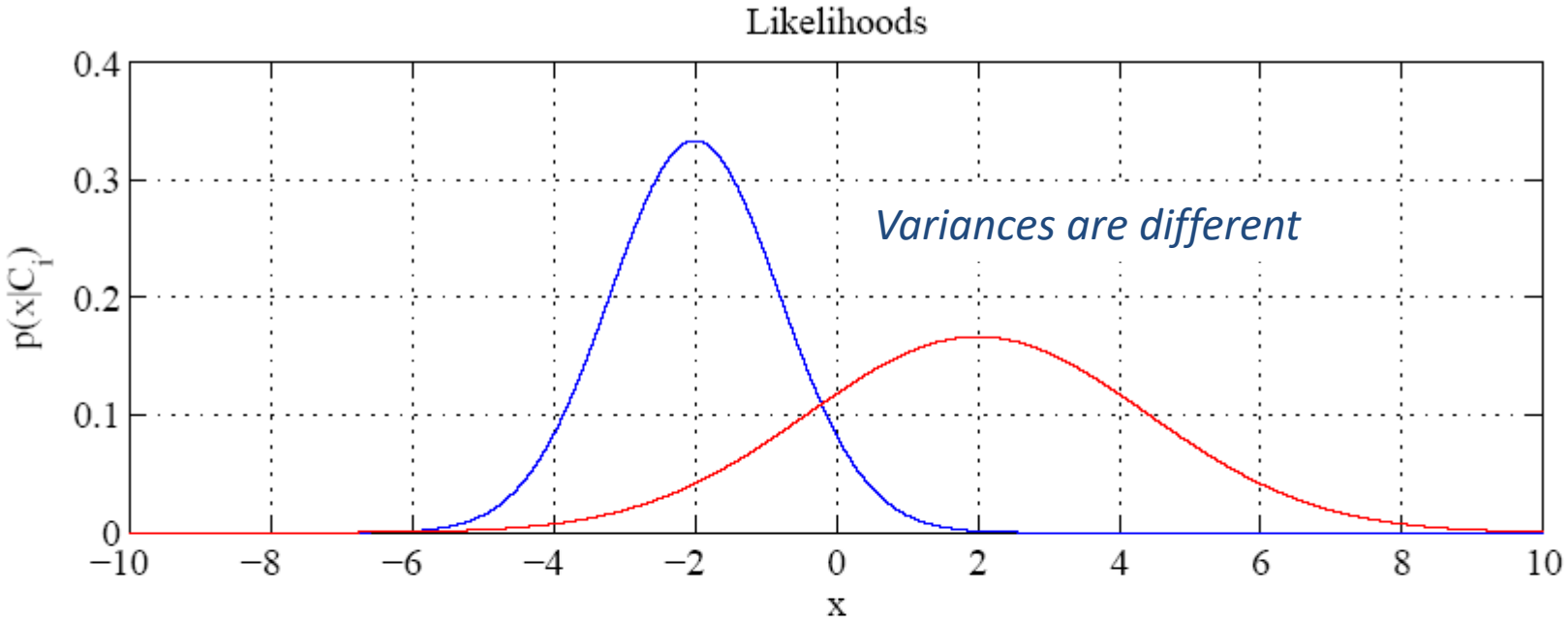
$$g_i(x) = -\frac{1}{2}\log 2\pi - \log s_i - \frac{(x - m_i)^2}{2s_i^2} + \log \hat{P}(C_i)$$

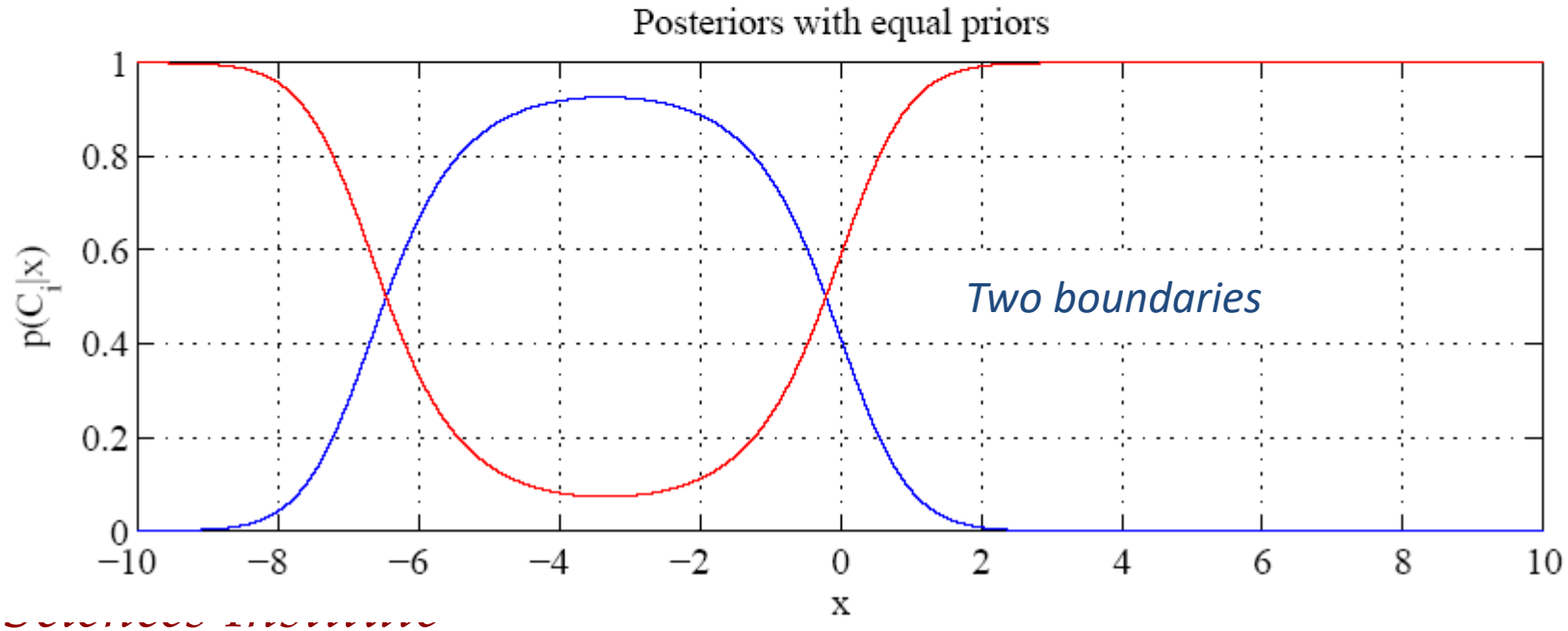
In English

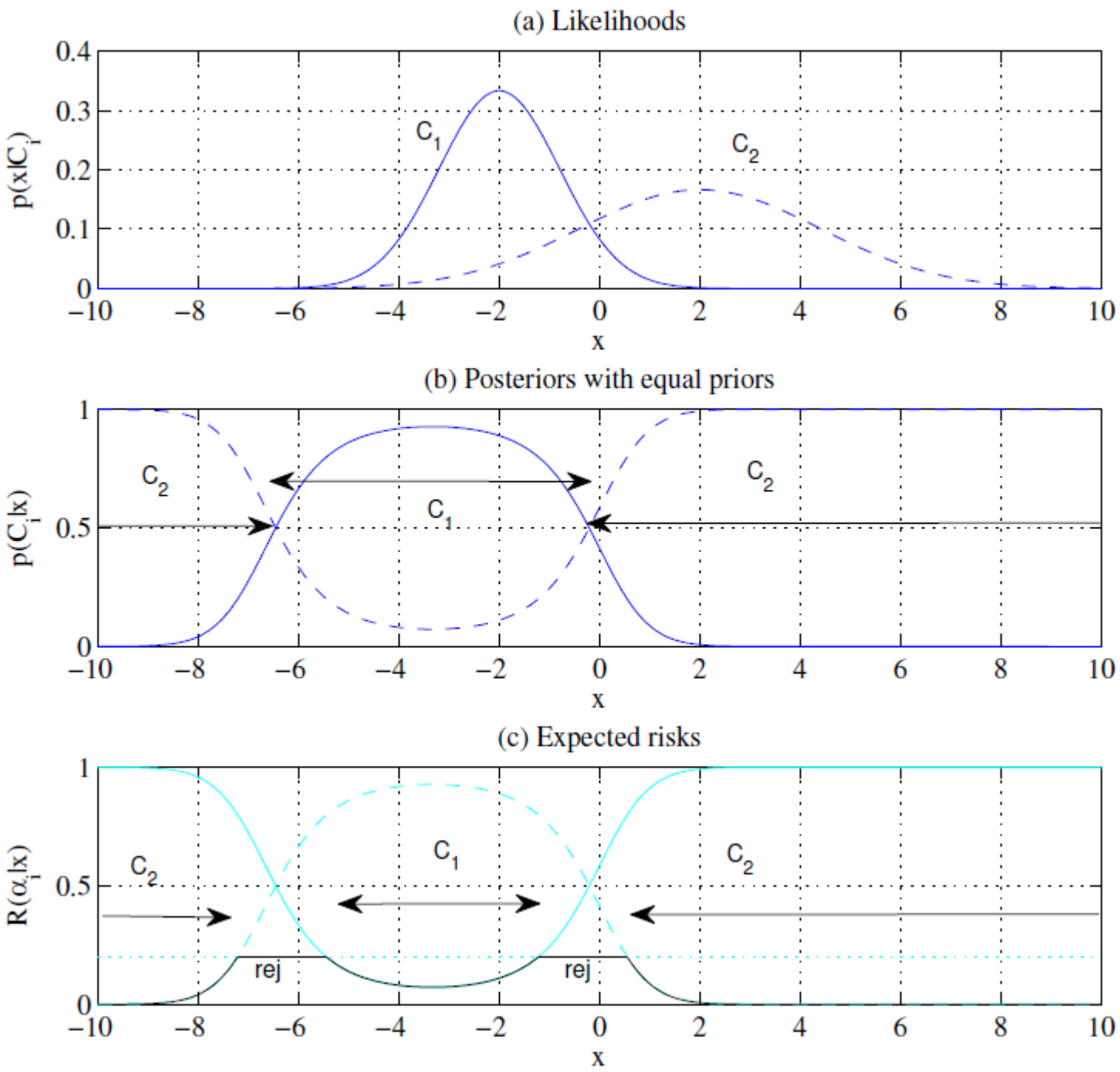
- Prior: how often each class appears
- Likelihood: how the data is distributed for each class
- Evidence: how often you see this set of feature values
- Posterior: Given you see a set of data, what is the likelihood of seeing class Ci given Ci's relative frequency (prior) and the frequency that the features would appear (likelihood)
- Discriminant is the unnormalized posterior
  - Whatever class has the largest discriminant has the largest posterior probability

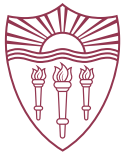

Likelihoods
0.4
0.3
0.2
0.1
0
p(x|C_i)
−10
−8
−6
−4
−2
0
2
4
6
8
10
x
Equal variances


Posteriors with equal priors
1
0.8
0.6
0.4
0.2
0
p(C_i|x)
−10
−8
−6
−4
−2
0
2
4
6
8
10
x
Single boundary at
halfway between means


Likelihoods
0.4
0.3
0.2
0.1
0
p(x|C_i)
−10
−8
−6
−4
−2
0
2
4
6
8
10
x
Variances are different


Posteriors with equal priors
1
0.8
0.6
0.4
0.2
0
p(C_i|x)
−10
−8
−6
−4
−2
0
2
4
6
8
10
x
Two boundaries


(a) Likelihoods
p(x|C_i)
C_1
C_2
x
(b) Posteriors with equal priors
p(C_i|x)
C_2
C_1
C_2
x
(c) Expected risks
R(α_i|x)
C_2
C_1
C_2
rej
rej
x



MULTIVARIATE DATA

Multivariate Data

- Multiple measurements (sensors)
- *d* inputs/features/attributes: *d*-variate
- *N* instances/observations/examples

$$\mathbf{X} = \begin{bmatrix} X_1^1 & X_2^1 & \cdots & X_d^1 \\ X_1^2 & X_2^2 & \cdots & X_d^2 \\ \vdots & & & \\ X_1^N & X_2^N & \cdots & X_d^N \end{bmatrix}$$

Multivariate Parameters

$$\text{Mean}: E[\mathbf{x}] = \boldsymbol{\mu} = [\mu_1, \ldots, \mu_d]^T$$

$$\text{Covariance}: \sigma_{ij} \equiv \text{Cov}(X_i, X_j)$$

$$\text{Correlation}: \text{Corr}(X_i, X_j) \equiv \rho_{ij} = \frac{\sigma_{ij}}{\sigma_i \sigma_j}$$

$$\Sigma \equiv \text{Cov}(\mathbf{X}) = E\left[(\mathbf{X} - \mu)(\mathbf{X} - \mu)^T\right] = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1d} \\ \sigma_{21} & \sigma_2^2 & \cdots & \sigma_{2d} \\ \vdots & & & \\ \sigma_{d1} & \sigma_{d2} & \cdots & \sigma_d^2 \end{bmatrix}$$

Parameter Estimation

$$\text{Sample mean } \mathbf{m}: m_i = \frac{\sum_{t=1}^{N} x_i^t}{N}, i = 1,\ldots,d$$

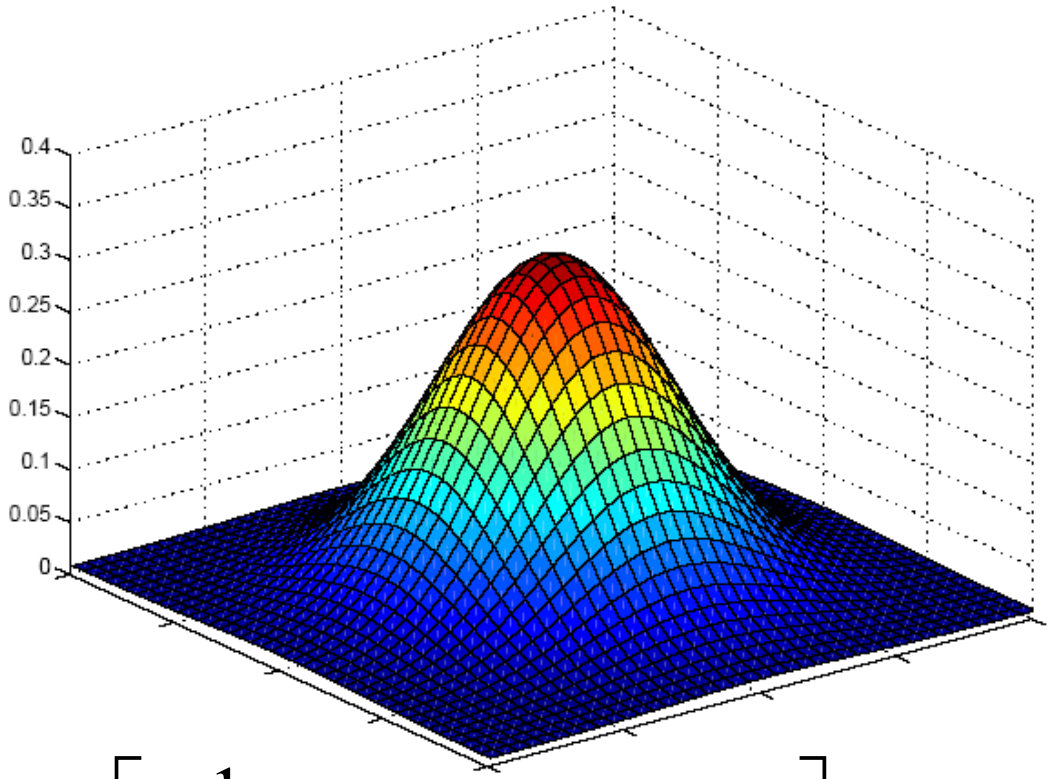
$$\text{Covariance matrix } \mathbf{S}: s_{ij} = \frac{\sum_{t=1}^{N} \left(x_i^t - m_i\right)\left(x_j^t - m_j\right)}{N}$$

$$\text{Correlation matrix } \mathbf{R}: r_{ij} = \frac{s_{ij}}{s_i s_j}$$

Estimation of Missing Values

- What to do if certain instances have missing attributes?
- Ignore those instances: not a good idea if the sample is small
- Use ‘missing’ as an attribute: may give information
- Imputation: Fill in the missing value
  - Mean imputation: Use the most likely value (e.g., mean)
  - Imputation by regression: Predict based on other attributes

Multivariate Normal Distribution



$$\mathbf{x} \sim \mathcal{N}_d(\boldsymbol{\mu}, \Sigma)$$

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right]$$

Multivariate Normal Distribution

- Mahalanobis distance: $(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu})$

  measures the distance from $\boldsymbol{x}$ to $\boldsymbol{\mu}$ in terms of $\boldsymbol{\Sigma}$
  (normalizes for difference in variances and correlations)

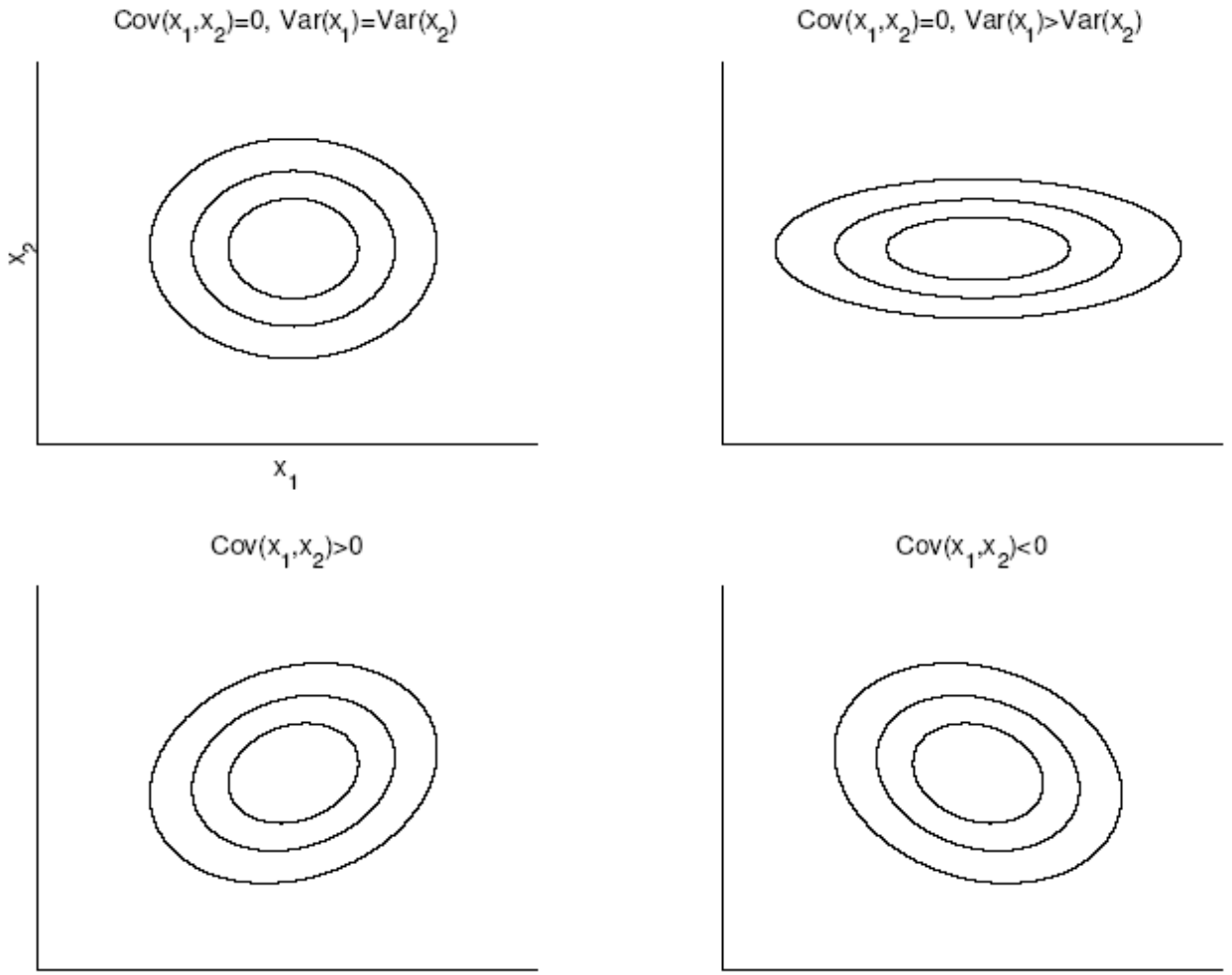
- Bivariate: $d = 2$

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

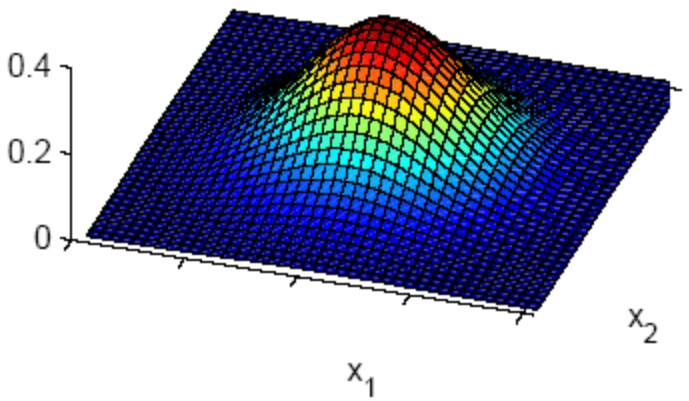
$$p(x_1, x_2) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \exp\left[-\frac{1}{2(1-\rho^2)}\left(z_1^2 - 2\rho z_1 z_2 + z_2^2\right)\right]$$

$$z_i = (x_i - \mu_i)/\sigma_i$$

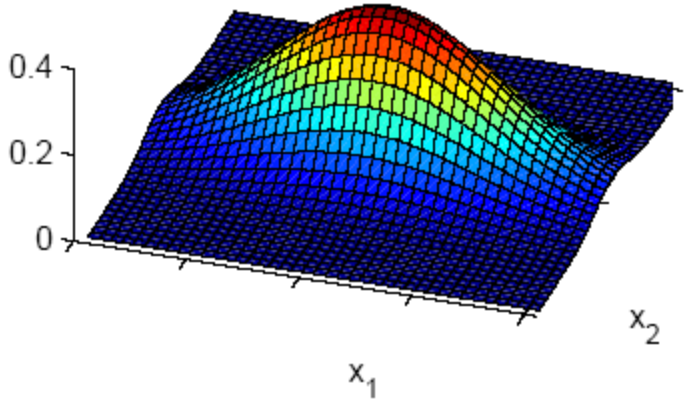
Bivariate Normal



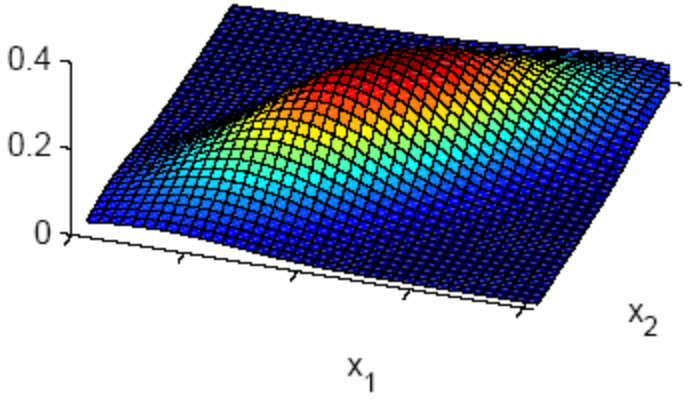
$\mathrm{Cov}(x_1,x_2)=0$, $\mathrm{Var}(x_1)=\mathrm{Var}(x_2)$



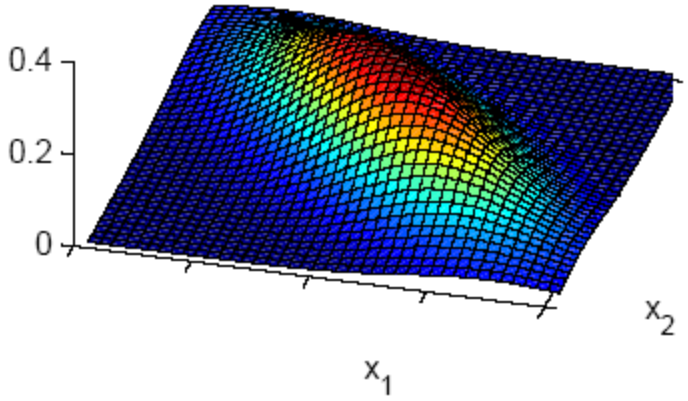
$\mathrm{Cov}(x_1,x_2)=0$, $\mathrm{Var}(x_1)>\mathrm{Var}(x_2)$



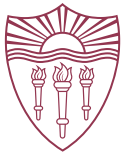
$\mathrm{Cov}(x_1,x_2)>0$



$\mathrm{Cov}(x_1,x_2)<0$



Independent Inputs: Naive Bayes



- If x_i are independent, offdiagonals of ∑ are 0, Mahalanobis distance reduces to weighted (by $1/\sigma_i$) Euclidean distance:

$$p(\mathbf{x}) = \prod_{i=1}^{d} p_i(x_i) = \frac{1}{(2\pi)^{d/2} \prod_{i=1}^{d} \sigma_i} \exp\left[-\frac{1}{2} \sum_{i=1}^{d} \left(\frac{x_i - \mu_i}{\sigma_i} \right)^2 \right]$$

- If variances are also equal, reduces to Euclidean distance

Multivariate Parametric Classification

- If $p(\boldsymbol{x} \mid C_i) \sim N(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$

$$p(\mathbf{x} \mid C_i) = \frac{1}{(2\pi)^{d/2} |\Sigma_i|^{1/2}} \exp\left[-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_i)^T \Sigma_i^{-1} (\mathbf{x} - \boldsymbol{\mu}_i) \right]$$

- Discriminant functions

$$g_i(\mathbf{x}) = \log p(\mathbf{x} \mid C_i) + \log P(C_i)$$

$$= -\frac{d}{2} \log 2\pi - \frac{1}{2} \log |\Sigma_i| - \frac{1}{2} (\mathbf{x} - \mu_i)^T \Sigma_i^{-1} (\mathbf{x} - \mu_i) + \log P(C_i)$$

Estimation of Parameters

$$\hat{P}(C_i) = \frac{\sum_t r_i^t}{N}$$

$$\mathbf{m}_i = \frac{\sum_t r_i^t \mathbf{x}^t}{\sum_t r_i^t}$$

$$\mathbf{S}_i = \frac{\sum_t r_i^t (\mathbf{x}^t - \mathbf{m}_i)(\mathbf{x}^t - \mathbf{m}_i)^T}{\sum_t r_i^t}$$

$$g_i(\mathbf{x}) = -\frac{1}{2}\log|\mathbf{S}_i| - \frac{1}{2}(\mathbf{x} - \mathbf{m}_i)^T \mathbf{S}_i^{-1}(\mathbf{x} - \mathbf{m}_i) + \log \hat{P}(C_i)$$

Different $\mathbf{S}_i$

- Quadratic discriminant

$$g_i(\mathbf{x}) = -\frac{1}{2}\log|\mathbf{S}_i| - \frac{1}{2}\left(\mathbf{x}^T\mathbf{S}_i^{-1}\mathbf{x} - 2\mathbf{x}^T\mathbf{S}_i^{-1}\mathbf{m}_i + \mathbf{m}_i^T\mathbf{S}_i^{-1}\mathbf{m}_i\right) + \log\hat{P}(C_i)$$

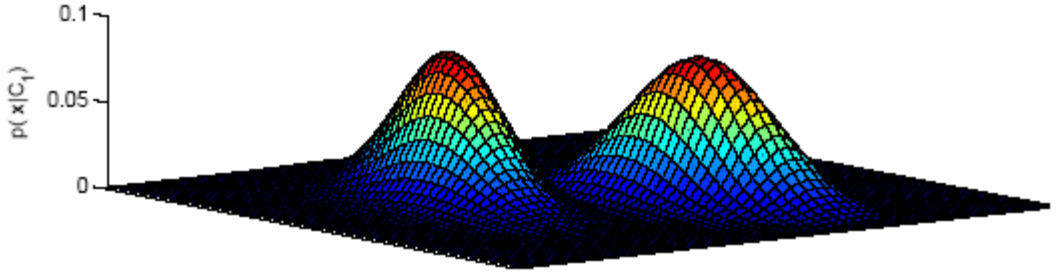
$$= \mathbf{x}^T\mathbf{W}_i\mathbf{x} + \mathbf{w}_i^T\mathbf{x} + w_{i0}$$

where

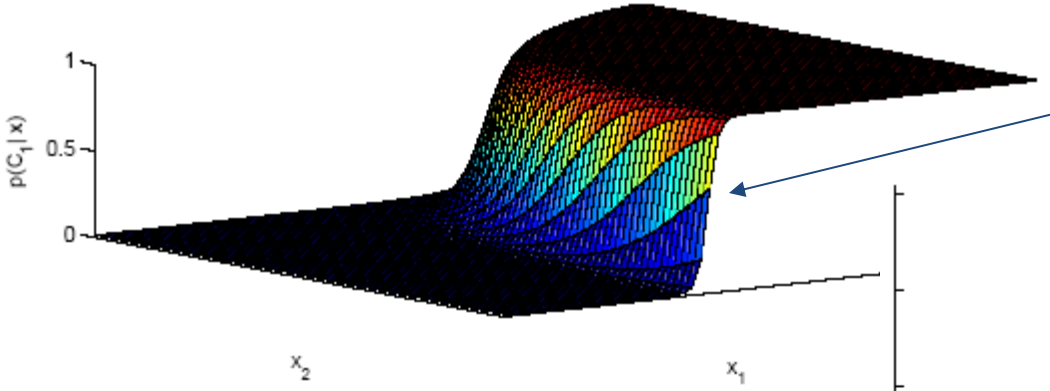
$$\mathbf{W}_i = -\frac{1}{2}\mathbf{S}_i^{-1}$$

$$\mathbf{w}_i = \mathbf{S}_i^{-1}\mathbf{m}_i$$

$$w_{i0} = -\frac{1}{2}\mathbf{m}_i^T\mathbf{S}_i^{-1}\mathbf{m}_i - \frac{1}{2}\log|\mathbf{S}_i| + \log\hat{P}(C_i)$$

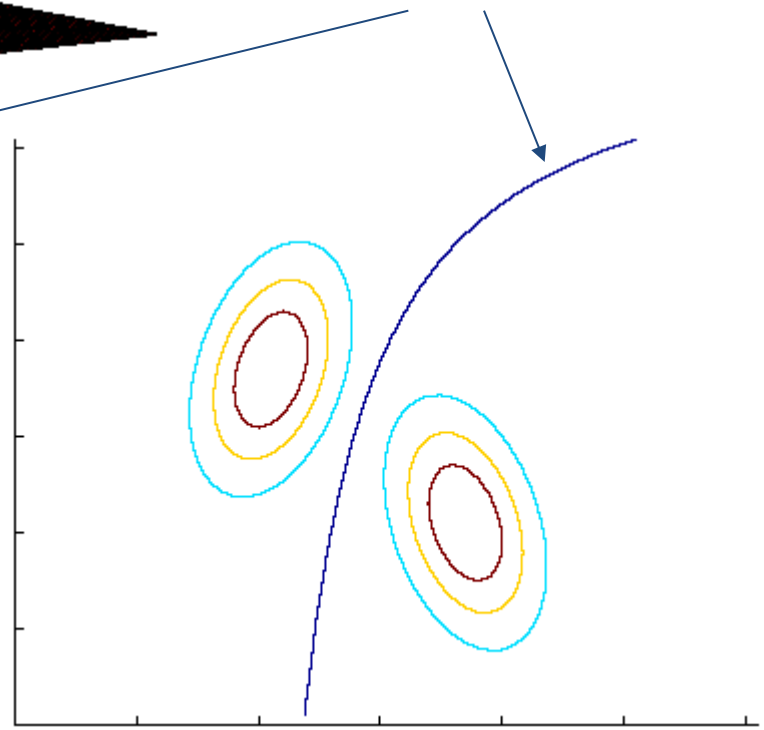


likelihoods



posterior for C_1

discriminant:
$P(C_1 | \boldsymbol{x}) = 0.5$



Common Covariance Matrix **S**

- Shared common sample covariance **S**

$$\mathbf{S} = \sum_i \hat{P}(C_i)\mathbf{S}_i$$

- Discriminant reduces to

$$g_i(\boldsymbol{x}) = -\frac{1}{2}(\boldsymbol{x} - \boldsymbol{m}_i)^T \mathbf{S}^{-1}(\boldsymbol{x} - \boldsymbol{m}_i) + \log \hat{P}(C_i)$$

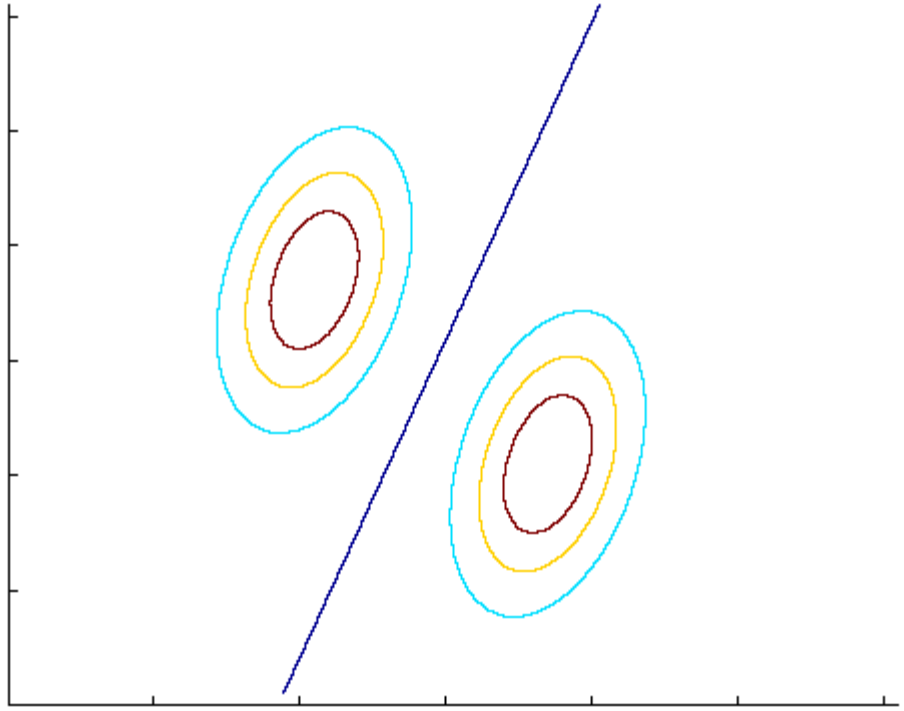
which is a linear discriminant

$$g_i(\mathbf{x}) = \mathbf{w}_i^T \mathbf{x} + w_{i0}$$

where

$$\mathbf{w}_i = \mathbf{S}^{-1}\mathbf{m}_i \quad w_{i0} = -\frac{1}{2}\mathbf{m}_i^T \mathbf{S}^{-1}\mathbf{m}_i + \log \hat{P}(C_i)$$

Common Covariance Matrix **S**



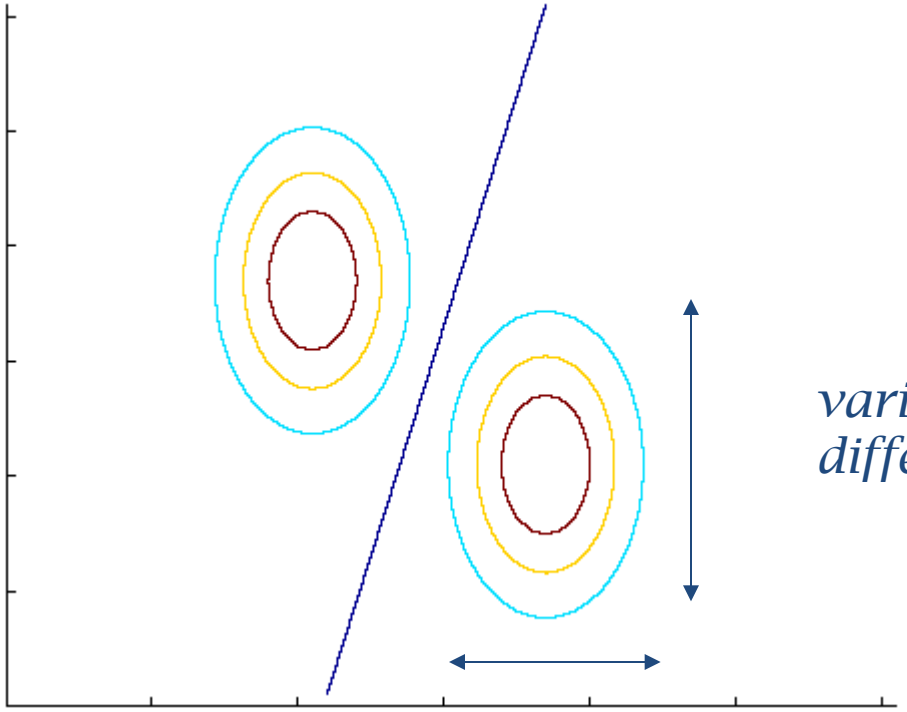
Diagonal **S**

- When x_j $j = 1,..d$, are independent, $\sum$ is diagonal
  $p(\boldsymbol{x}|C_i) = \prod_j p(x_j|C_i)$ (Naive Bayes' assumption)

$$g_i(\mathbf{x}) = -\frac{1}{2}\sum_{j=1}^{d}\left(\frac{x_j^t - m_{ij}}{s_j}\right)^2 + \log \hat{P}(C_i)$$

Classify based on weighted Euclidean distance (in s_j units) to the nearest mean

Diagonal **S**



variances may be different

Diagonal **S**, equal variances

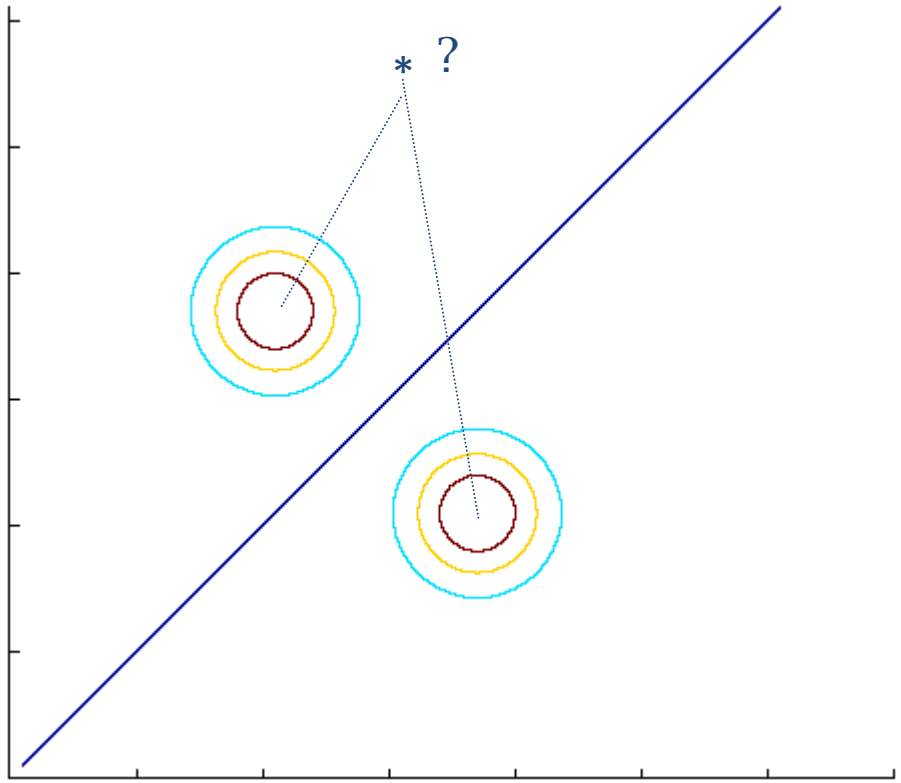
- Nearest mean classifier: Classify based on Euclidean distance to the nearest mean

$$g_i(\mathbf{x}) = -\frac{\|\mathbf{x}-\mathbf{m}_i\|^2}{2s^2} + \log \hat{P}(C_i)$$

$$= -\frac{1}{2s^2}\sum_{j=1}^{d}\left(x_j^t - m_{ij}\right)^2 + \log \hat{P}(C_i)$$

- Each mean can be considered a prototype or template and this is template matching

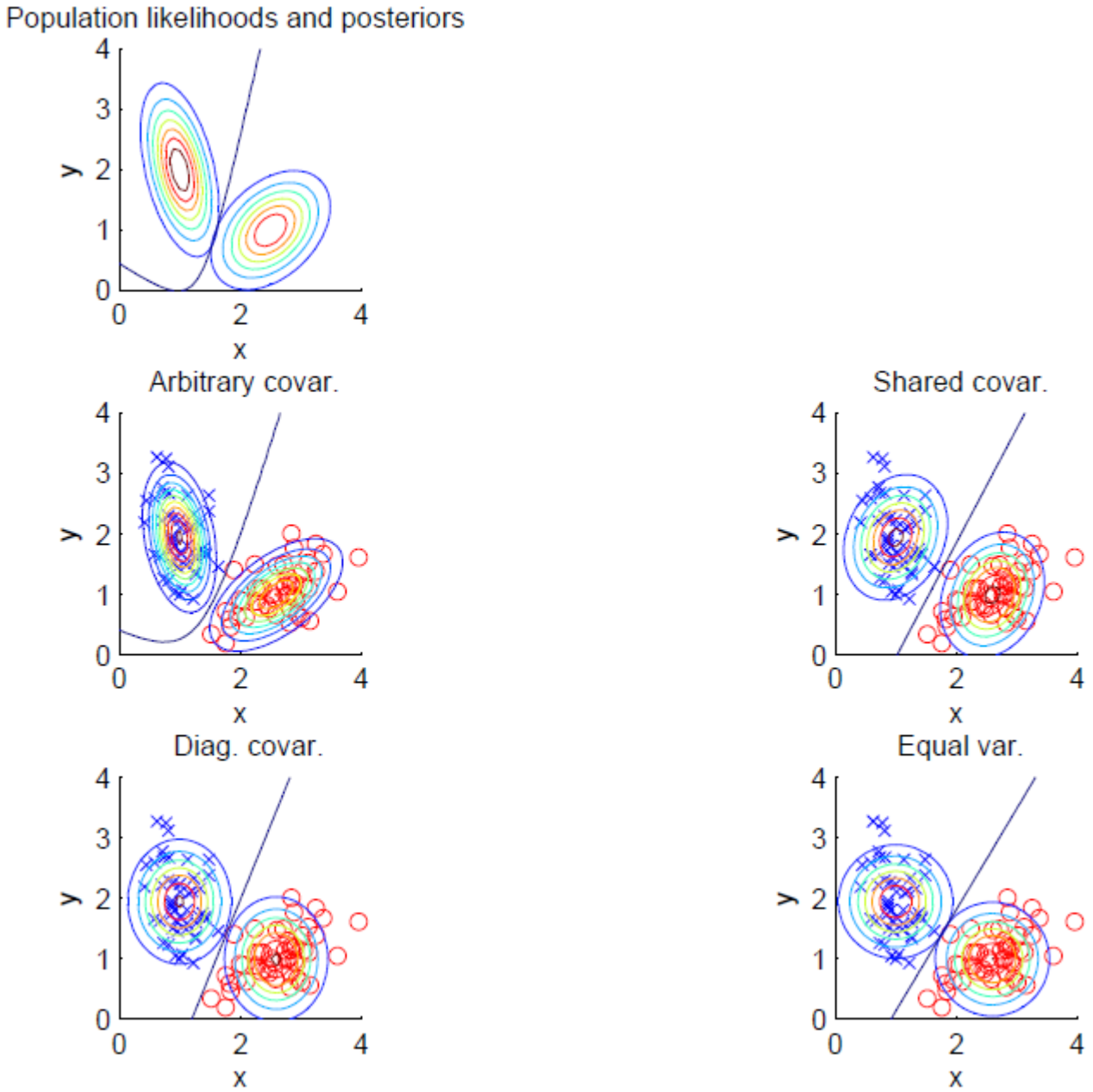
Diagonal **S**, equal variances



Model Selection

Assumption	*Covariance matrix*	*No of parameters*
Shared, Hyperspheric	$\mathbf{S}_i=\mathbf{S}=s^2\mathbf{I}$	1
Shared, Axis-aligned	$\mathbf{S}_i=\mathbf{S}$, with $s_{ij}=0$	d
Shared, Hyperellipsoidal	$\mathbf{S}_i=\mathbf{S}$	$d(d+1)/2$
Different, Hyperellipsoidal	$\mathbf{S}_i$	$K\,d(d+1)/2$

- As we increase complexity (less restricted **S**), bias decreases and variance increases
- Assume simple models (allow some bias) to control variance (regularization)


Population likelihoods and posteriors
Arbitrary covar.
Shared covar.
Diag. covar.
Equal var.
x
y
0
1
2
3
4

Discrete Features

- Binary features: $p_{ij} \equiv p\left(x_j = 1 \mid C_i\right)$

  if x_j are independent (Naive Bayes')

$$p\left(x \mid C_i\right) = \prod_{j=1}^{d} p_{ij}^{x_j} \left(1 - p_{ij}\right)^{\left(1 - x_j\right)}$$

  the discriminant is linear

$$g_i\left(\mathbf{x}\right) = \log p\left(\mathbf{x} \mid C_i\right) + \log P\left(C_i\right)$$
$$= \sum_j \left[x_j \log p_{ij} + \left(1 - x_j\right) \log \left(1 - p_{ij}\right)\right] + \log P\left(C_i\right)$$

  Estimated parameters $\hat{p}_{ij} = \frac{\sum_t x_j^t r_i^t}{\sum_t r_i^t}$

Discrete Features

- Multinomial (1-of-n_j) features: $x_j \hat{I} \{v_1, v_2, \ldots, v_{n_j}\}$

$$p_{ijk} \equiv p(z_{jk}=1 \mid C_i) = p(x_j = v_k \mid C_i)$$

if x_j are independent

$$p(\mathbf{x} \mid C_i) = \prod_{j=1}^{d} \prod_{k=1}^{n_j} p_{ijk}^{z_{jk}}$$

$$g_i(\mathbf{x}) = \sum_j \sum_k z_{jk} \log p_{ijk} + \log P(C_i)$$

$$\hat{p}_{ijk} = \frac{\sum_t z_{jk}^t r_i^t}{\sum_t r_i^t}$$

PARAMETRIC CLASSIFIERS

Likelihood- vs. Discriminant-based Classification

- Likelihood-based: Assume a model for $p(\boldsymbol{x}|C_i)$, use Bayes' rule to calculate $P(C_i|\boldsymbol{x})$

  $$g_i(\boldsymbol{x}) = \log P(C_i|\boldsymbol{x})p(\boldsymbol{x})$$

- Discriminant-based: Assume a model for $g_i(\boldsymbol{x}|\Phi_i)$; no density estimation
- Estimating the boundaries is enough; no need to accurately estimate the densities inside the boundaries
- Again, this is because it is an unnormalized version of the posterior. If the goal is simply to predict the most likely class, the probabilities are not as high a concern
- BUT: posterior will give you confidence in each class, which is needed in many situations

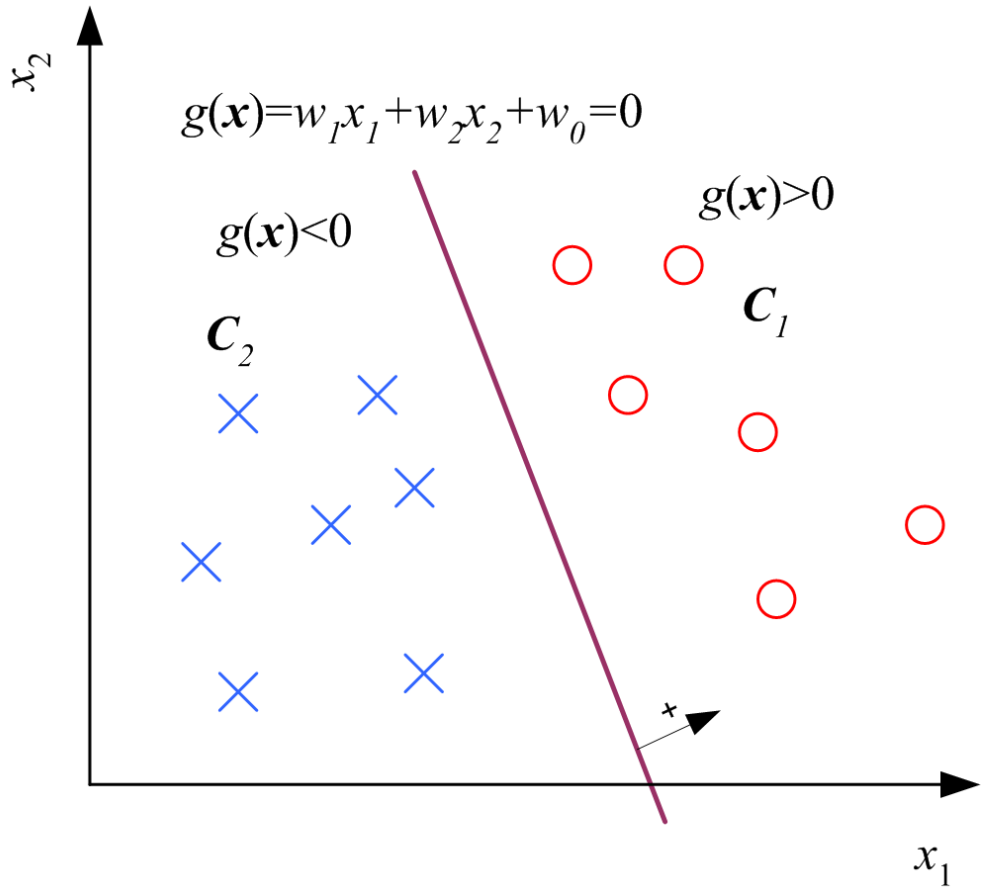
Linear Discriminant

- Linear discriminant:

$$g_i(\mathbf{x} \mid \mathbf{w}_i, w_{i0}) = \mathbf{w}_i^T \mathbf{x} + w_{i0} = \sum_{j=1}^{d} w_{ij} x_j + w_{i0}$$

- Advantages:
  - Simple: O(d) space/computation
  - Knowledge extraction: Weighted sum of attributes; positive/negative weights, magnitudes (credit scoring)
  - Optimal when $p(\boldsymbol{x} \mid C_i)$ are Gaussian with shared cov matrix; useful when classes are (almost) linearly separable

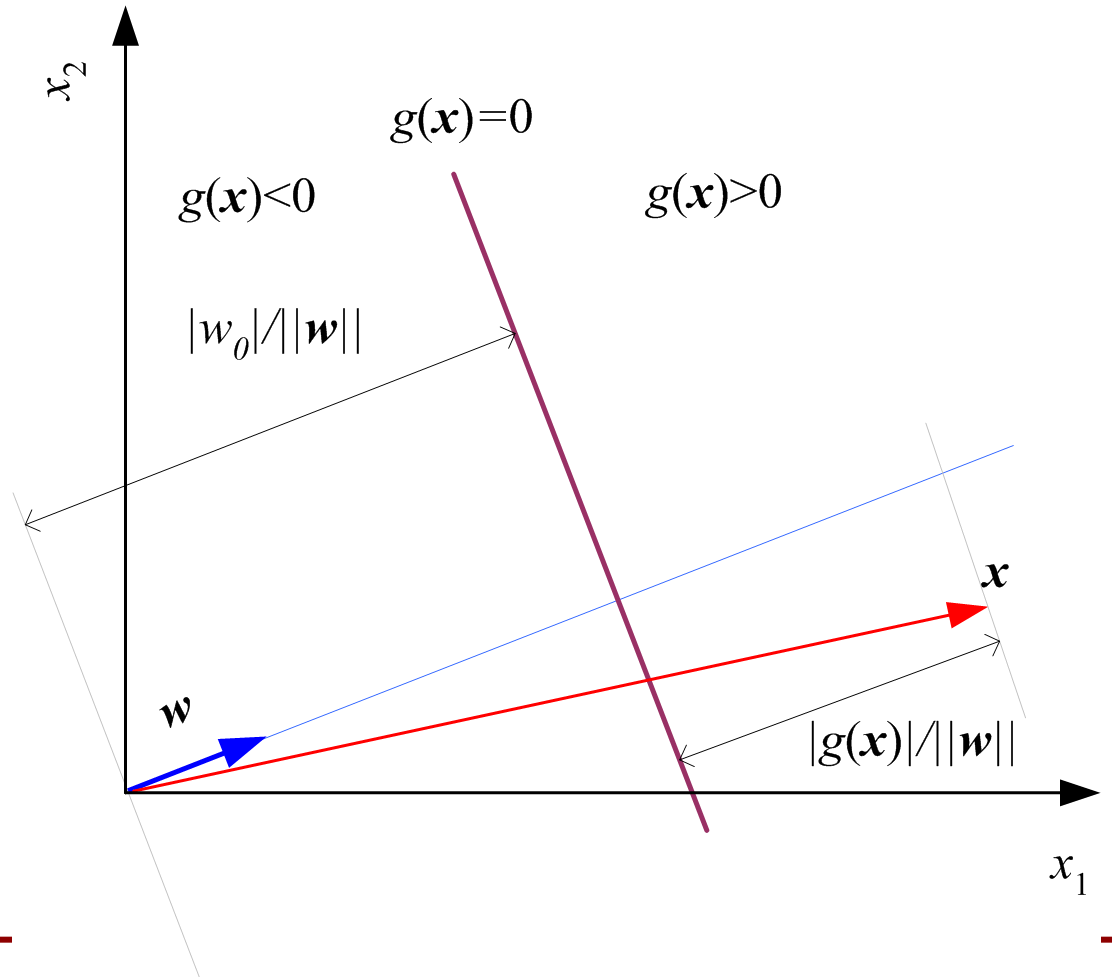
Two Classes



$$
\begin{aligned}
g(\mathbf{x}) &= g_1(\mathbf{x}) - g_2(\mathbf{x}) \\
&= \left(\mathbf{w}_1^T\mathbf{x} + w_{10}\right) - \left(\mathbf{w}_2^T\mathbf{x} + w_{20}\right) \\
&= \left(\mathbf{w}_1 - \mathbf{w}_2\right)^T\mathbf{x} + \left(w_{10} - w_{20}\right) \\
&= \mathbf{w}^T\mathbf{x} + w_0
\end{aligned}
$$

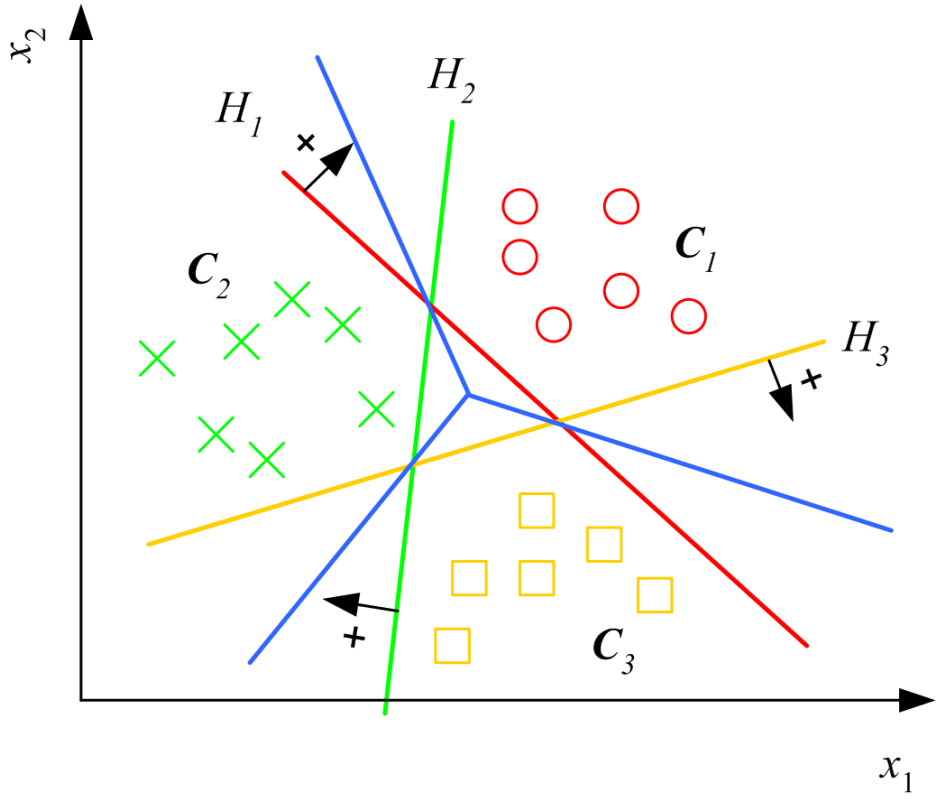
$$
\text{choose} \begin{cases} C_1 & \text{if } g(\mathbf{x}) > 0 \\ C_2 & \text{otherwise} \end{cases}
$$

Geometry


x_2
$g(\boldsymbol{x})=0$
$g(\boldsymbol{x})<0$
$g(\boldsymbol{x})>0$
$|w_0|/||\boldsymbol{w}||$
$\boldsymbol{x}$
$\boldsymbol{w}$
$|g(\boldsymbol{x})|/||\boldsymbol{w}||$
x_1

Multiple Classes

$$g_i(\mathbf{x} \mid \mathbf{w}_i, w_{i0}) = \mathbf{w}_i^T \mathbf{x} + w_{i0}$$

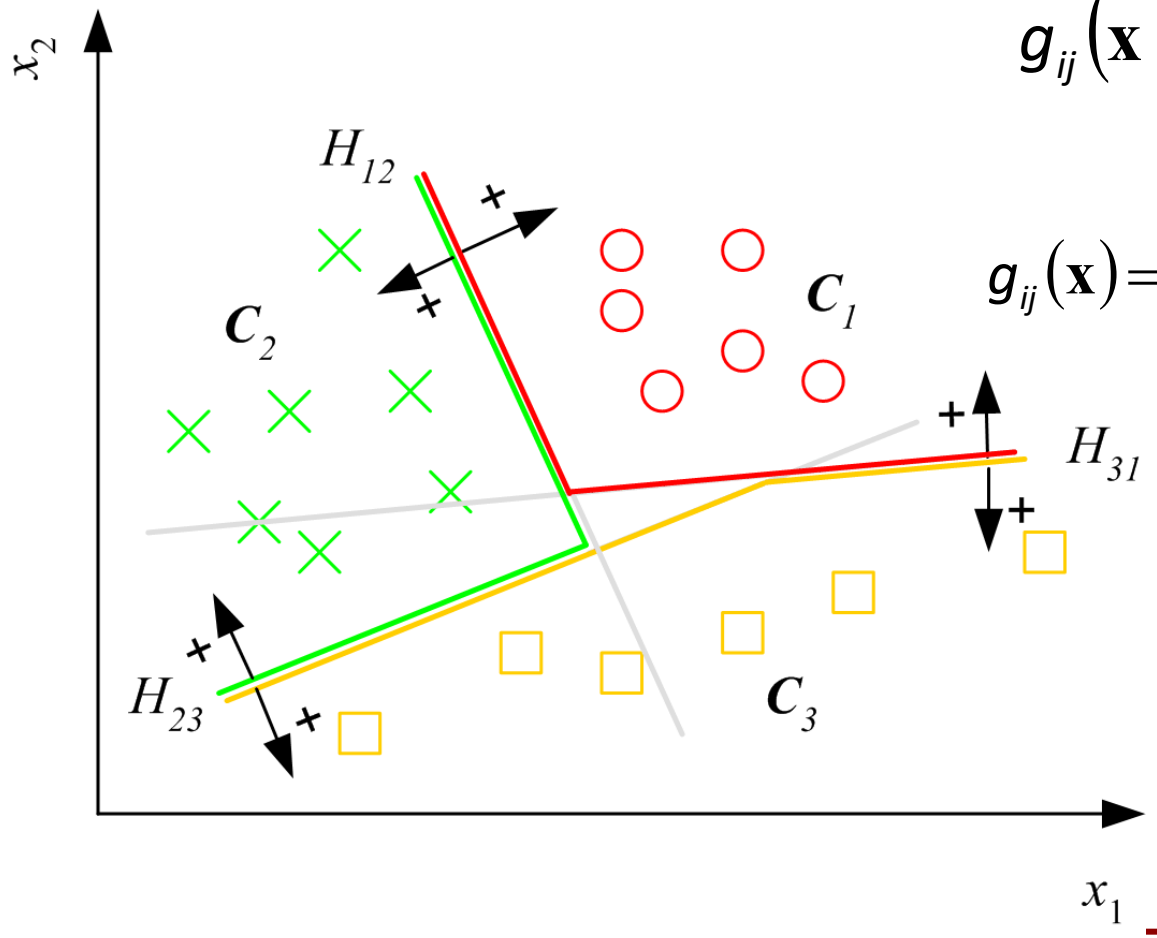


Choose C_i if

$$g_i(\mathbf{x}) = \max_{j=1}^{K} g_j(\mathbf{x})$$

Classes are linearly separable

Pairwise Separation



$$g_{ij}\left(\mathbf{x} \mid \mathbf{w}_{ij}, w_{ij0}\right) = \mathbf{w}_{ij}^T \mathbf{x} + w_{ij0}$$

$$g_{ij}(\mathbf{x}) = \begin{cases} > 0 & \text{if } \mathbf{x} \in C_i \\ \leq 0 & \text{if } \mathbf{x} \in C_j \\ \text{don't care} & \text{otherwise} \end{cases}$$

choose C_i if

$$\forall j \neq i, g_{ij}(\mathbf{x}) > 0$$

From Discriminants to Posteriors

When $p(\mathbf{x} \mid C_i) \sim N(\boldsymbol{\mu}_i, \boldsymbol{\Sigma})$

$$g_i(\mathbf{x} \mid \mathbf{w}_i, w_{i0}) = \mathbf{w}_i^T \mathbf{x} + w_{i0}$$

$$\mathbf{w}_i = \Sigma^{-1}\mu_i \quad w_{i0} = -\frac{1}{2}\mu_i^T \Sigma^{-1}\mu_i + \log P(C_i)$$

$$y \equiv P(C_1 \mid \mathbf{x}) \text{ and } P(C_2 \mid \mathbf{x}) = 1 - y$$

$$\text{choose } C_1 \text{ if } \begin{cases} y > 0.5 \\ y/(1-y) > 1 \\ \log[y/(1-y)] > 0 \end{cases} \text{ and } C_2 \text{ otherwise}$$

$$\text{logit}(P(C_1 \mid \mathbf{x})) = \log \frac{P(C_1 \mid \mathbf{x})}{1 - P(C_1 \mid \mathbf{x})} = \log \frac{P(C_1 \mid \mathbf{x})}{P(C_2 \mid \mathbf{x})}$$

$$= \log \frac{p(\mathbf{x} \mid C_1)}{p(\mathbf{x} \mid C_2)} + \log \frac{P(C_1)}{P(C_2)}$$

$$= \log \frac{(2\pi)^{-d/2} |\Sigma|^{-1/2} \exp\left[-(1/2)(\mathbf{x} - \mu_1)^T \Sigma^{-1} (\mathbf{x} - \mu_1)\right]}{(2\pi)^{-d/2} |\Sigma|^{-1/2} \exp\left[-(1/2)(\mathbf{x} - \mu_2)^T \Sigma^{-1} (\mathbf{x} - \mu_2)\right]} + \log \frac{P(C_1)}{P(C_2)}$$

$$= \mathbf{w}^T \mathbf{x} + w_0$$

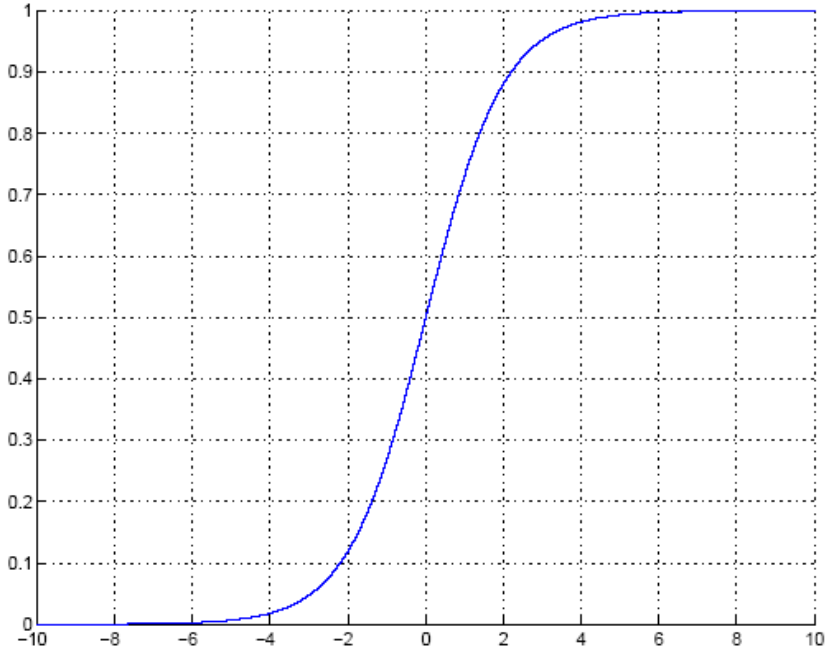
where $\mathbf{w} = \Sigma^{-1}(\mu_1 - \mu_2)$ $w_0 = -\frac{1}{2}(\mu_1 + \mu_2)^T \Sigma^{-1}(\mu_1 - \mu_2)$

The inverse of logit

$$\log \frac{P(C_1 \mid \mathbf{x})}{1 - P(C_1 \mid \mathbf{x})} = \mathbf{w}^T \mathbf{x} + w_0$$

$$P(C_1 \mid \mathbf{x}) = \text{sigmoid}(\mathbf{w}^T \mathbf{x} + w_0) = \frac{1}{1 + \exp\left[-(\mathbf{w}^T \mathbf{x} + w_0)\right]}$$

Sigmoid (Logistic) Function



Calculate $g(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0$ and choose C_1 if $g(\mathbf{x}) > 0$, or

Calculate $y = \text{sigmoid}(\mathbf{w}^T\mathbf{x} + w_0)$ and choose C_1 if $y > 0.5$

Gradient-Descent

- We have not yet discussed how to estimate the parameters…
- E($\boldsymbol{w}$|X) is error with parameters $\boldsymbol{w}$ on sample X

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} E(\boldsymbol{w} \mid X)$$

- Gradient

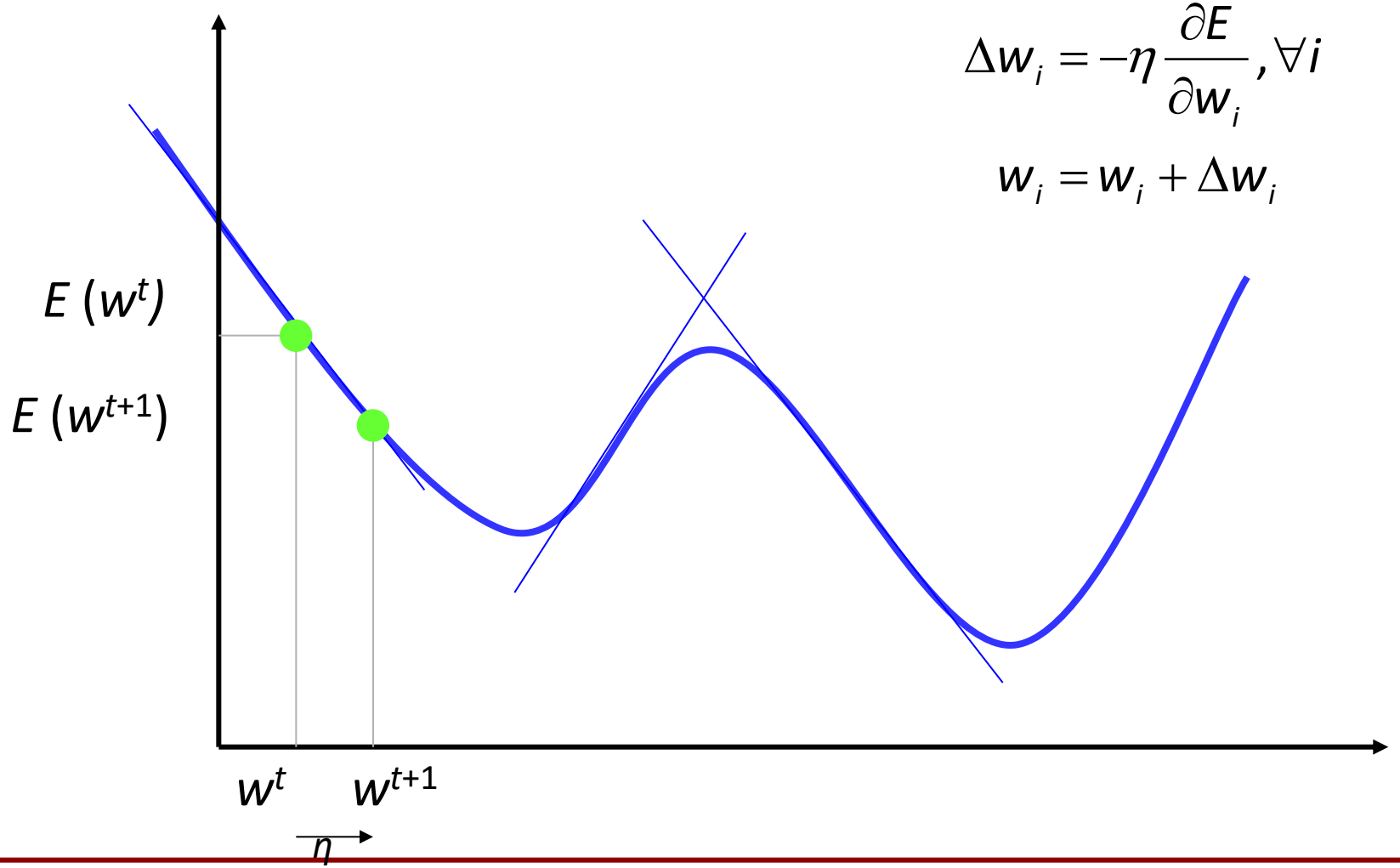
$$\nabla_w E = \left[\frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \ldots, \frac{\partial E}{\partial w_d} \right]^T$$

- Gradient-descent:
  Starts from random $\boldsymbol{w}$ and updates $\boldsymbol{w}$ iteratively in the negative direction of gradient

Gradient-Descent

$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i}, \forall i$$

$$w_i = w_i + \Delta w_i$$



Logistic Discrimination

Two classes: Assume log likelihood ratio is linear

$$\log\frac{p(\mathbf{x}\mid C_1)}{p(\mathbf{x}\mid C_2)} = \mathbf{w}^T\mathbf{x} + w_0^o$$

$$\begin{aligned}\text{logit}(P(C_1\mid\mathbf{x})) &= \log\frac{P(C_1\mid\mathbf{x})}{1-P(C_1\mid\mathbf{x})} = \log\frac{p(\mathbf{x}\mid C_1)}{p(\mathbf{x}\mid C_2)} + \log\frac{P(C_1)}{P(C_2)}\\ &= \mathbf{w}^T\mathbf{x} + w_0\end{aligned}$$

$$\text{where } w_0 = w_0^o + \log\frac{P(C_1)}{P(C_2)}$$

$$y = \hat{P}(C_1\mid\mathbf{x}) = \frac{1}{1+\exp\left[-\left(\mathbf{w}^T\mathbf{x}+w_0\right)\right]}$$

Training: Two Classes

$$\mathcal{X} = \left\{\mathbf{x}^t, r^t\right\}_t \quad r^t \mid \mathbf{x}^t \sim \text{Bernoulli}\left(y^t\right)$$

$$y = P\left(C_1 \mid \mathbf{x}\right) = \frac{1}{1+\exp\left[-\left(\mathbf{w}^T\mathbf{x}+w_0\right)\right]}$$

$$l\left(\mathbf{w}, w_0 \mid \mathcal{X}\right) = \prod_t \left(y^t\right)^{\left(r^t\right)}\left(1-y^t\right)^{\left(1-r^t\right)}$$

$$E = -\log l$$

$$E\left(\mathbf{w}, w_0 \mid \mathcal{X}\right) = -\sum_t r^t \log y^t + \left(1-r^t\right)\log\left(1-y^t\right)$$

Training: Gradient-Descent

$$E(\mathbf{w}, w_0 \mid \mathcal{X}) = -\sum_t r^t \log y^t + (1 - r^t) \log (1 - y^t)$$

$$\text{If } y = \text{sigmoid}(a) \quad \frac{dy}{da} = y(1-y)$$

$$\Delta w_j = -\eta \frac{\partial E}{\partial w_j} = \eta \sum_t \left(\frac{r^t}{y^t} - \frac{1 - r^t}{1 - y^t} \right) y^t (1 - y^t) x_j^t$$

$$= \eta \sum_t (r^t - y^t) x_j^t, j = 1, \ldots, d$$

$$\Delta w_0 = -\eta \frac{\partial E}{\partial w_0} = \eta \sum_t (r^t - y^t)$$

For $j = 0, \ldots, d$

$w_j \leftarrow$ rand(-0.01,0.01)

Repeat

For $j = 0, \ldots, d$

$\Delta w_j \leftarrow 0$

For $t = 1, \ldots, N$

$o \leftarrow 0$

For $j = 0, \ldots, d$

$o \leftarrow o + w_j x_j^t$

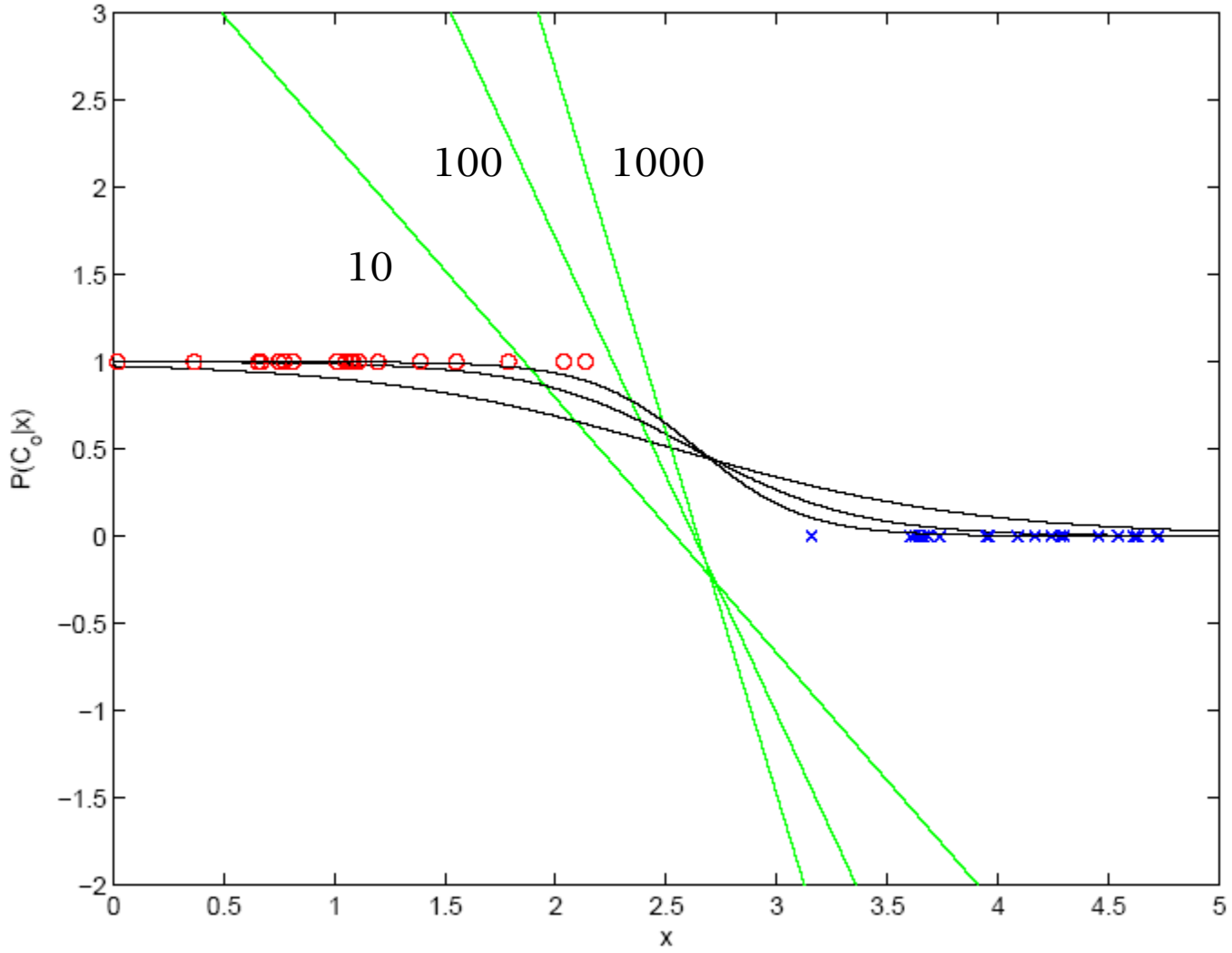
$y \leftarrow \text{sigmoid}(o)$

$\Delta w_j \leftarrow \Delta w_j + (r^t - y) x_j^t$

For $j = 0, \ldots, d$

$w_j \leftarrow w_j + \eta \Delta w_j$

Until convergence


100
1000
10
P(C_o|x)
x
3
2.5
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5

$K>2$ Classes

$$\mathcal{X}=\left\{\mathbf{x}^t,\mathbf{r}^t\right\}_t \quad r^t \mid \mathbf{x}^t \sim \text{Mult}_K\left(1,\mathbf{y}^t\right)$$

$$\log\frac{p\left(\mathbf{x}\mid C_i\right)}{p\left(\mathbf{x}\mid C_K\right)}=\mathbf{w}_i^T\mathbf{x}+w_{i0}^o$$

$$y_i=\hat{P}\left(C_i\mid\mathbf{x}\right)=\frac{\exp\left[\mathbf{w}_i^T\mathbf{x}+w_{i0}\right]}{\sum_{j=1}^{K}\exp\left[\mathbf{w}_j^T\mathbf{x}+w_{j0}\right]},i=1,\ldots,K$$

softmax

$$l\left(\left\{\mathbf{w}_i,w_{i0}\right\}_i\mid\mathcal{X}\right)=\prod_t\prod_i\left(y_i^t\right)^{\left(r_i^t\right)}$$

$$E\left(\left\{\mathbf{w}_i,w_{i0}\right\}_i\mid\mathcal{X}\right)=-\sum_t r_i^t\log y_i^t$$

$$\Delta\mathbf{w}_j=\eta\sum_t\left(r_j^t-y_j^t\right)\mathbf{x}^t \quad \Delta w_{j0}=\eta\sum_t\left(r_j^t-y_j^t\right)$$

For $i = 1, \ldots, K$, For $j = 0, \ldots, d$, $w_{ij} \leftarrow \text{rand}(-0.01, 0.01)$

Repeat

  For $i = 1, \ldots, K$, For $j = 0, \ldots, d$, $\Delta w_{ij} \leftarrow 0$

  For $t = 1, \ldots, N$

    For $i = 1, \ldots, K$

      $o_i \leftarrow 0$

      For $j = 0, \ldots, d$

        $o_i \leftarrow o_i + w_{ij} x_j^t$

    For $i = 1, \ldots, K$

      $y_i \leftarrow \exp(o_i) / \sum_k \exp(o_k)$

    For $i = 1, \ldots, K$

      For $j = 0, \ldots, d$

        $\Delta w_{ij} \leftarrow \Delta w_{ij} + (r_i^t - y_i) x_j^t$

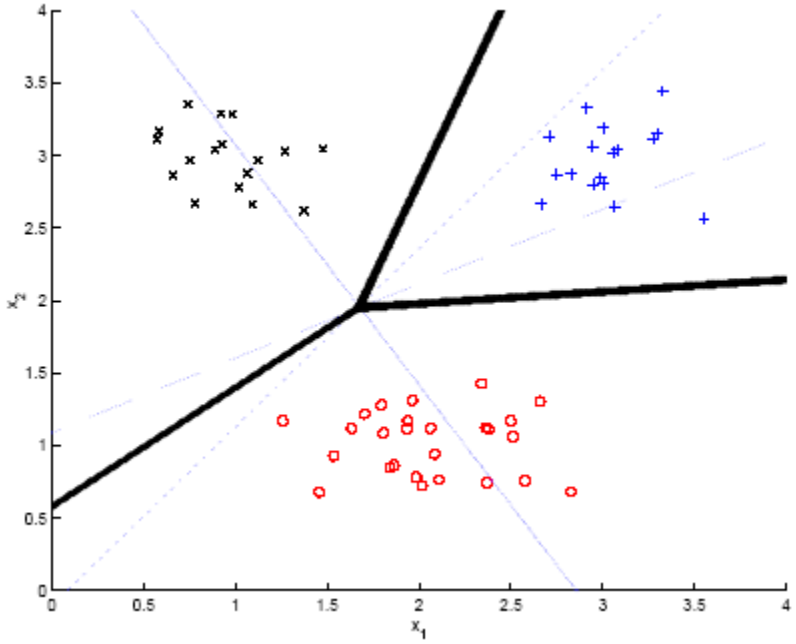
  For $i = 1, \ldots, K$

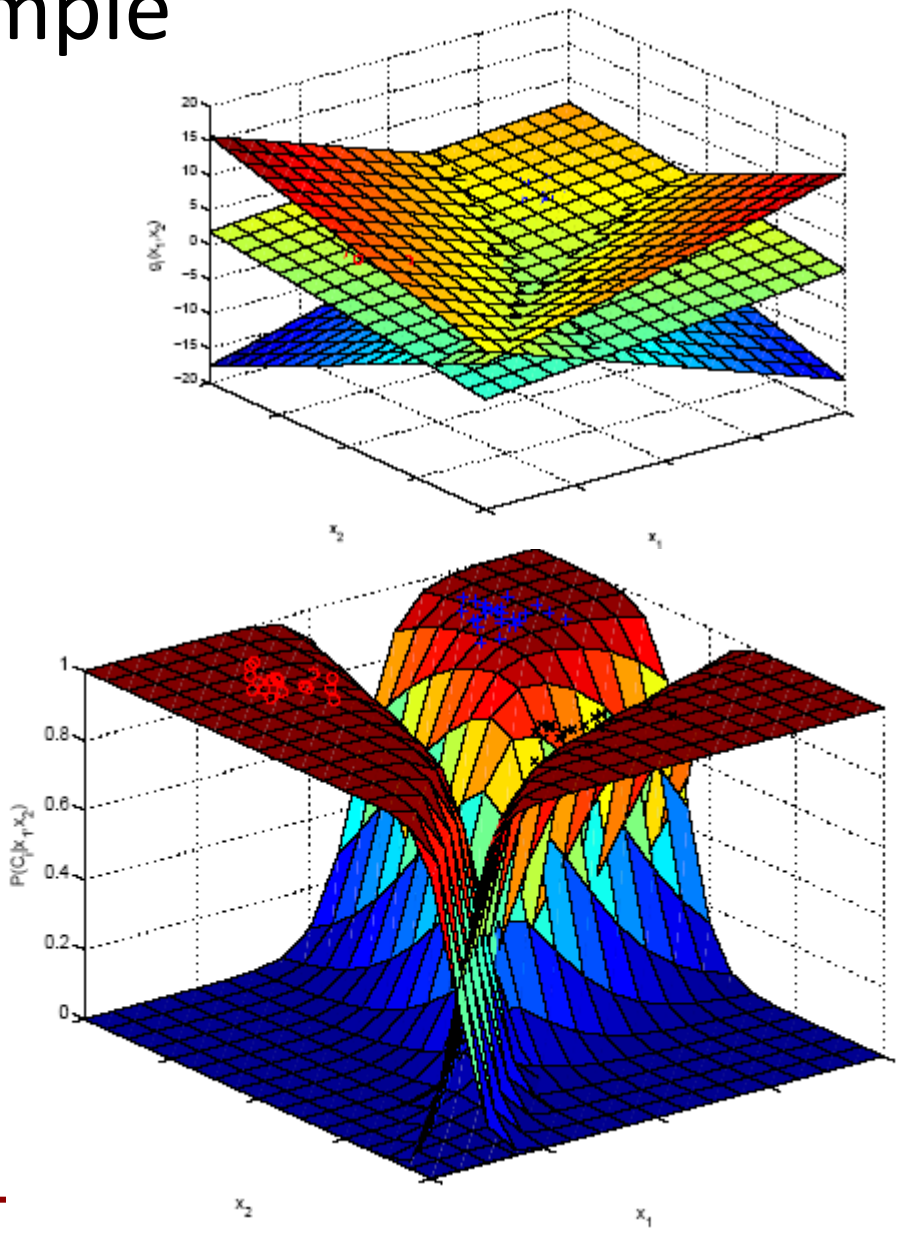
    For $j = 0, \ldots, d$

      $w_{ij} \leftarrow w_{ij} + \eta \Delta w_{ij}$

Until convergence

Example





Generalizing the Linear Model

- Quadratic:

$$\log \frac{p(\mathbf{x}|C_i)}{p(\mathbf{x}|C_K)} = \mathbf{x}^T \mathbf{W}_i \mathbf{x} + \mathbf{w}_i^T \mathbf{x} + w_{i0}$$

- Sum of basis functions:

$$\log \frac{p(\mathbf{x}|C_i)}{p(\mathbf{x}|C_K)} = \mathbf{w}_i^T \phi(\mathbf{x}) + w_{i0}$$

where $\varphi(\boldsymbol{x})$ are basis functions. Examples:

  - Hidden units in neural networks (Chapters 11 and 12)
  - Kernels in SVM (Chapter 13)

Discrimination by Regression

- Classes are NOT mutually exclusive and exhaustive

$$r^t = y^t + \varepsilon \text{ where } \varepsilon \sim \mathcal{N}(0, \sigma^2)$$

$$y^t = \text{sigmoid}(\mathbf{w}^T \mathbf{x}^t + w_0) = \frac{1}{1 + \exp\left[-\left(\mathbf{w}^T \mathbf{x}^t + w_0\right)\right]}$$

$$l(\mathbf{w}, w_0 \mid \mathcal{X}) = \prod_t \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{\left(r^t - y^t\right)^2}{2\sigma^2}\right]$$

$$E(\mathbf{w}, w_0 \mid \mathcal{X}) = \frac{1}{2} \sum_t \left(r^t - y^t\right)^2$$

$$\Delta \mathbf{w} = \eta \sum_t \left(r^t - y^t\right) y^t \left(1 - y^t\right) \mathbf{x}^t$$