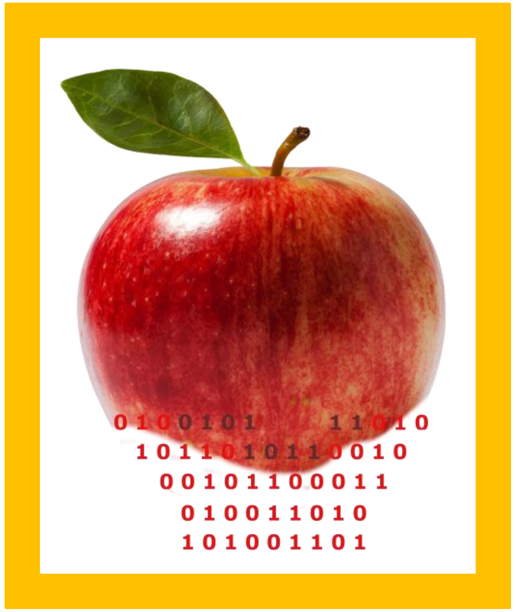
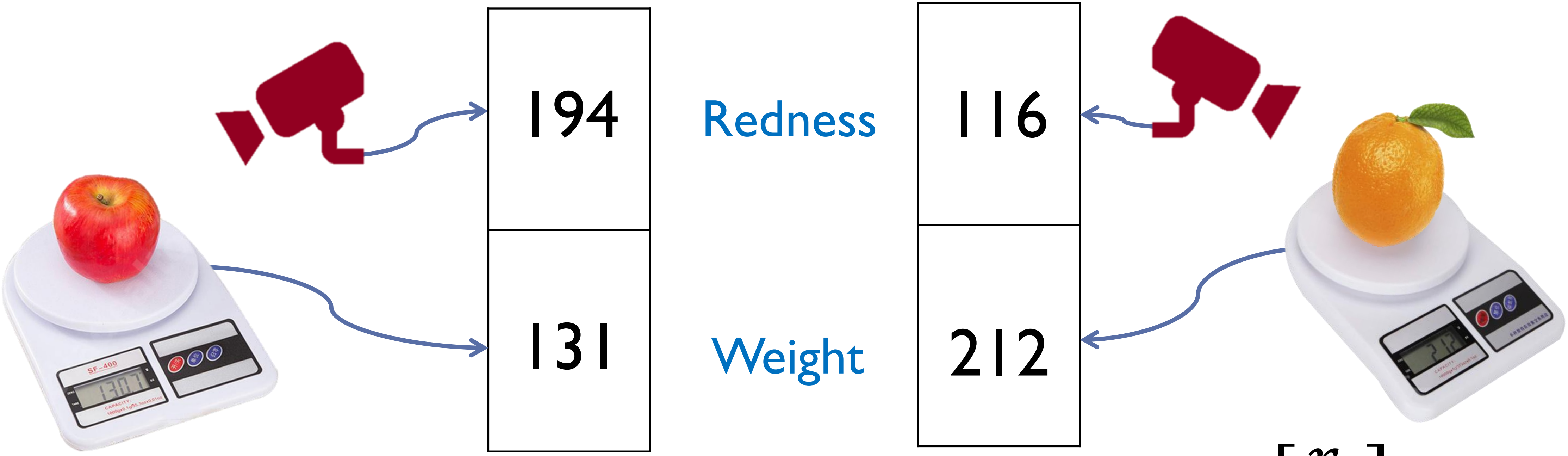


Data Representation and Classification

A RECAP

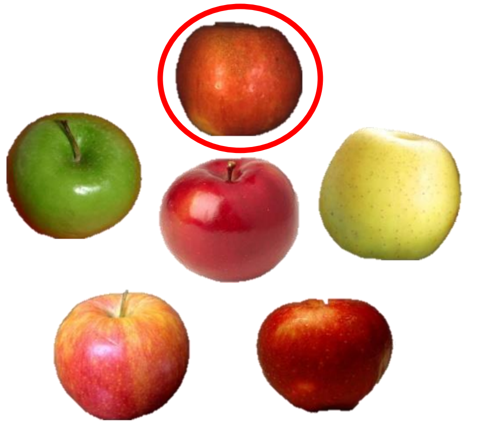
Characterising Apples and Oranges

- What makes Apples and Oranges different?

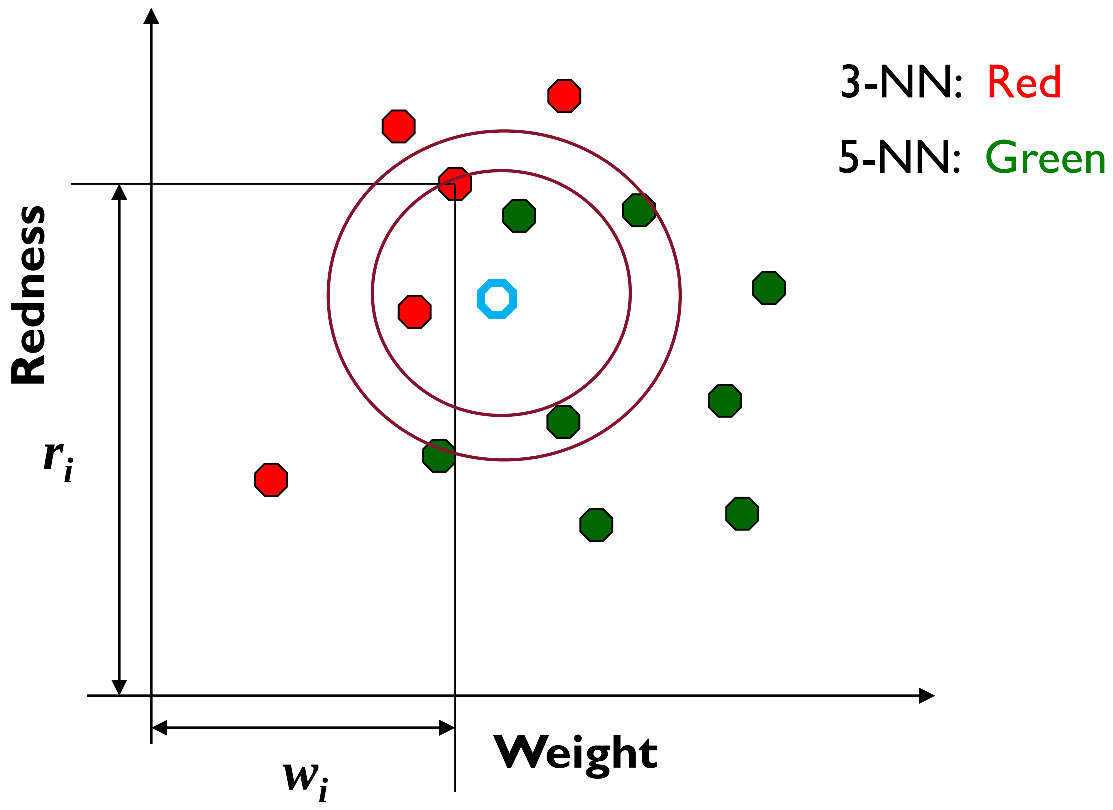


- Now each fruit is represented as a 2D vector: $\begin{bmatrix} r_i \\ w_i \end{bmatrix}$
- The components r_i and w_i are called features

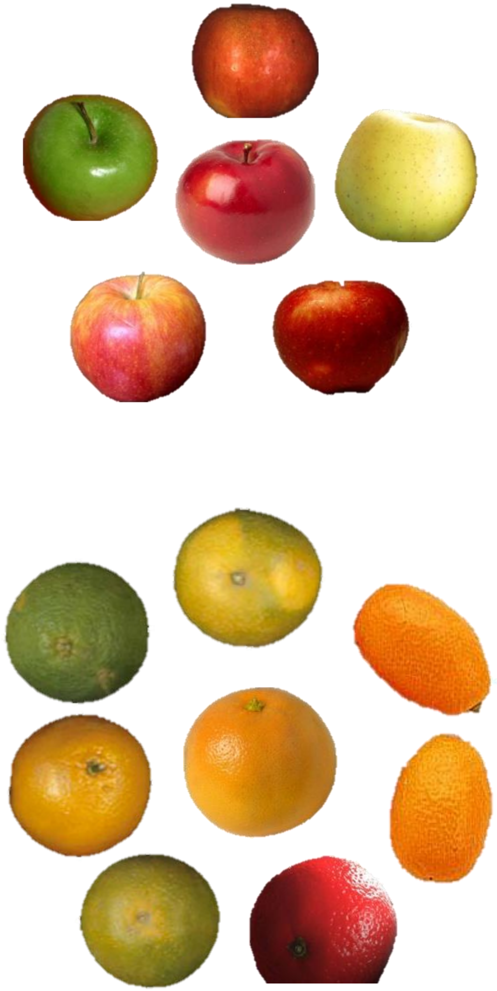
Feature Space: k-NN

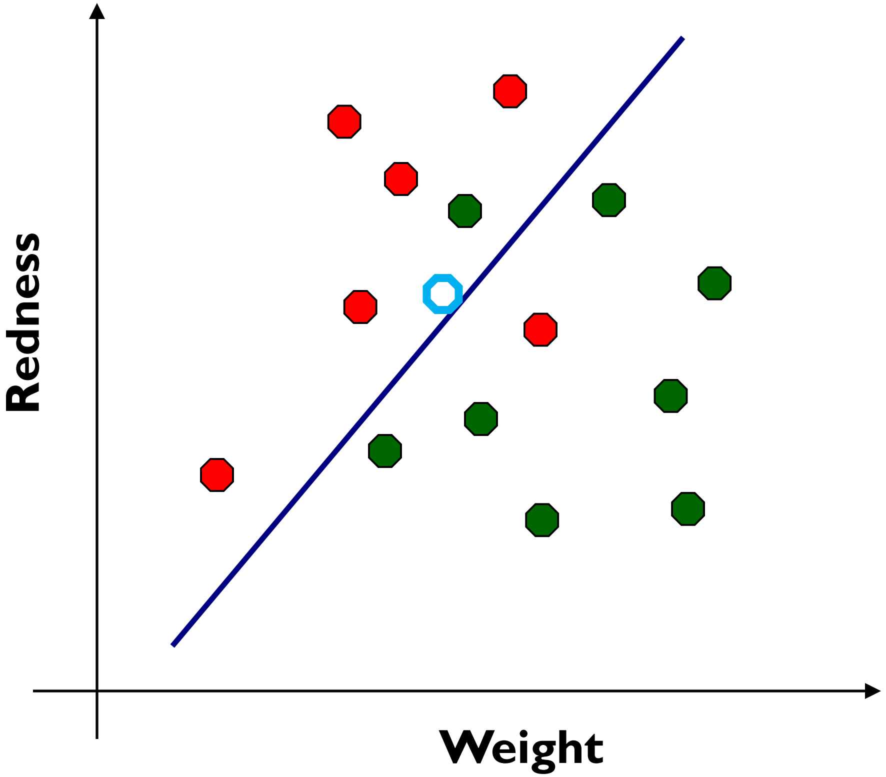


Feature Space Representation



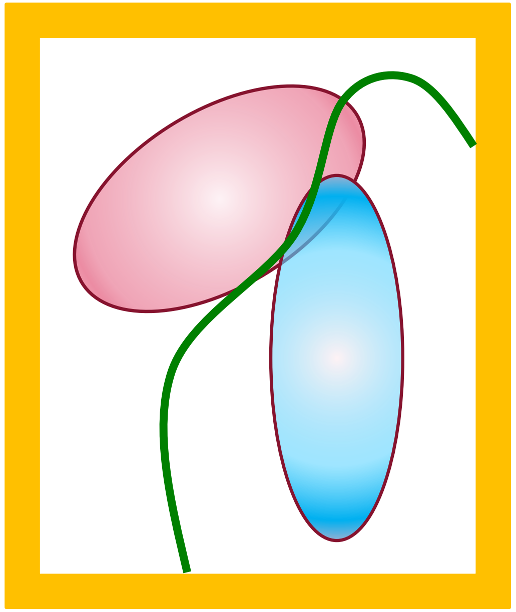
Feature Space: Linear Classifier





Feature Space Representation

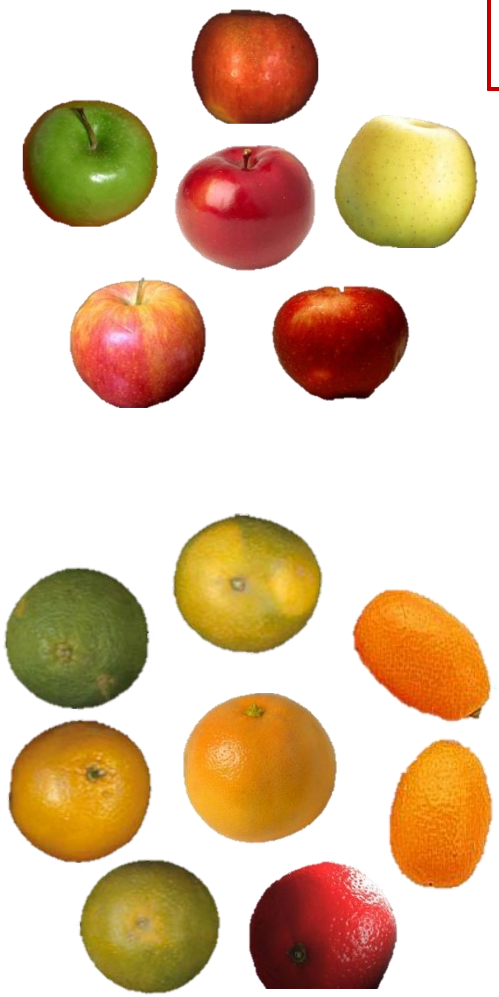
Questions?



Bayes Classifier

Finding the most likely answer

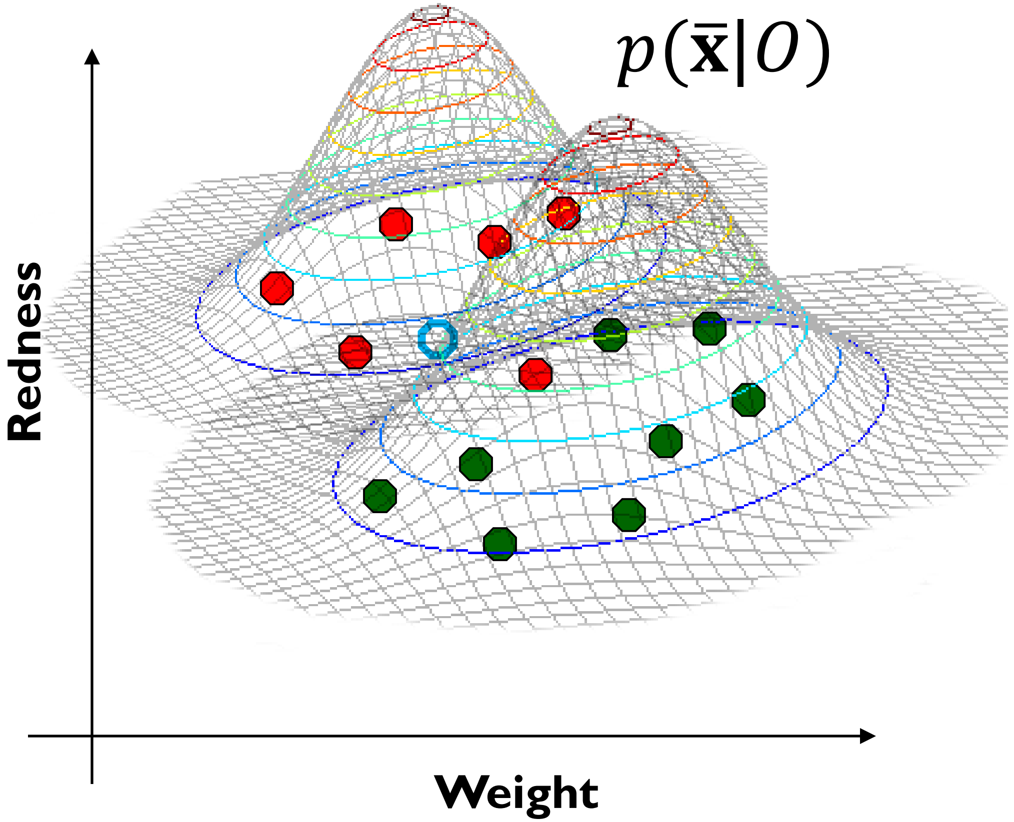
Feature Space: Bayes Classifier



$$\boxed{P(A|\bar{\mathbf{x}})\ \ vs\ P(O|\bar{\mathbf{x}})}$$

$p(\bar{\mathbf{x}}|A)$

$p(\bar{\mathbf{x}}|O)$



Feature Space Representation

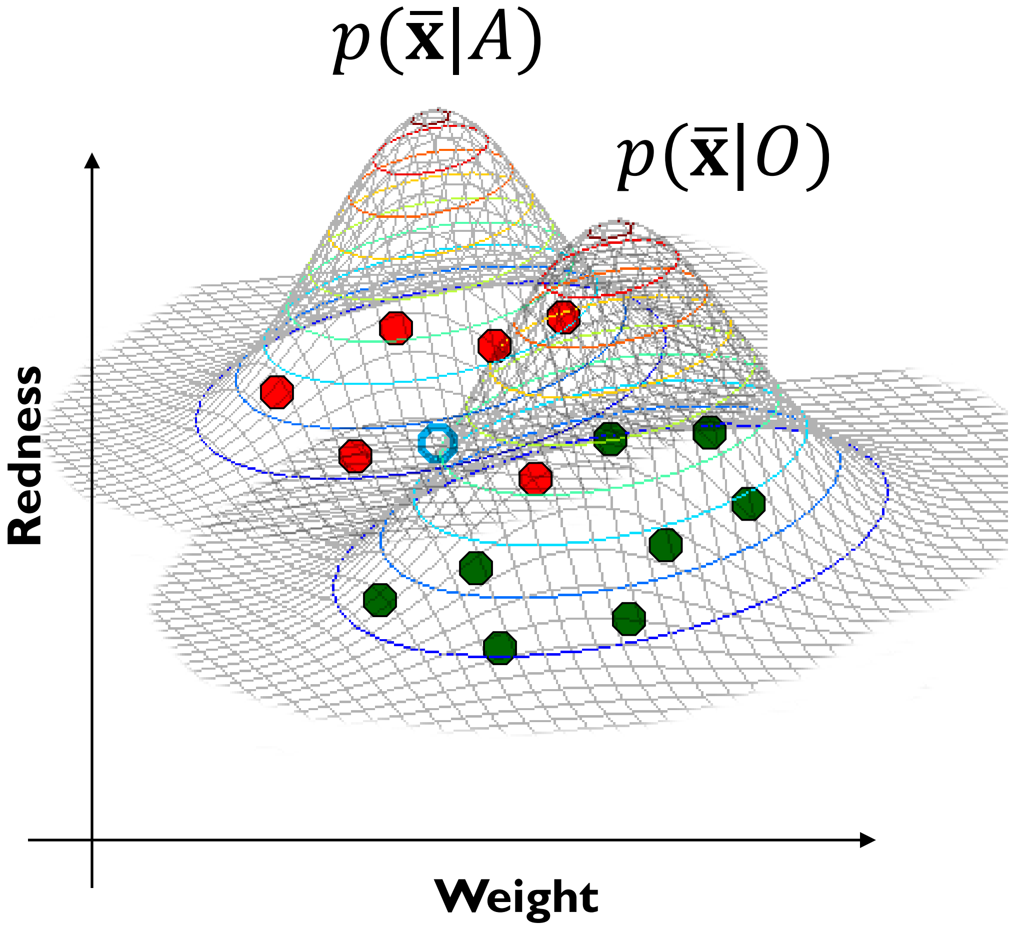
Feature Space: Bayes Classifier

- Compute the likelihoods:
- Compute the posteriors:

$$P(A|\bar{\mathbf{x}}) \;\; vs \; P(O|\bar{\mathbf{x}})$$

$$\frac{p(\bar{\mathbf{x}}|A) \times P(A)}{p(\bar{\mathbf{x}})} \; vs \; \frac{p(\bar{\mathbf{x}}|O) \times P(O)}{p(\bar{\mathbf{x}})}$$

- Assign class with highest posterior probability



Feature Space Representation

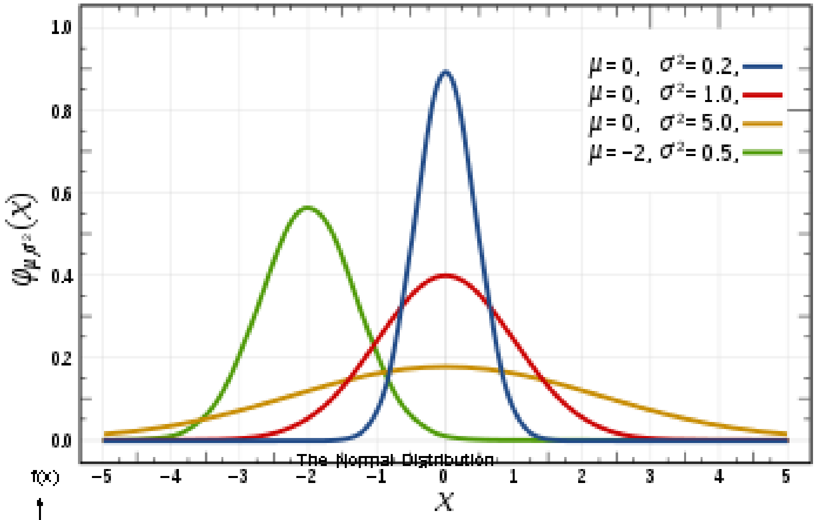
Estimating Density

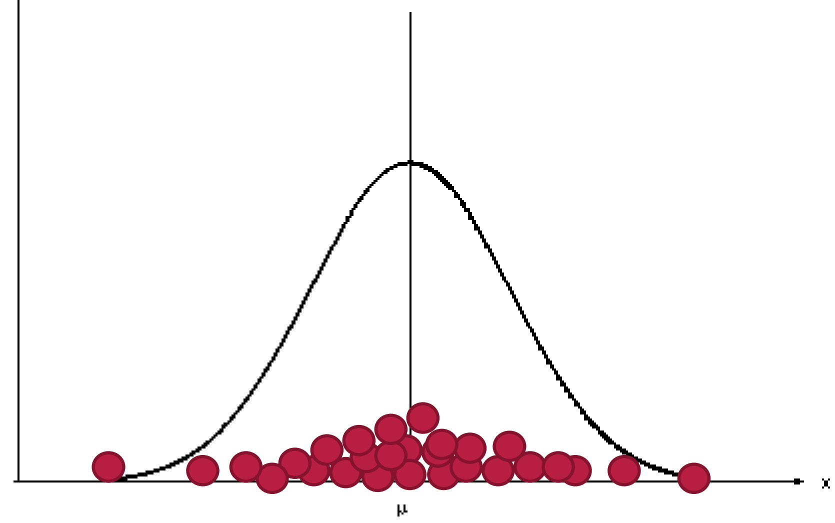
- Assume a density function: Say Gaussian (or Normal)

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

where μ is the mean and σ^2 is the variance

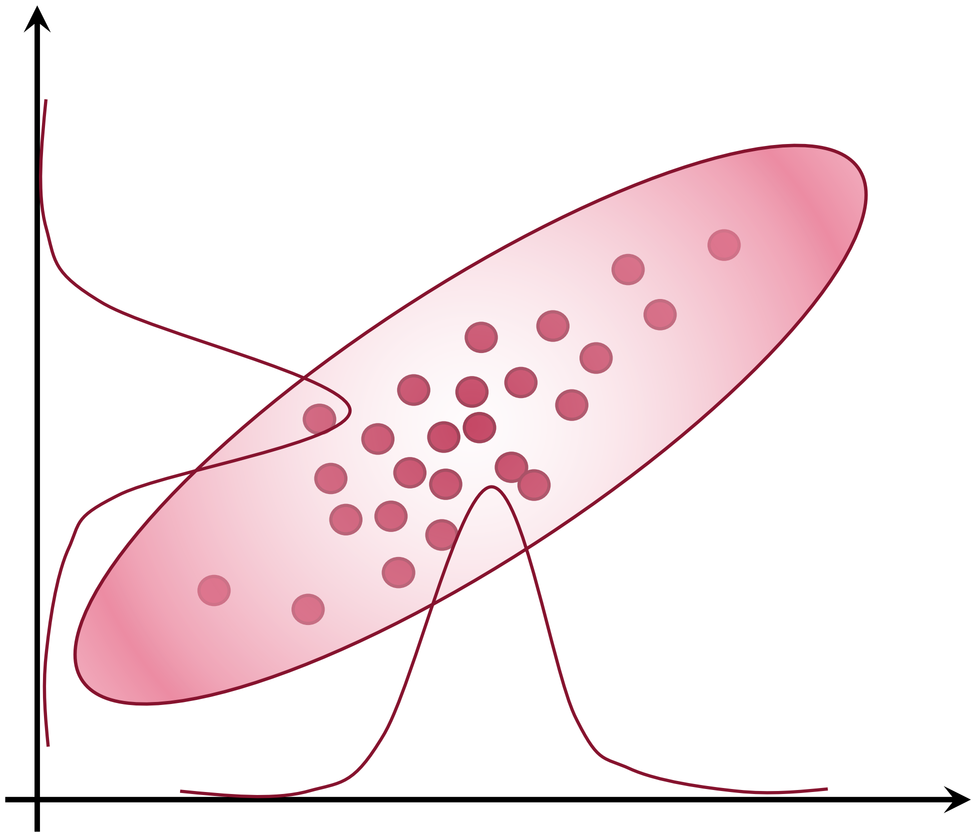
- There are 2 parameters to estimate from the training data points
- Given μ and σ, we can calculate $p(x|A)$ for any value of x.





What about Multiple Dimensions?

- We have variances along each dimension
- The samples also co-vary.
  i.e, features are not independent
- Captured using a covariance matrix



$$\begin{bmatrix} V_a & C_{a,b} & C_{a,c} & C_{a,d} & C_{a,e} \\ C_{a,b} & V_b & C_{b,c} & C_{b,d} & C_{b,e} \\ C_{a,c} & C_{b,c} & V_c & C_{c,d} & C_{c,e} \\ C_{a,d} & C_{b,d} & C_{c,d} & V_d & C_{d,e} \\ C_{a,e} & C_{b,e} & C_{c,e} & C_{d,e} & V_e \end{bmatrix}$$

$$\hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i$$

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{N}\sum_{i=1}^{N} (\mathbf{x}_i - \hat{\boldsymbol{\mu}})(\mathbf{x}_i - \hat{\boldsymbol{\mu}})^\top$$

Likelihood Function

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

becomes

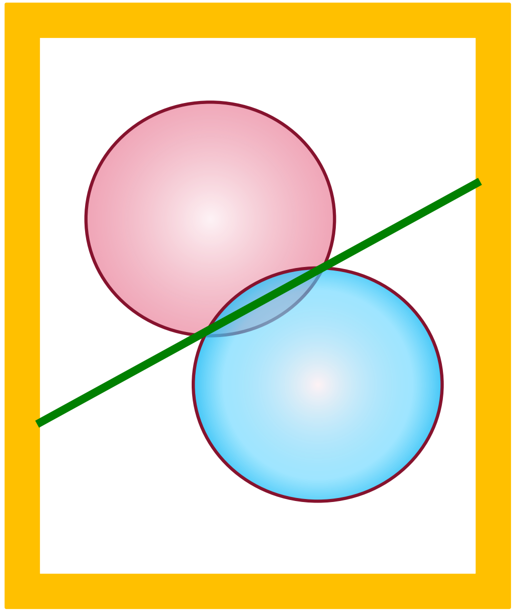
- $N(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})}$

Challenge of Data

$$\begin{bmatrix} V_a & C_{a,b} & C_{a,c} & C_{a,d} & C_{a,e} \\ C_{a,b} & V_b & C_{b,c} & C_{b,d} & C_{b,e} \\ C_{a,c} & C_{b,c} & V_c & C_{c,d} & C_{c,e} \\ C_{a,d} & C_{b,d} & C_{c,d} & V_d & C_{d,e} \\ C_{a,e} & C_{b,e} & C_{c,e} & C_{d,e} & V_e \end{bmatrix}$$

- 1-dim had 2 parameters to estimate
- d-dim will have not just $2d$, but over $d^2/2$ parameters.

Questions?



Naïve Bayes Classifier

Simplifying Densities

Solving the Challenge

- Assume Σ to be diagonal
- i.e., features are independent
- We lose some information about the data

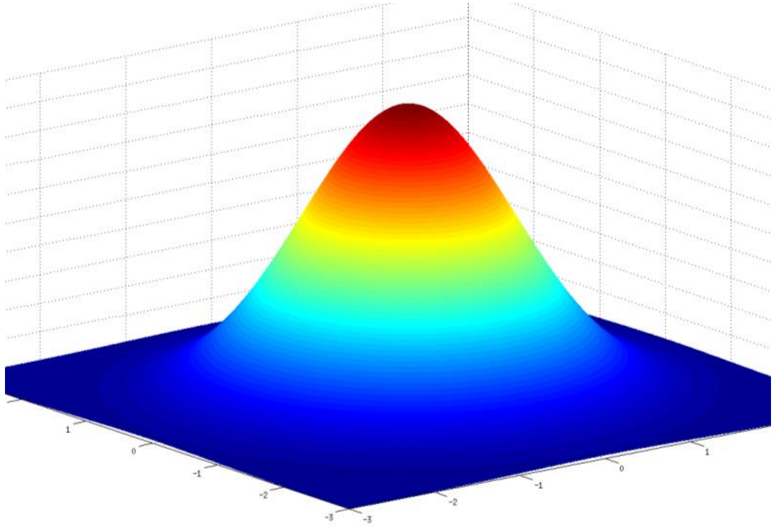
$$\begin{bmatrix} V_a & & & & \\ & V_b & & 0 & \\ & & V_c & & \\ & 0 & & V_d & \\ & & & & V_e \end{bmatrix}$$

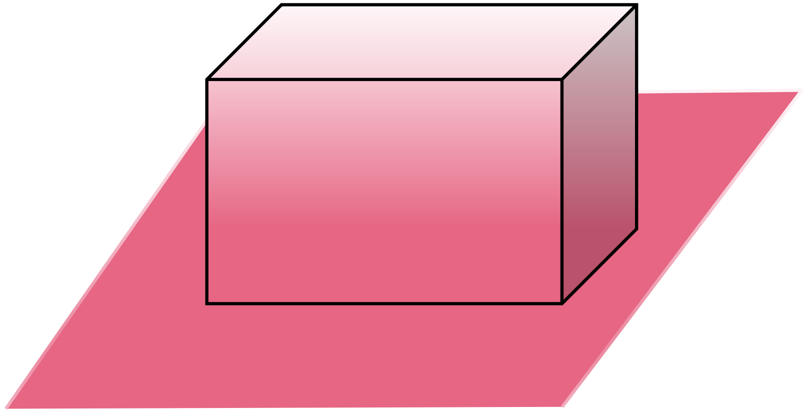
Simpler likelihood

- $p(\mathbf{x}|A) = p(x_1|A) \times p(x_2|A) \times p(x_3|A) \times \cdots \times p(x_d|A)$
- A multivariate density $p(\mathbf{x}|A)$ is approximated with the product of d univariate densities: $p(x_i|A)$.
- Equivalent to assuming diagonal covariance for Normal density.
- Otherwise, Naïve Bayes classifier is same as a regular Bayesian Classifier

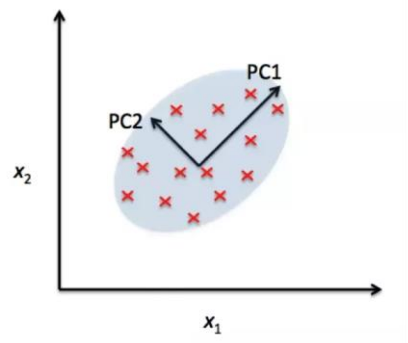
Naïve Bayes: Summary and Comments

- Bayes Classifier with feature independence
- Multivariate density $p(\mathbf{x}|A)$ is approximated with the product of d univariate densities: $p(x_i|A)$.
- Compute the parameters (μ and σ for Normal) of each feature independently, thus estimating $p(x_i|A)$ for each feature i.
- Number of parameters to be estimated is reduced back to 2d for Normal desnsity
- This is true for any density function, not just Normal density





Questions?



Principal Component Analysis (PCA)

Simplifying Representations

Selecting Features as Matrix Multiplication

Selecting first and third feature

$$\begin{bmatrix} x_1 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

Selecting first and fourth feature

$$\begin{bmatrix} x_1 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

$$\mathbf{x}' = \mathbf{A}\mathbf{x}$$

A “new” set of features are selected/extracted from the original one by a matrix multiplication.

Rows of **A** decide what the new features are. (They need not be 0 and 1.)

Often #rows of A is smaller than #columns of A. This is also called **dimensionality reduction**.

Feature Selection and Extraction

- Selection:
  - Select some features out of a pool. (Simple A with 0/1.)
- Extraction:
  - Extract a set of new features. (elements of A need not be 0/1)
- Extraction is often required:
  - To visualize in 2D/3D.
  - To remove some “useless” or “less useful” features.
  - Make computations efficient. (Note: original data could be 1000s of dimension!!)

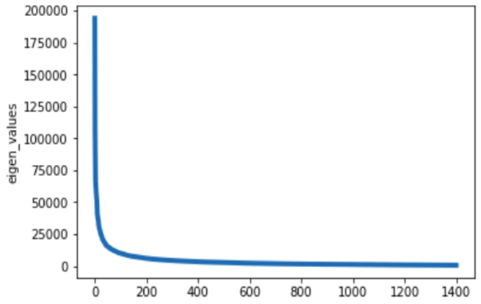
PCA based Feature Extraction

K X 1

=

M is a K X d matrix

d X 1



Plot of Eigen values in decreasing order.

Usually only "K" (a very small are useful). Most of them are near zero or even zero.

Eg. K = 500 d = 3072

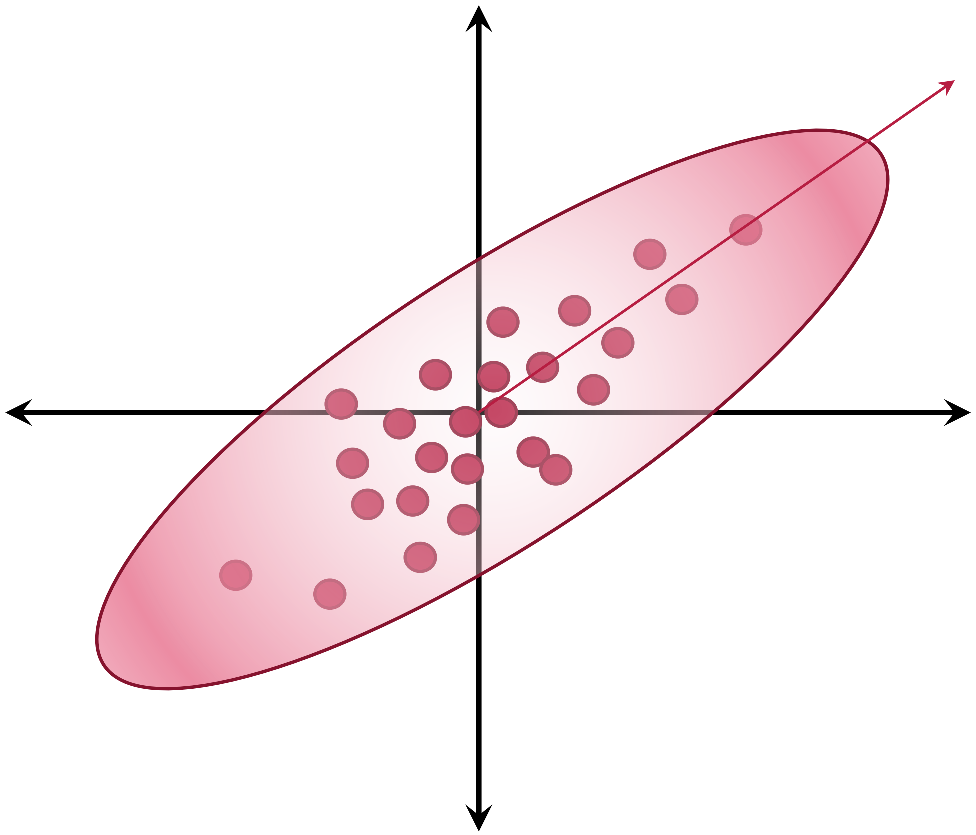
Let $\mathbf{x}_i$ be an image represented as a column vector.

Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \text{---}, \mathbf{x}_N]$ be zero mean a d X N matrix.
(Here d = 3072, N = 10K)
Let $\mathbf{A} = \mathbf{X}\mathbf{X}^T$ be an d X d Matrix. It also has d Eigen vectors each of d dimension.

Rows of matrix "**M**" are the selected K Eigen vectors of the matrix **A**.

Projection of Point to a Line

- Consider a line through origin
  - Vector along the line
- What does dot product mean?
- Dot product with which vector? Vector along the line or line coefficients?
  - We consider dot product with vector along the line here
  - We talked about dot product with the coefficient vector for linear classifiers

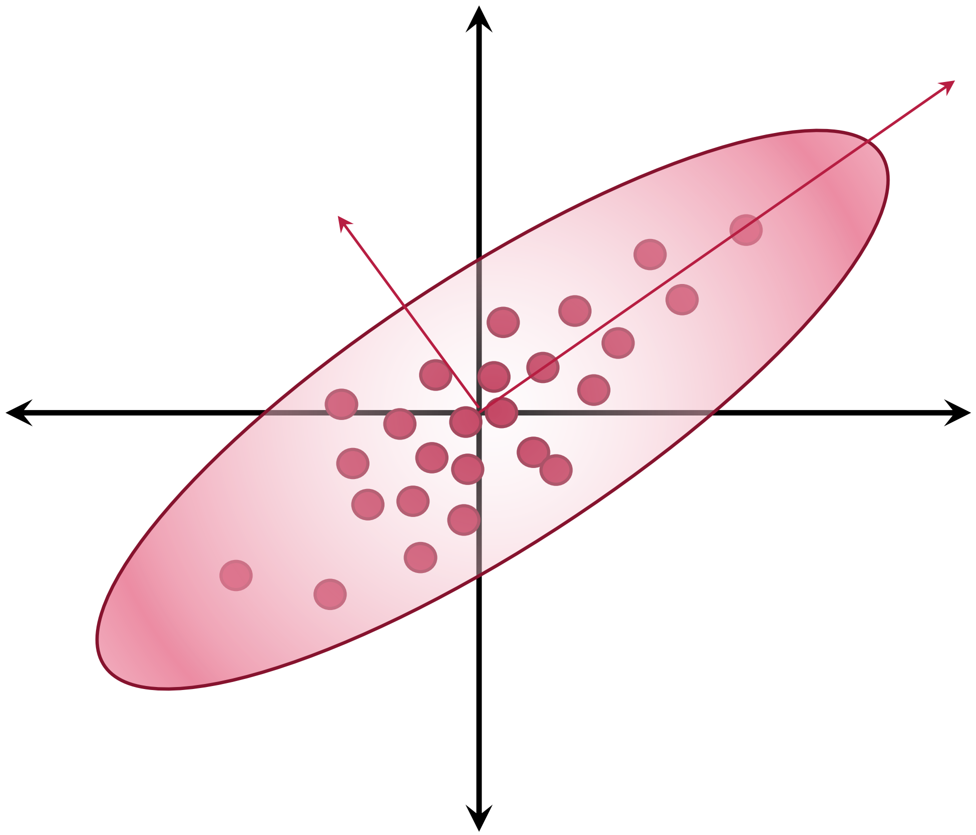


PCA and Covariance Matrix

- Going from 2D to 1D
- Which feature to select?
  - This may be any feature (or vector in any direction)
- Two view points:
  - Maximize variance
  - Minimize error
- Solution to both happens to be:

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$$

- What is A? How do we solve this?



Questions?