Bayes' Theorem

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

- $p(Y=y)$ is the ***prior probability***: it expresses the probability ***before*** we observe any data
- $p(Y=y\mid X=x)$ is the ***posterior probability***: it expresses the probability ***after*** we observed the data
- The effect of the observed data is captured through the conditional probability $p(X=x\mid Y=y)$

Curve fitting re-visited

- We can adopt a Bayesian approach when estimating the parameters $\boldsymbol{w}$ for polynomial curve fitting.
- $p(\boldsymbol{w})$ captures our assumptions about $\boldsymbol{w}$ before observing the data.
- The effect of the observed data D is captured by the conditional probability $p(D\mid \boldsymbol{w})$
- Bayes' theorem allows to evaluate the uncertainty in $\boldsymbol{w}$ ***after*** we have observed the data D (in the form of posterior probability): $$p(\boldsymbol{w}\mid D)=\frac{p(D\mid \boldsymbol{w})\,p(\boldsymbol{w})}{p(D)}$$
- $p(D\mid \boldsymbol{w})$ is the ***likelihood function***
- ***Maximum likelihood*** approach: set $\boldsymbol{w}$ to the value that maximizes $p(D\mid \boldsymbol{w})$

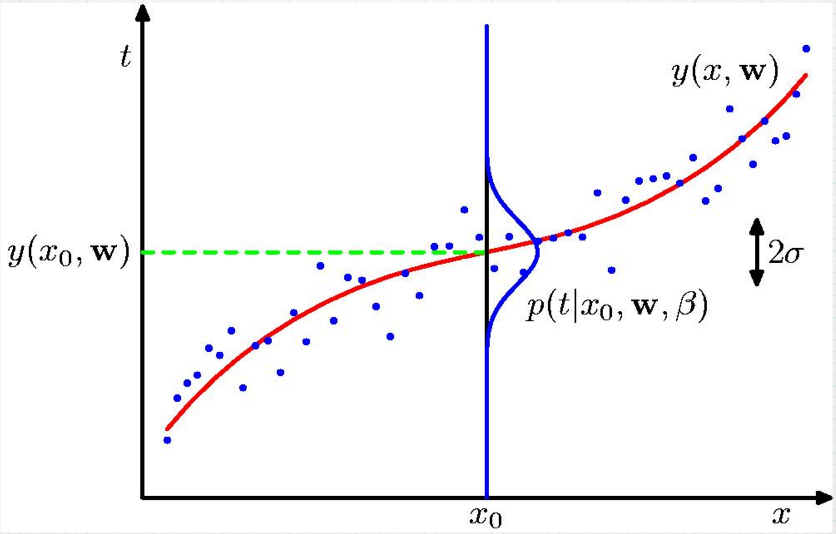
Curve fitting re-visited: ML approach

- Training data: $\boldsymbol{x} = (x_1,\cdots,x_N)^T, \boldsymbol{t} = (t_1,\cdots t_N)^T$
- We can express our uncertainty over the value of the target variable using a probability distribution
- Assumption: Given a value of x, the corresponding value of t has a *Gaussian* distribution with a mean equal to

$$y(x,\boldsymbol{w}) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

- Thus: $p(t \mid x,\boldsymbol{w},\beta) = \mathrm{N}\left(t \mid y(x,\boldsymbol{w}),\beta^{-1}\right)$

Curve fitting re-visited: ML approach



Curve fitting re-visited: ML approach

- We use the training data $\{\boldsymbol{x},\boldsymbol{t}\}$ to estimate $\boldsymbol{w},\beta$ by maximum likelihood

- Assuming data are drawn *independently*, the likelihood function can be written as the product of the *marginal distributions*:

$$p(t \mid x,\boldsymbol{w},\beta) = \prod_{n=1}^{N} \mathrm{N}\left(t_n \mid y(x_n,\boldsymbol{w}),\beta^{-1}\right)$$

Curve fitting re-visited: ML approach

- Gaussian distribution: $\mathrm{N}\left(x \mid \mu,\sigma^2\right) = \dfrac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

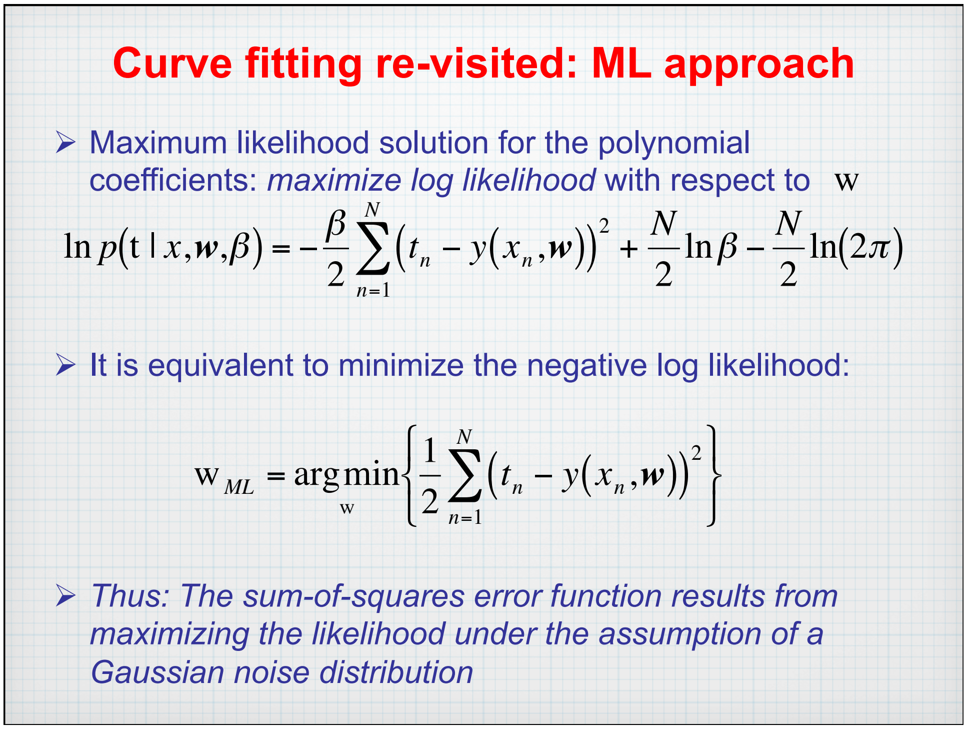
$$\mu = y(x,\mathrm{w}),\sigma^2 = \beta^{-1}$$

$$\Rightarrow \ln p(\mathrm{t} \mid x,\boldsymbol{w},\beta) = \ln\prod_{n=1}^{N} \mathrm{N}\left(t_n \mid y(x_n,\boldsymbol{w}),\beta^{-1}\right)$$

$$= \ln\prod_{n=1}^{N} \frac{1}{\sqrt{2\pi\beta^{-1}}} e^{\frac{-(t_n - y(x_n,\mathrm{w}))^2}{2\beta^{-1}}}$$

$$= -\frac{\beta}{2}\sum_{n=1}^{N}\left(t_n - y(x_n,\boldsymbol{w})\right)^2 - \sum_{n=1}^{N}\ln\sqrt{2\pi\beta^{-1}}$$

$$= -\frac{\beta}{2}\sum_{n=1}^{N}\left(t_n - y(x_n,\boldsymbol{w})\right)^2 + \frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi)$$



Curve fitting re-visited: ML approach

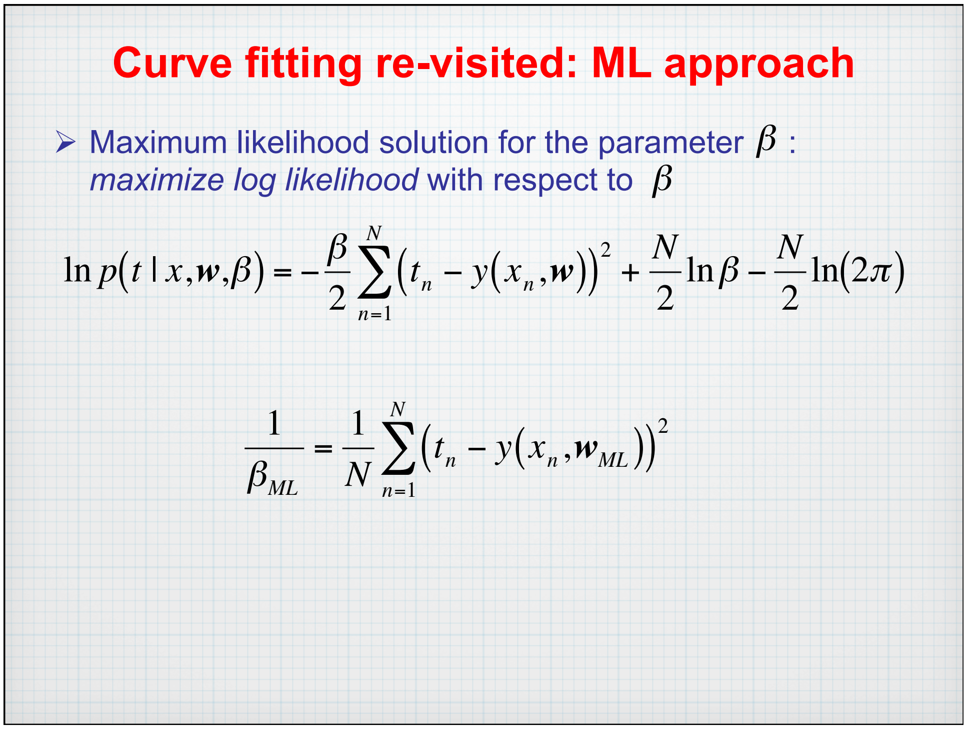
- Maximum likelihood solution for the polynomial coefficients: *maximize log likelihood* with respect to w

$$\ln p(\mathrm{t} \mid x, \boldsymbol{w}, \beta) = -\frac{\beta}{2}\sum_{n=1}^{N}\left(t_n - y(x_n, \boldsymbol{w})\right)^2 + \frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi)$$

- It is equivalent to minimize the negative log likelihood:

$$\mathrm{w}_{ML} = \arg\min_{\mathrm{w}}\left\{\frac{1}{2}\sum_{n=1}^{N}\left(t_n - y(x_n, \boldsymbol{w})\right)^2\right\}$$

- *Thus: The sum-of-squares error function results from maximizing the likelihood under the assumption of a Gaussian noise distribution*



Curve fitting re-visited: ML approach

- Maximum likelihood solution for the parameter β : *maximize log likelihood* with respect to β

$$\ln p(t \mid x, \boldsymbol{w}, \beta) = -\frac{\beta}{2}\sum_{n=1}^{N}\left(t_n - y(x_n, \boldsymbol{w})\right)^2 + \frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi)$$

$$\frac{1}{\beta_{ML}} = \frac{1}{N}\sum_{n=1}^{N}\left(t_n - y(x_n, \boldsymbol{w}_{ML})\right)^2$$

Curve fitting re-visited: ML approach

- We now have the maximum likelihood solutions for the parameters: $\mathrm{w}_{ML}, \beta_{ML}$
- We can now make predictions for new values of *x* by using the resulting probability distribution over *t* (*predictive distribution*)

$$p\left(t \mid x, \boldsymbol{w}_{ML}, \beta_{ML}\right) = \mathrm{N}\left(t \mid y\left(x, \boldsymbol{w}_{ML}\right), \beta_{ML}^{-1}\right)$$

Maximum a Posteriori (MAP) approach

- Let us introduce a prior distribution over the polynomial coefficients w
- Recall: Gaussian distribution of a D-dimensional vector $\boldsymbol{x}$

$$\mathrm{N}(x, \mu, \Sigma) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} e^{\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right)}$$

- Prior distribution:

$$p(\boldsymbol{w} \mid \alpha) = \mathrm{N}\left(\boldsymbol{w} \mid 0, \alpha^{-1} \boldsymbol{I}\right) = \left(\frac{\alpha}{2\pi}\right)^{(M+1)/2} e^{\left(-\frac{\alpha}{2} \boldsymbol{w}^T \boldsymbol{w}\right)}$$

- Using Bayes' theorem:

$$p(\boldsymbol{w} \mid \boldsymbol{x}, \boldsymbol{t}, \alpha, \beta) \propto p(\boldsymbol{t} \mid \boldsymbol{x}, \boldsymbol{w}, \beta) p(\boldsymbol{w} \mid \alpha)$$

MAP approach

- Maximum a Posteriori solution for the parameters $\boldsymbol{w}$ *maximize the posterior distribution*

$$p(\boldsymbol{w}\mid\boldsymbol{x},\boldsymbol{t},\alpha,\beta)\propto p(\boldsymbol{t}\mid\boldsymbol{x},\boldsymbol{w},\beta)\,p(\boldsymbol{w}\mid\alpha)$$

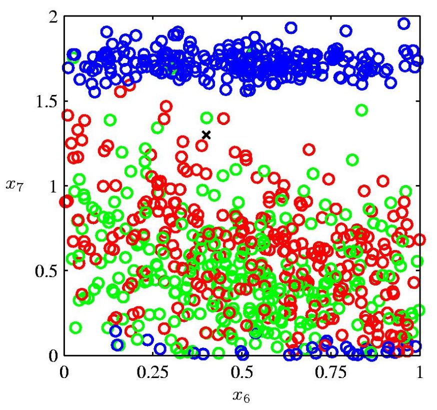
- It is equivalent to minimize the negative log posterior distribution:

$$\boldsymbol{w}_{MAP}=\arg\min_{\boldsymbol{w}}\left\{\frac{\beta}{2}\sum_{n=1}^{N}\left(t_n-y(x_n,\boldsymbol{w})\right)^2+\frac{\alpha}{2}\boldsymbol{w}^T\boldsymbol{w}\right\}$$

- *Thus: maximizing the posterior distribution is equivalent to minimizing the regularized sum-of-squares error function*

Curse of Dimensionality

- Real world applications deal with spaces with high dimensionality
- High dimensionality poses serious challenges for the design of pattern recognition techniques



Oil flaw data in two dimensions

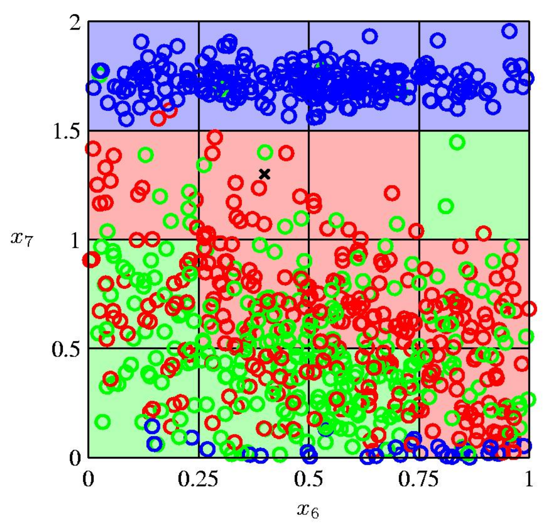
Red = homogeneous

Green = annular

Blue = laminar

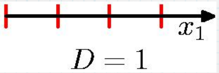
Curse of Dimensionality

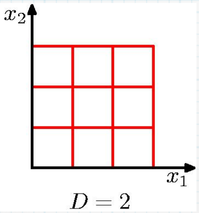
- A simple classification approach

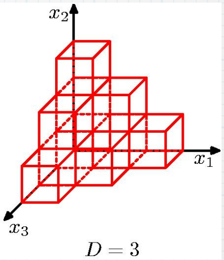


Curse of Dimensionality

- Going higher in dimensionality…

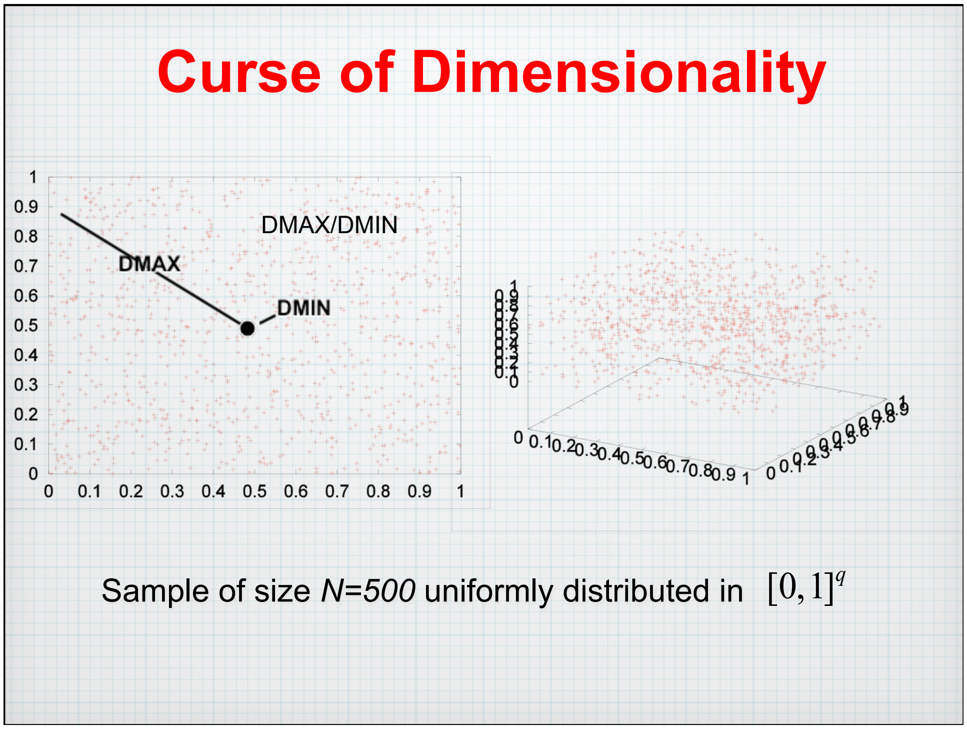






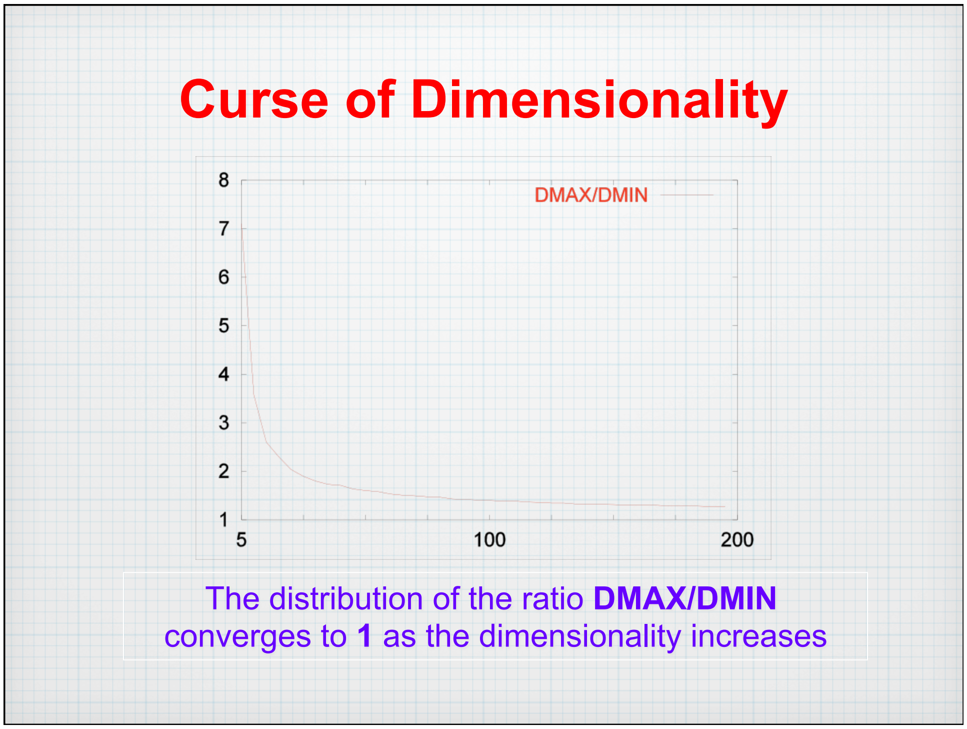
3^1 3^2 3^3

The number of regions of a regular grid grows exponentially with the dimensionality D of the space



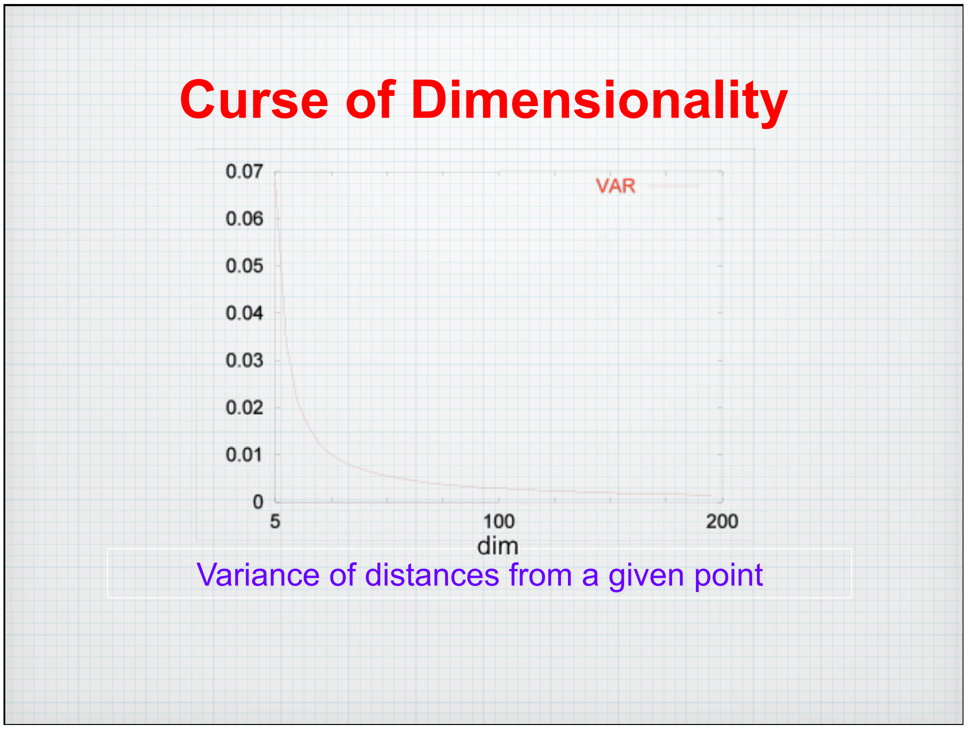
Curse of Dimensionality

Sample of size *N=500* uniformly distributed in $[0,1]^q$



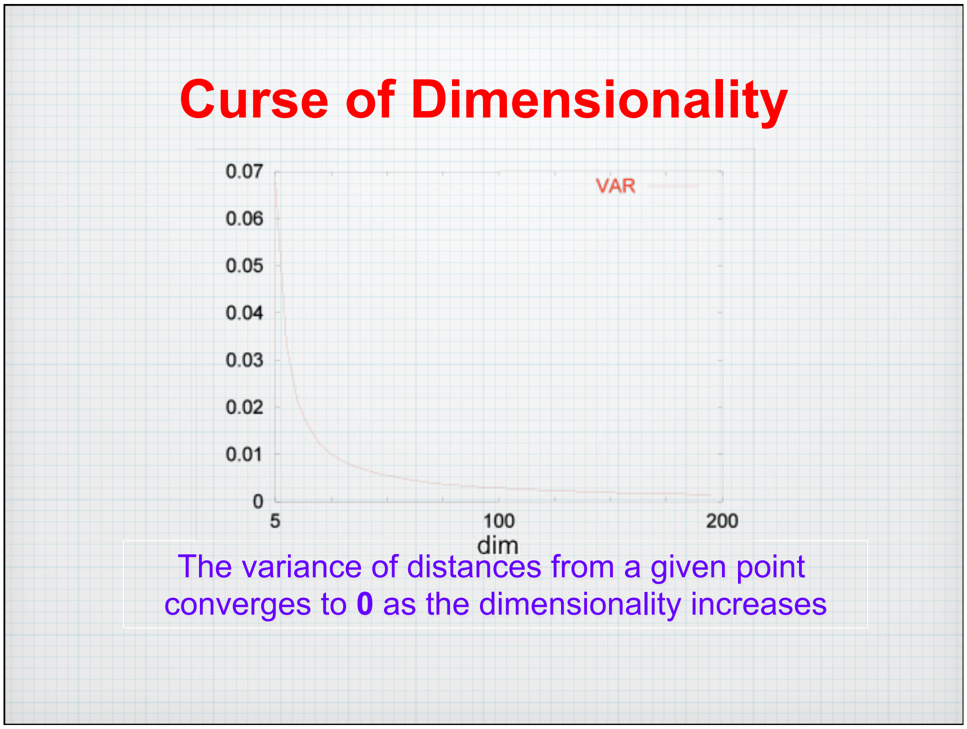
Curse of Dimensionality

The distribution of the ratio **DMAX/DMIN** converges to **1** as the dimensionality increases



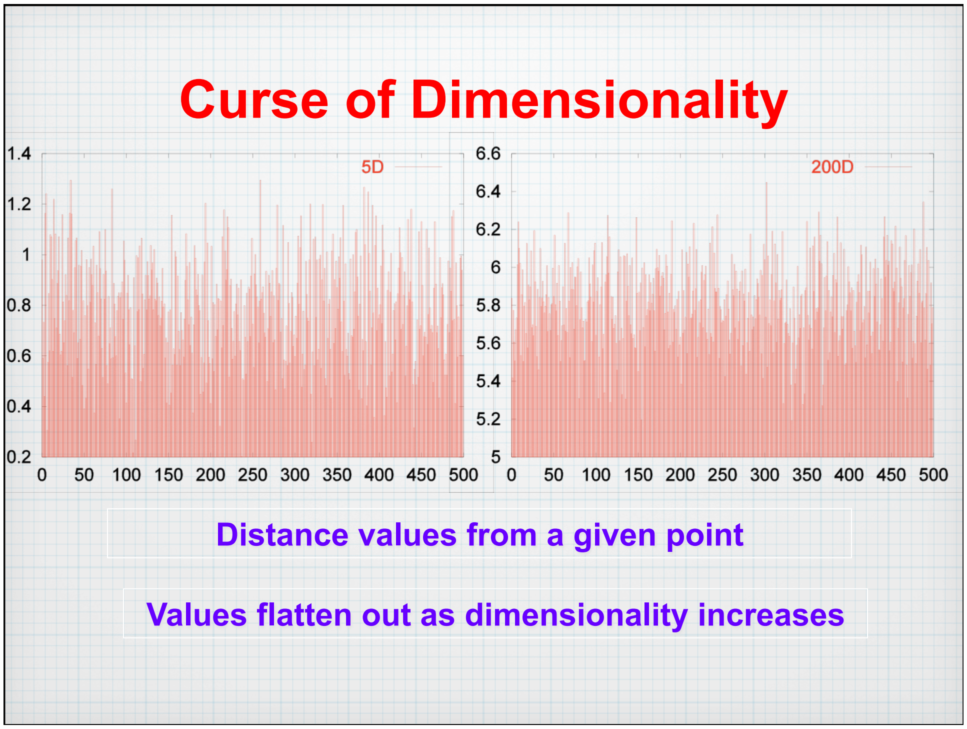
Curse of Dimensionality

Variance of distances from a given point



Curse of Dimensionality

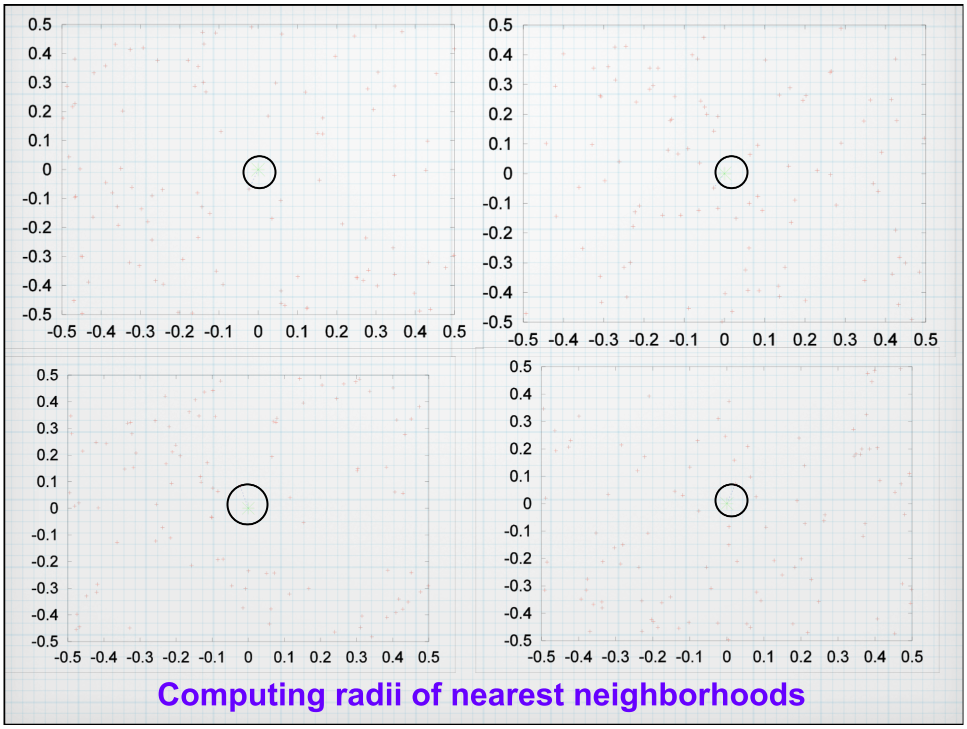
The variance of distances from a given point converges to **0** as the dimensionality increases



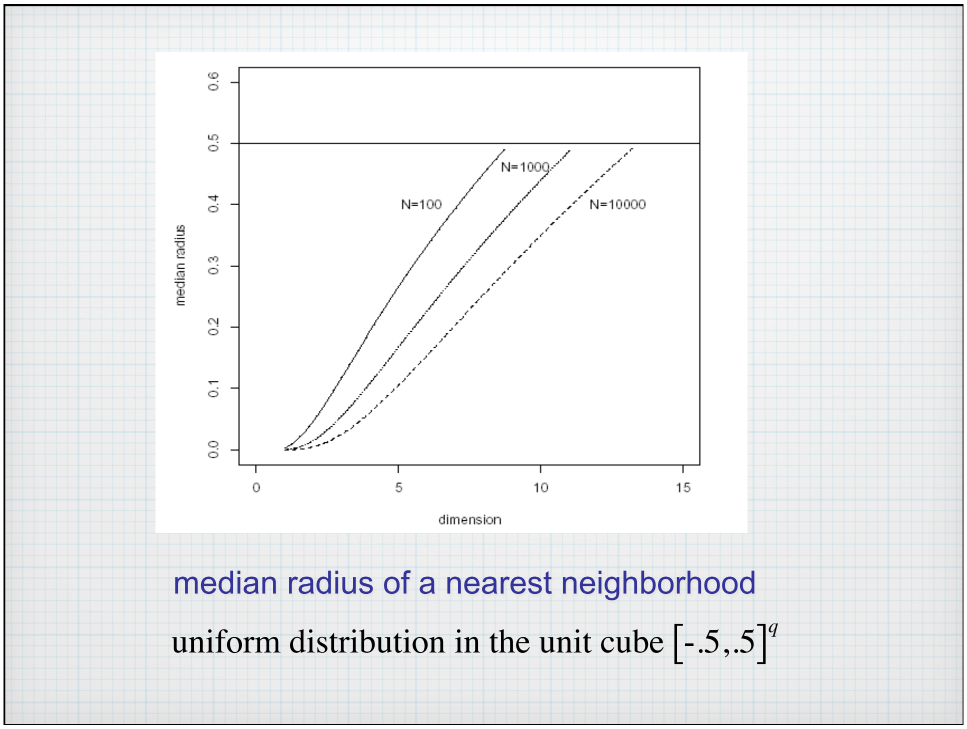
Curse of Dimensionality

Distance values from a given point

Values flatten out as dimensionality increases

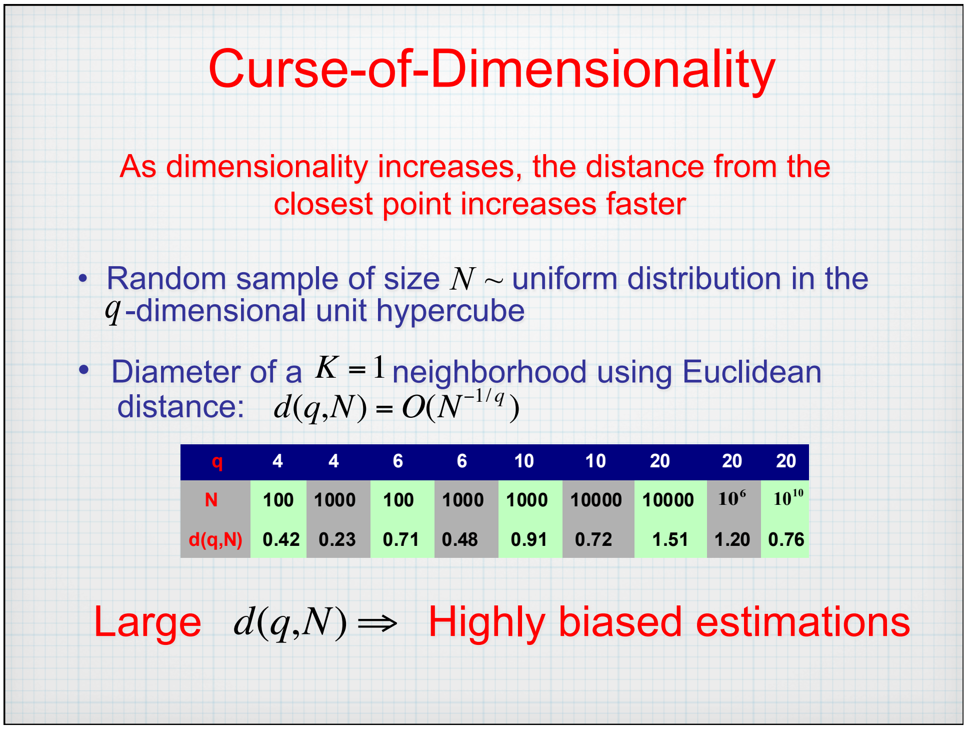


Computing radii of nearest neighborhoods



median radius of a nearest neighborhood

uniform distribution in the unit cube $\left[-.5,.5\right]^q$



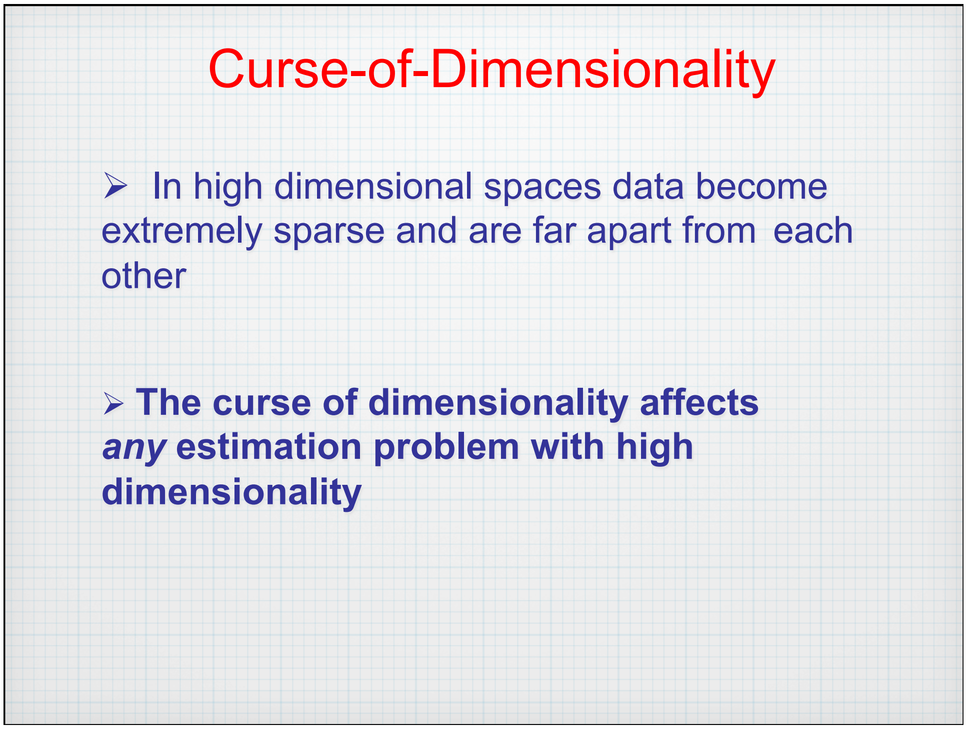
Curse-of-Dimensionality

As dimensionality increases, the distance from the closest point increases faster

- Random sample of size N ~ uniform distribution in the q-dimensional unit hypercube
- Diameter of a $K=1$ neighborhood using Euclidean distance: $d(q,N)=O(N^{-1/q})$

q	4	4	6	6	10	10	20	20	20
N	100	1000	100	1000	1000	10000	10000	10^6	10^{10}
d(q,N)	0.42	0.23	0.71	0.48	0.91	0.72	1.51	1.20	0.76

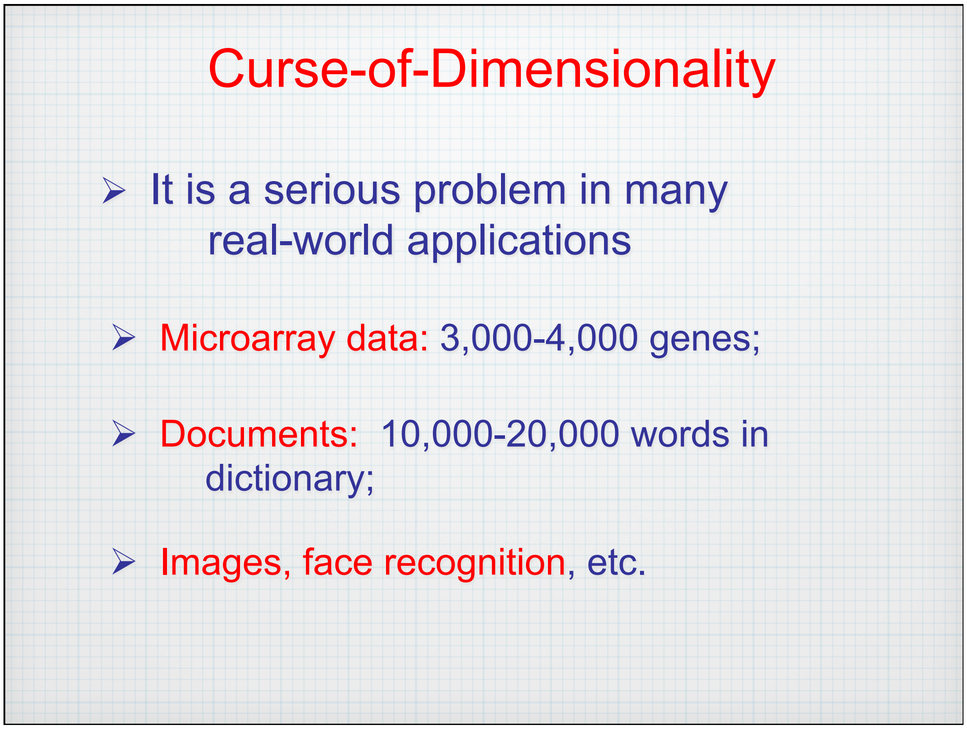
Large $d(q,N) \Rightarrow$ Highly biased estimations



Curse-of-Dimensionality

- In high dimensional spaces data become extremely sparse and are far apart from each other

- **The curse of dimensionality affects *any* estimation problem with high dimensionality**



Curse-of-Dimensionality

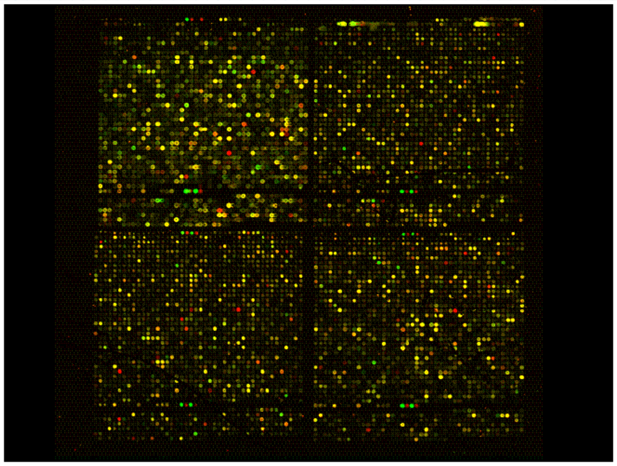
- It is a serious problem in many real-world applications

- Microarray data: 3,000-4,000 genes;

- Documents: 10,000-20,000 words in dictionary;

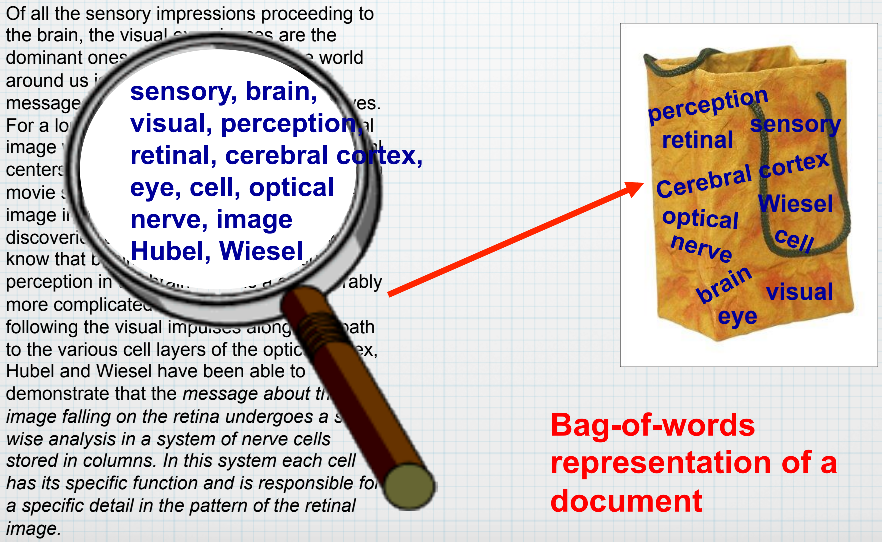
- Images, face recognition, etc.

Microarray Data Analysis



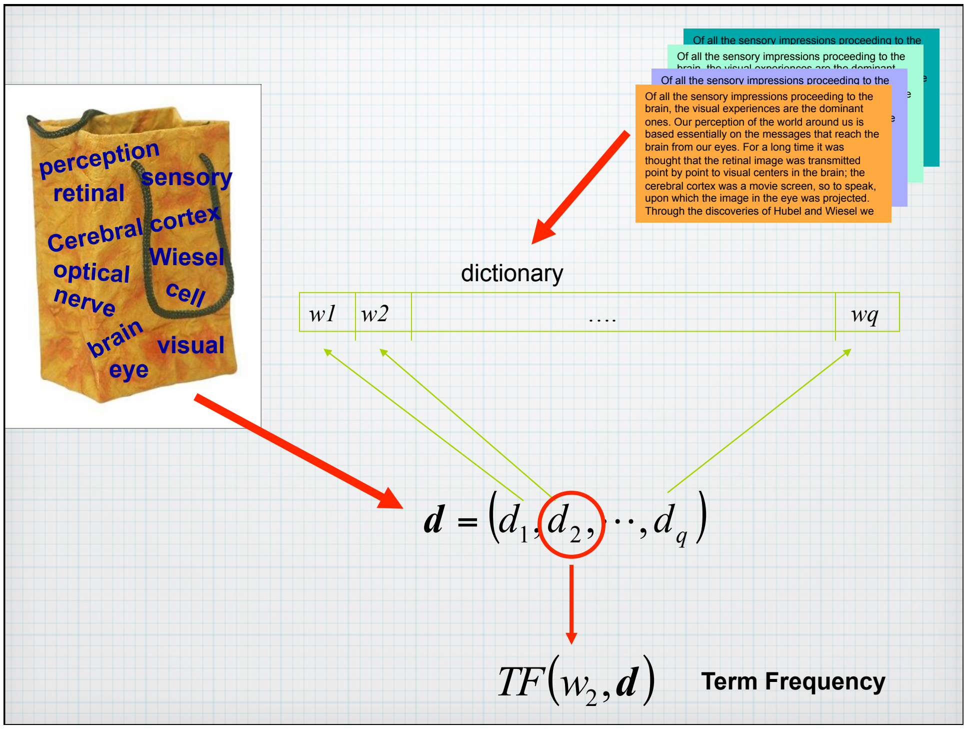
Problem: Which samples are most similar to each other, in terms of their expression profiles across genes

Document Classification



Of all the sensory impressions proceeding to
the brain, the visual ... are the
dominant ones ... world
around us i...
message ... es.
For a lo ... al
image ... al
centers
movie
image i
discoveri
know that b
perception in ... ably
more complicated
following the visual impulses along ... path
to the various cell layers of the optic ... ex,
Hubel and Wiesel have been able to
demonstrate that the *message about th*
image falling on the retina undergoes a s
wise analysis in a system of nerve cells
stored in columns. In this system each cell
has its specific function and is responsible fo
a specific detail in the pattern of the retinal
image.

Bag-of-words representation of a document

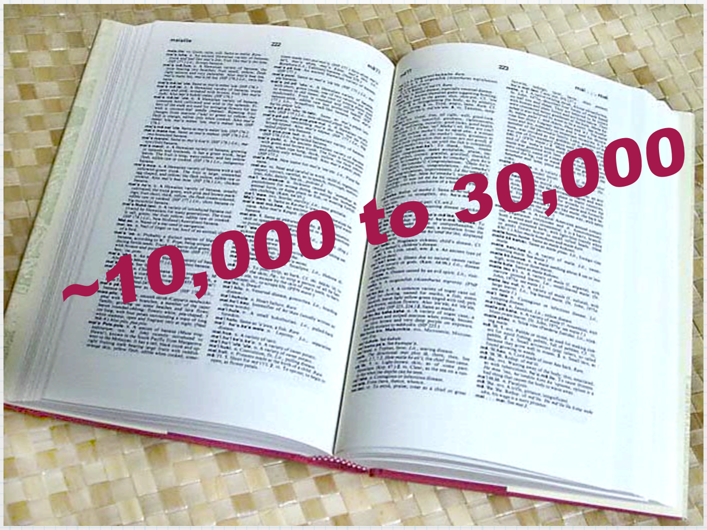


Of all the sensory impressions proceeding to the brain, the visual experiences are the dominant ones. Our perception of the world around us is based essentially on the messages that reach the brain from our eyes. For a long time it was thought that the retinal image was transmitted point by point to visual centers in the brain; the cerebral cortex was a movie screen, so to speak, upon which the image in the eye was projected. Through the discoveries of Hubel and Wiesel we

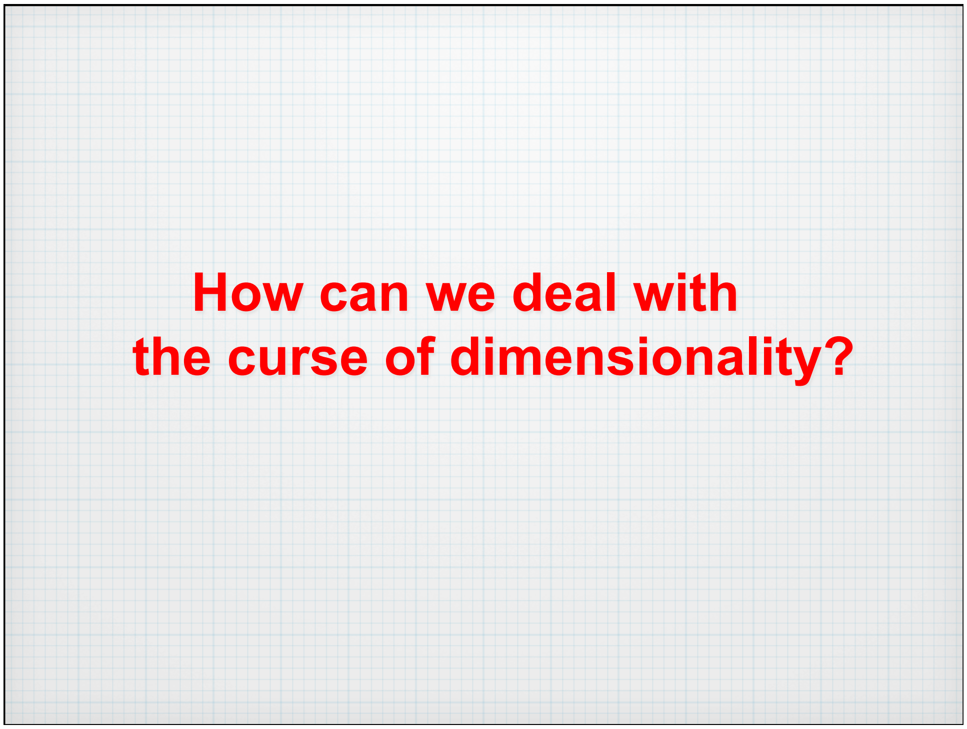
$$\mathbf{d} = \left(d_1, d_2, \cdots, d_q\right)$$

$TF\left(w_2, \mathbf{d}\right)$ **Term Frequency**

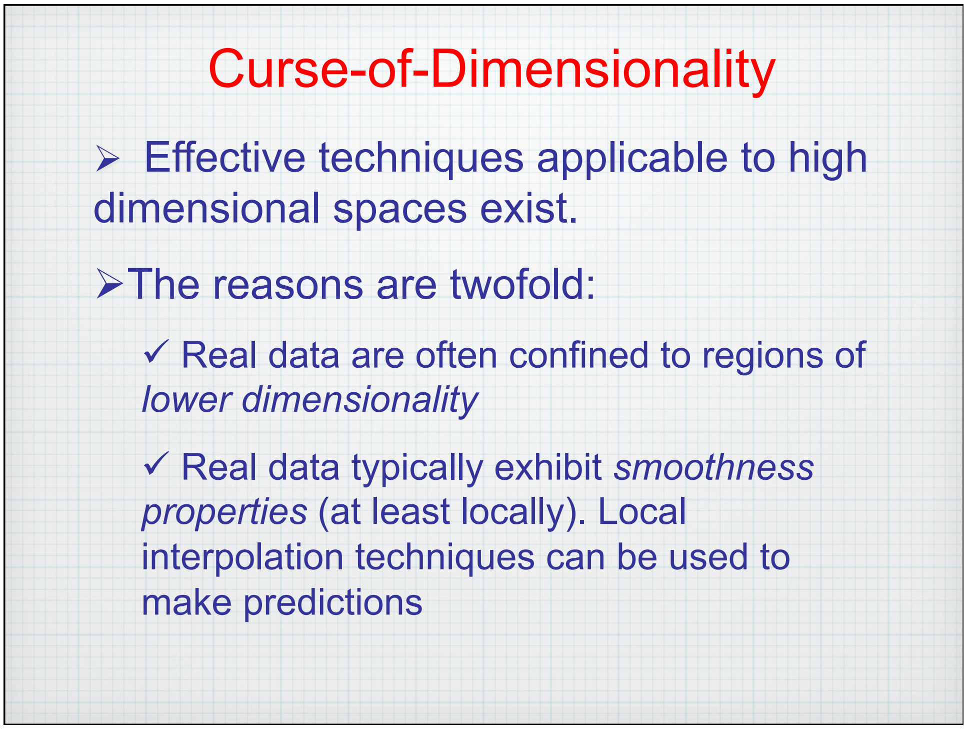
What's the size of the dictionary?



~10,000 to 30,000

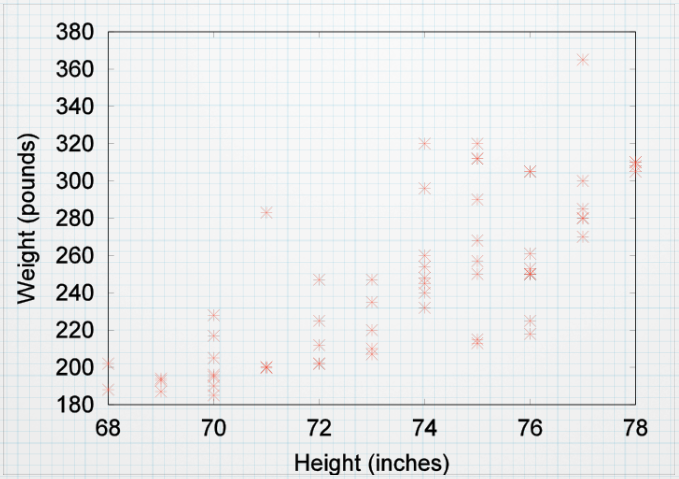


How can we deal with the curse of dimensionality?



Curse-of-Dimensionality

- Effective techniques applicable to high dimensional spaces exist.
- The reasons are twofold:
  - ✓ Real data are often confined to regions of *lower dimensionality*
  - ✓ Real data typically exhibit *smoothness properties* (at least locally). Local interpolation techniques can be used to make predictions



$$\begin{bmatrix} 7.68 & 92.2 \\ 92.2 & 1912.5 \end{bmatrix}$$

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \qquad \mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i$$

2×2 covariance matrix:

$$E\left[(\boldsymbol{x}-\mu)(\boldsymbol{x}-\mu)^T\right] =$$

$$E\left[\begin{pmatrix} x_1-\mu_1 \\ x_2-\mu_2 \end{pmatrix}(x_1-\mu_1, x_2-\mu_2)\right] =$$

$$E\left[\begin{matrix} (x_1-\mu_1)^2 & (x_1-\mu_1)(x_2-\mu_2) \\ (x_1-\mu_1)(x_2-\mu_2) & (x_2-\mu_2)^2 \end{matrix}\right] =$$

$$\frac{1}{N-1}\sum_{i=1}^{N}\left[\begin{matrix} (x_1^i-\mu_1)^2 & (x_1^i-\mu_1)(x_2^i-\mu_2) \\ (x_1^i-\mu_1)(x_2^i-\mu_2) & (x_2^i-\mu_2)^2 \end{matrix}\right]$$

$$\frac{1}{N-1}\sum_{i=1}^{N}\begin{bmatrix}\left(x_1^i-\mu_1\right)^2 & \left(x_1^i-\mu_1\right)\left(x_2^i-\mu_2\right)\\ \left(x_1^i-\mu_1\right)\left(x_2^i-\mu_2\right) & \left(x_2^i-\mu_2\right)^2\end{bmatrix}=$$

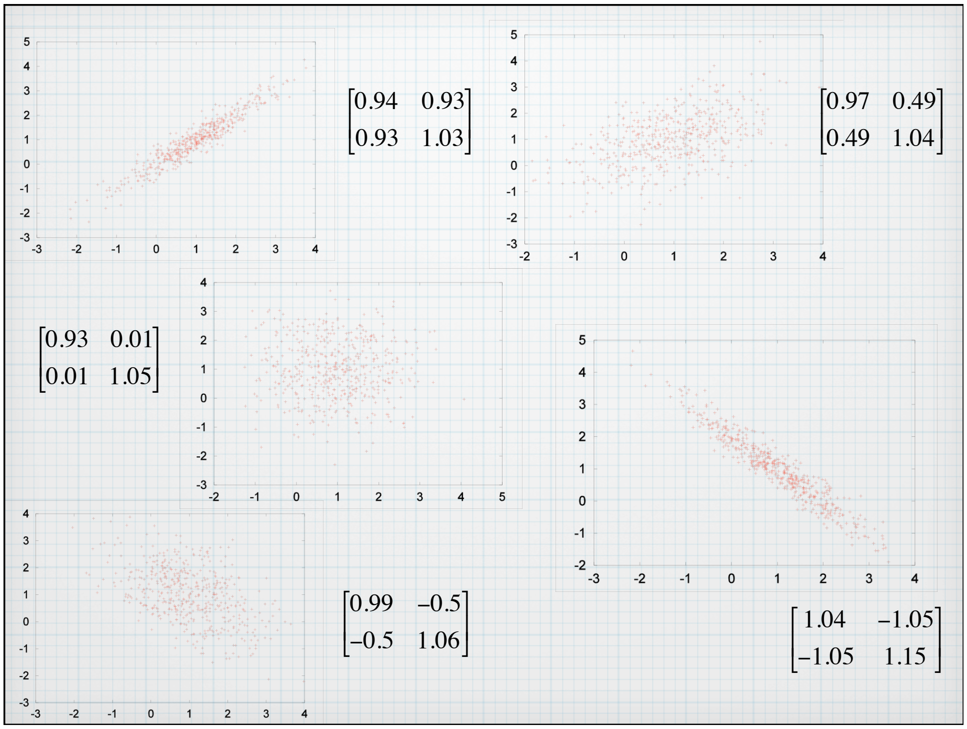
variance

covariance

$$\begin{bmatrix}\frac{1}{N-1}\sum_{i=1}^{N}\left(x_1^i-\mu_1\right)^2 & \frac{1}{N-1}\sum_{i=1}^{N}\left[\left(x_1^i-\mu_1\right)\left(x_2^i-\mu_2\right)\right]\\ \frac{1}{N-1}\sum_{i=1}^{N}\left[\left(x_1^i-\mu_1\right)\left(x_2^i-\mu_2\right)\right] & \frac{1}{N-1}\sum_{i=1}^{N}\left(x_2^i-\mu_2\right)^2\end{bmatrix}$$

covariance

variance



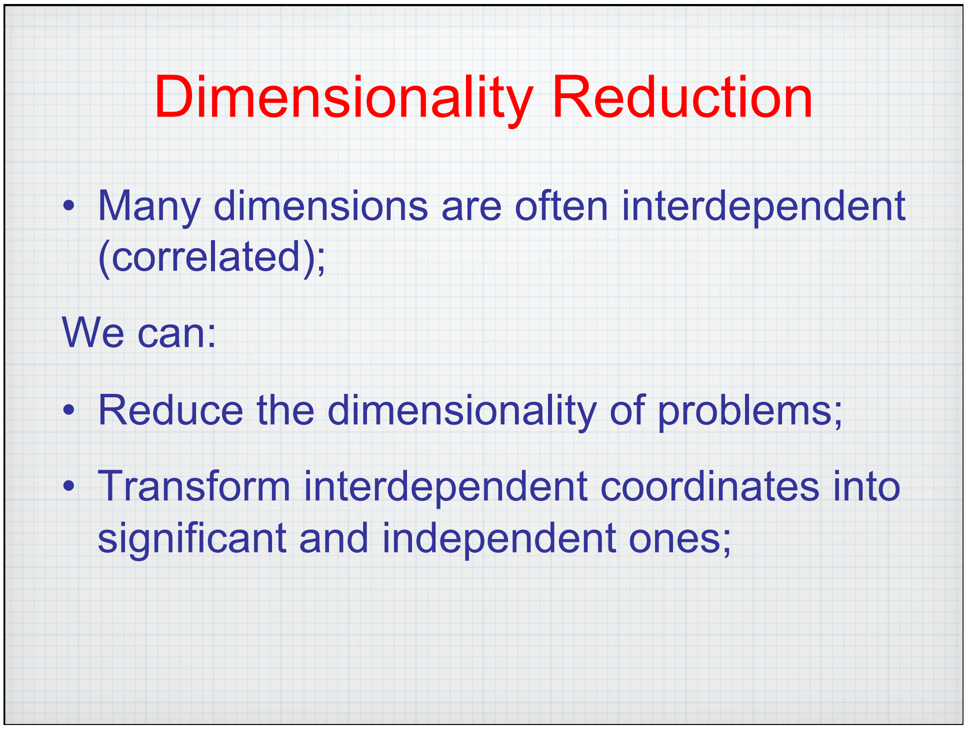
$$\begin{bmatrix}0.94 & 0.93\\ 0.93 & 1.03\end{bmatrix}$$

$$\begin{bmatrix}0.97 & 0.49\\ 0.49 & 1.04\end{bmatrix}$$

$$\begin{bmatrix}0.93 & 0.01\\ 0.01 & 1.05\end{bmatrix}$$

$$\begin{bmatrix}0.99 & -0.5\\ -0.5 & 1.06\end{bmatrix}$$

$$\begin{bmatrix}1.04 & -1.05\\ -1.05 & 1.15\end{bmatrix}$$

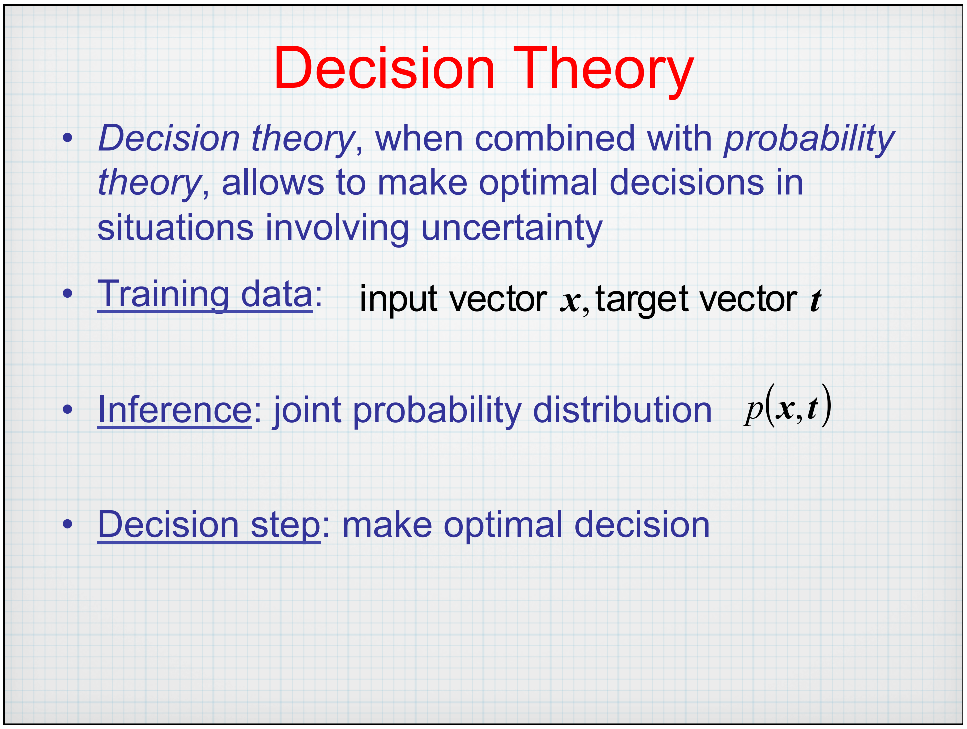


Dimensionality Reduction

- Many dimensions are often interdependent (correlated);

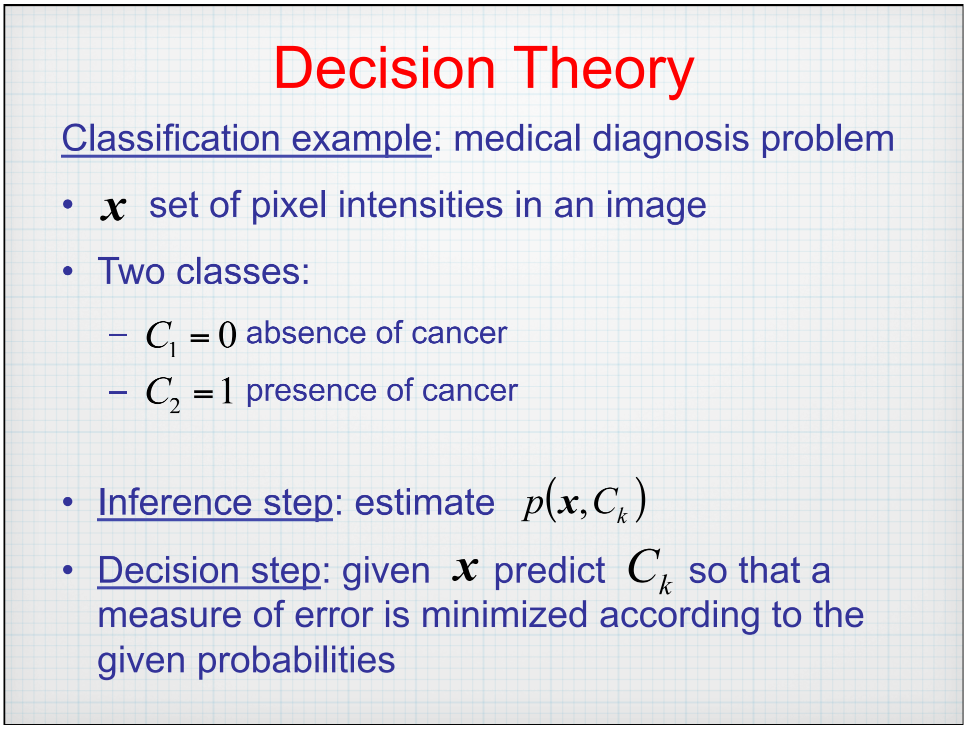
We can:

- Reduce the dimensionality of problems;
- Transform interdependent coordinates into significant and independent ones;



Decision Theory

- *Decision theory*, when combined with *probability theory*, allows to make optimal decisions in situations involving uncertainty
- Training data: input vector $\boldsymbol{x}$, target vector $\boldsymbol{t}$
- Inference: joint probability distribution $p(\boldsymbol{x},\boldsymbol{t})$
- Decision step: make optimal decision

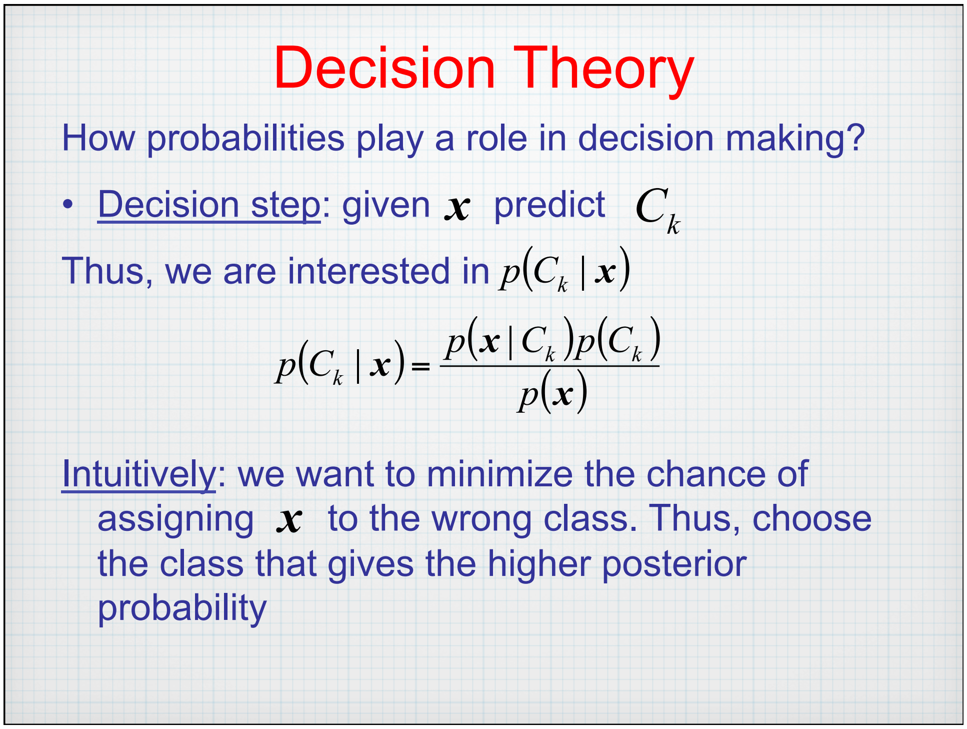


Decision Theory

Classification example: medical diagnosis problem

- $\boldsymbol{x}$ set of pixel intensities in an image
- Two classes:
  - $C_1 = 0$ absence of cancer
  - $C_2 = 1$ presence of cancer

- Inference step: estimate $p(\boldsymbol{x}, C_k)$
- Decision step: given $\boldsymbol{x}$ predict C_k so that a measure of error is minimized according to the given probabilities



Decision Theory

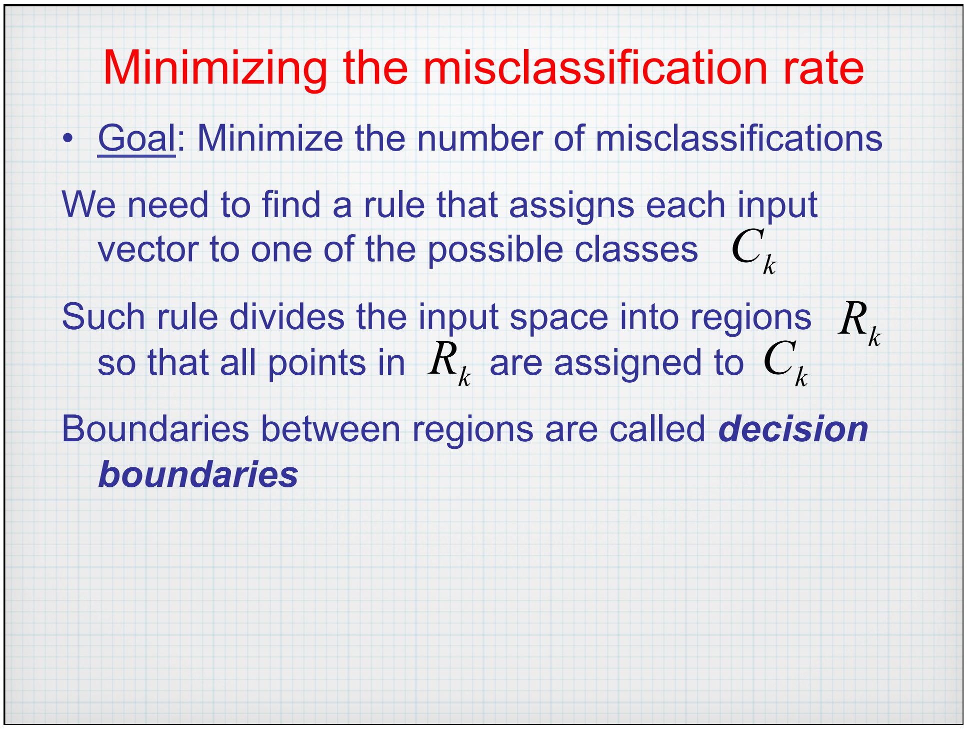
How probabilities play a role in decision making?

- Decision step: given $\boldsymbol{x}$ predict C_k

Thus, we are interested in $p(C_k \mid \boldsymbol{x})$

$$p(C_k \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid C_k) p(C_k)}{p(\boldsymbol{x})}$$

Intuitively: we want to minimize the chance of assigning $\boldsymbol{x}$ to the wrong class. Thus, choose the class that gives the higher posterior probability



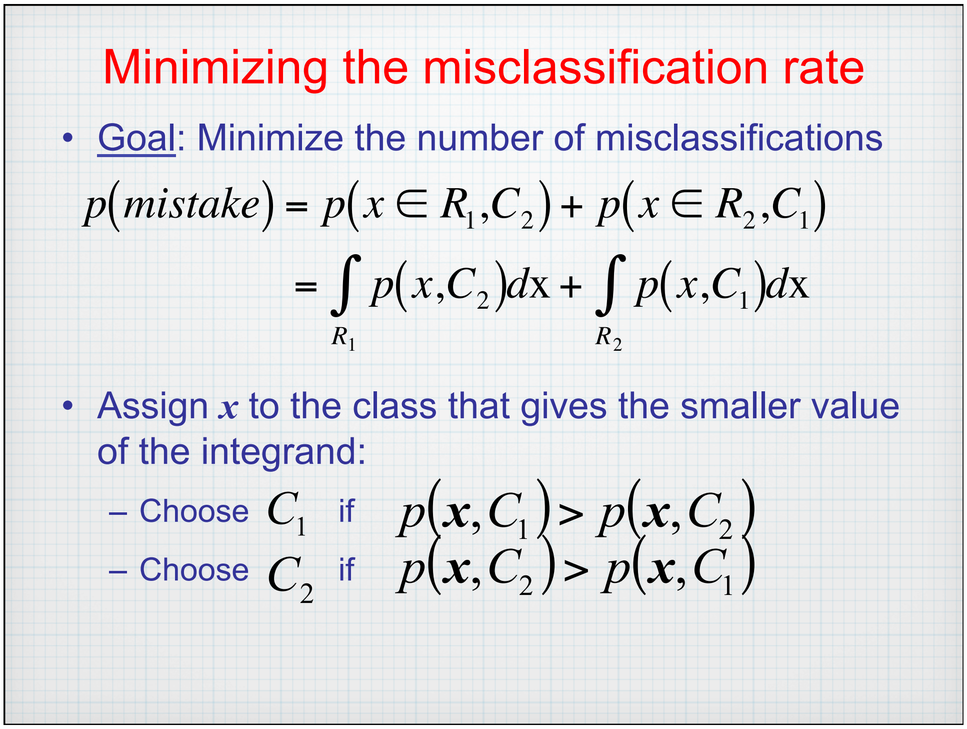
Minimizing the misclassification rate

- Goal: Minimize the number of misclassifications

We need to find a rule that assigns each input vector to one of the possible classes C_k

Such rule divides the input space into regions R_k so that all points in R_k are assigned to C_k

Boundaries between regions are called ***decision boundaries***



Minimizing the misclassification rate

- Goal: Minimize the number of misclassifications

$$p(mistake) = p(x \in R_1, C_2) + p(x \in R_2, C_1)$$
$$= \int_{R_1} p(x, C_2)d\mathrm{x} + \int_{R_2} p(x, C_1)d\mathrm{x}$$

- Assign x to the class that gives the smaller value of the integrand:
  - Choose C_1 if $p(\mathbf{x}, C_1) > p(\mathbf{x}, C_2)$
  - Choose C_2 if $p(\mathbf{x}, C_2) > p(\mathbf{x}, C_1)$

Minimizing the misclassification rate

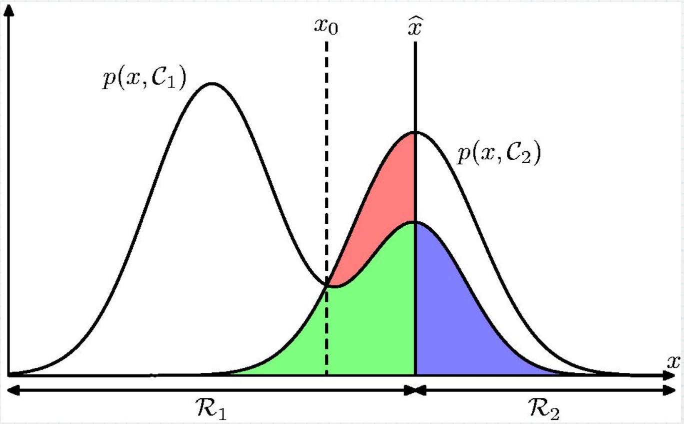
- Choose C_1 if $p(\boldsymbol{x}, C_1) > p(\boldsymbol{x}, C_2)$
- Choose C_2 if $p(\boldsymbol{x}, C_2) > p(\boldsymbol{x}, C_1)$

$$p(\boldsymbol{x}, C_k) = p(C_k \mid \boldsymbol{x})p(\boldsymbol{x})$$

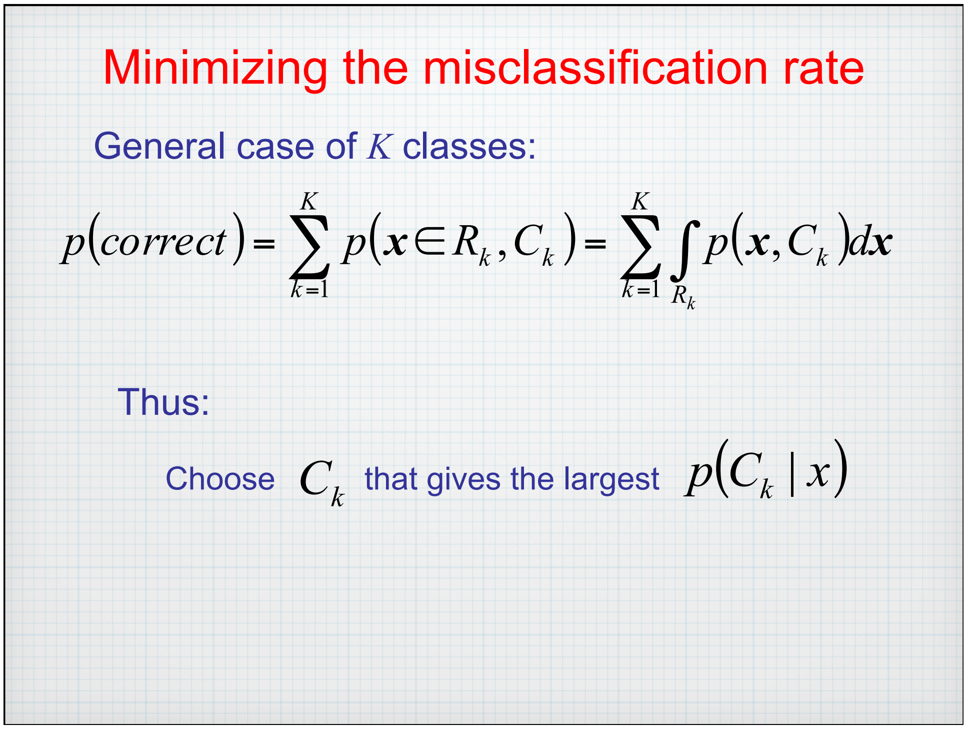
Thus:

- Choose C_1 if $p(C_1 \mid \boldsymbol{x}) > p(C_2 \mid \boldsymbol{x})$
- Choose C_2 if $p(C_2 \mid \boldsymbol{x}) > p(C_1 \mid \boldsymbol{x})$

Minimizing the misclassification rate



Optimal decision boundary: $\hat{x} = x_0$



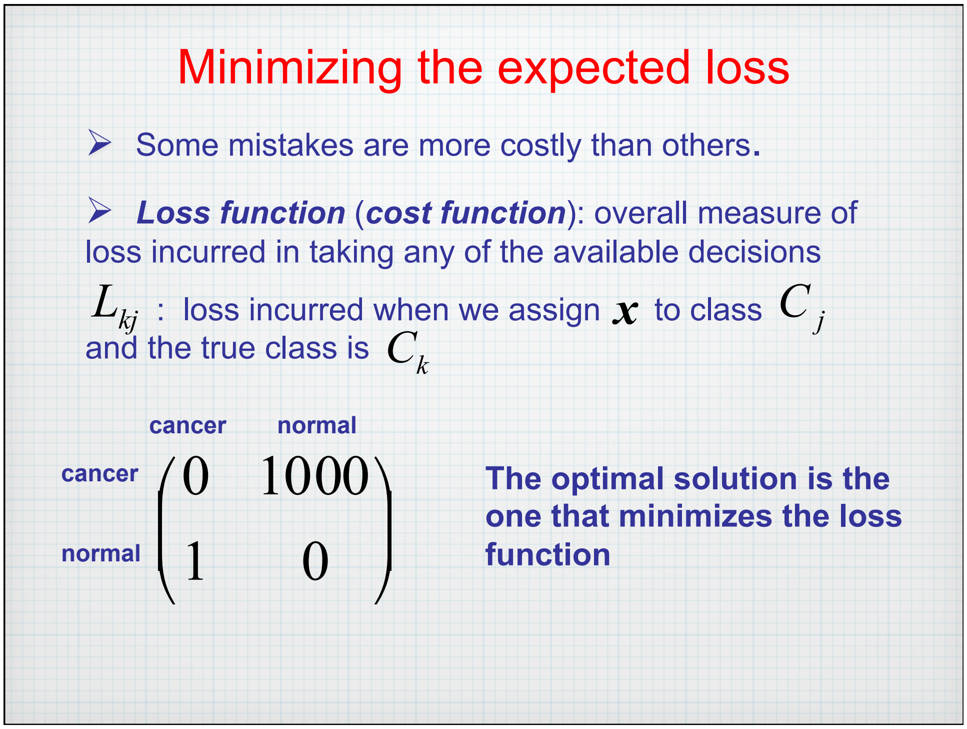
Minimizing the misclassification rate

General case of K classes:

$$p(correct) = \sum_{k=1}^{K} p(\boldsymbol{x} \in R_k, C_k) = \sum_{k=1}^{K} \int_{R_k} p(\boldsymbol{x}, C_k) d\boldsymbol{x}$$

Thus:

Choose C_k that gives the largest $p(C_k \mid x)$



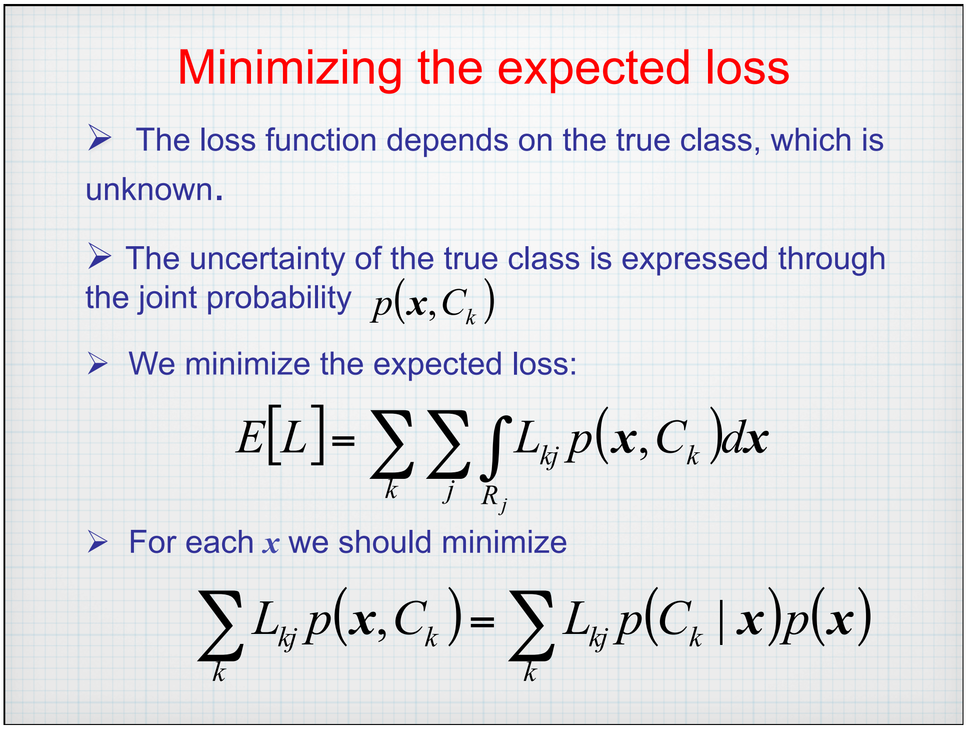
Minimizing the expected loss

- Some mistakes are more costly than others.
- ***Loss function*** (***cost function***): overall measure of loss incurred in taking any of the available decisions

L_{kj} : loss incurred when we assign $\boldsymbol{x}$ to class C_j and the true class is C_k

$$\begin{array}{ccc} & \text{cancer} & \text{normal} \\ \text{cancer} & 0 & 1000 \\ \text{normal} & 1 & 0 \end{array}$$

The optimal solution is the one that minimizes the loss function



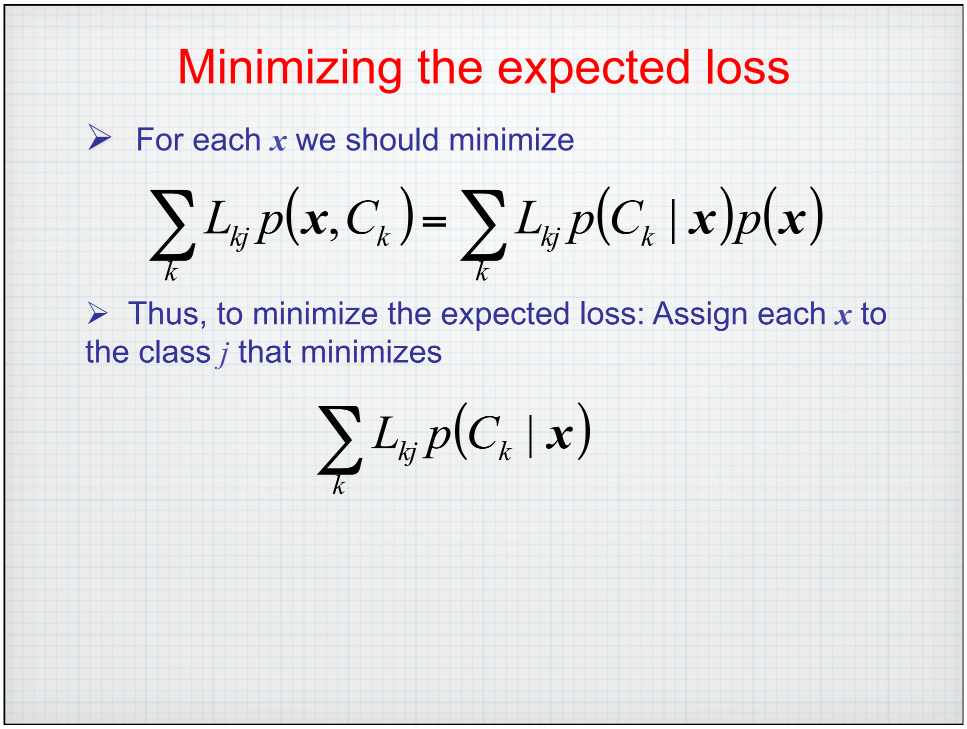
Minimizing the expected loss

- The loss function depends on the true class, which is unknown.
- The uncertainty of the true class is expressed through the joint probability $p(\boldsymbol{x}, C_k)$
- We minimize the expected loss:

$$E[L] = \sum_k \sum_j \int_{R_j} L_{kj}\, p(\boldsymbol{x}, C_k)\, d\boldsymbol{x}$$

- For each x we should minimize

$$\sum_k L_{kj}\, p(\boldsymbol{x}, C_k) = \sum_k L_{kj}\, p(C_k \mid \boldsymbol{x})\, p(\boldsymbol{x})$$



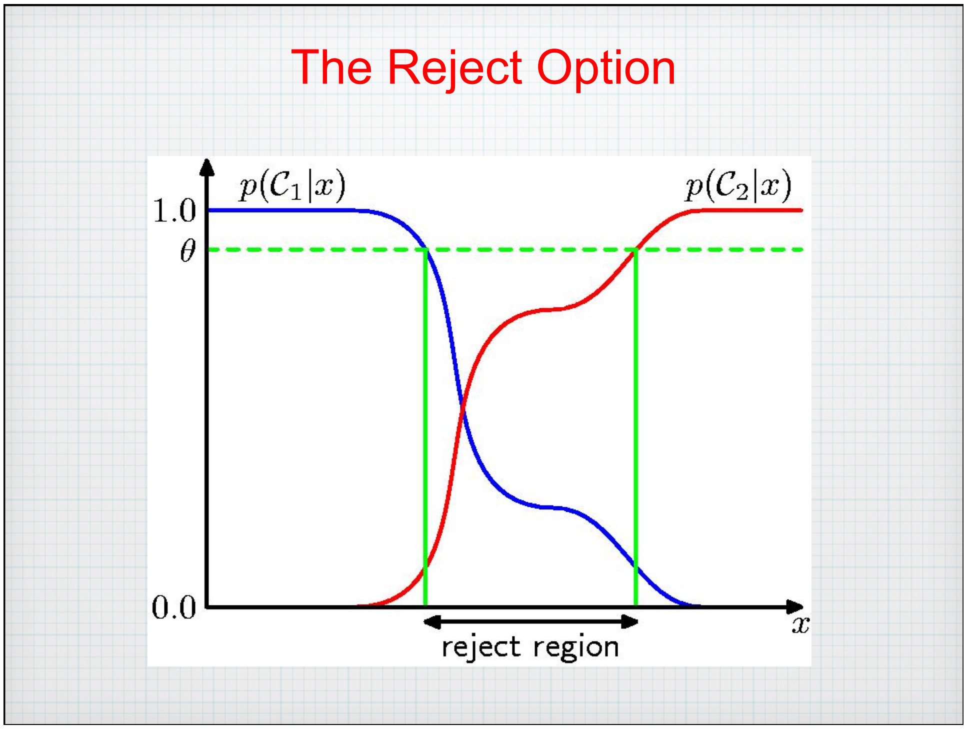
Minimizing the expected loss

- For each x we should minimize

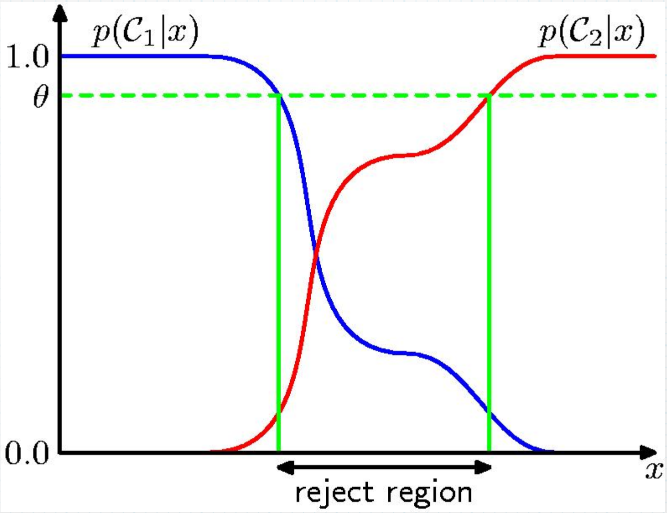
$$\sum_k L_{kj}\, p(\boldsymbol{x}, C_k) = \sum_k L_{kj}\, p(C_k \mid \boldsymbol{x})\, p(\boldsymbol{x})$$

- Thus, to minimize the expected loss: Assign each x to the class j that minimizes

$$\sum_k L_{kj}\, p(C_k \mid \boldsymbol{x})$$



The Reject Option


$p(\mathcal{C}_1|x)$
$p(\mathcal{C}_2|x)$
1.0
θ
0.0
x
reject region