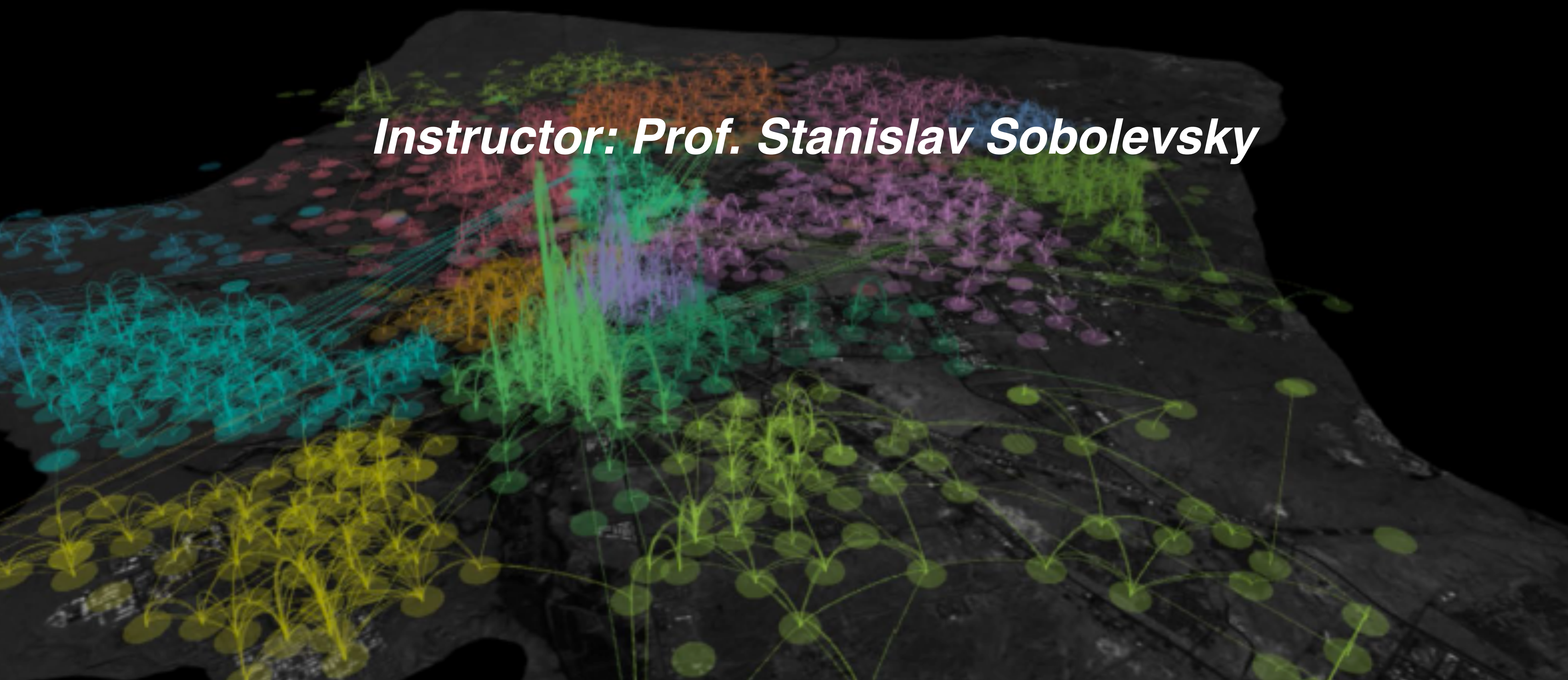
CENTER FOR URBAN SCIENCE+PROGRESS

APPLIED DATA SCIENCE

6004.002, Fall 2019

Principle component analysis. Classification

Instructor: Prof. Stanislav Sobolevsky

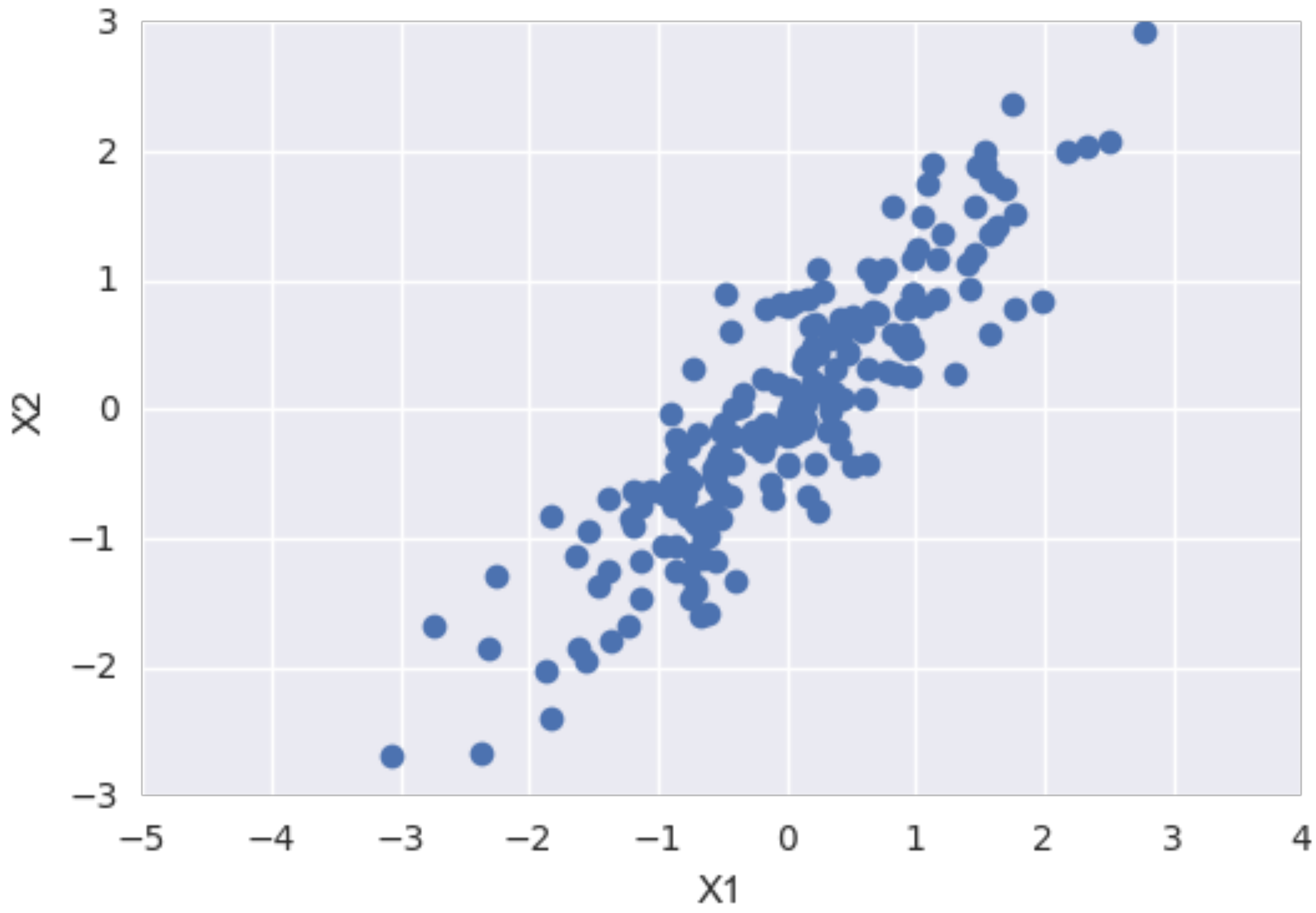


Principal component analysis

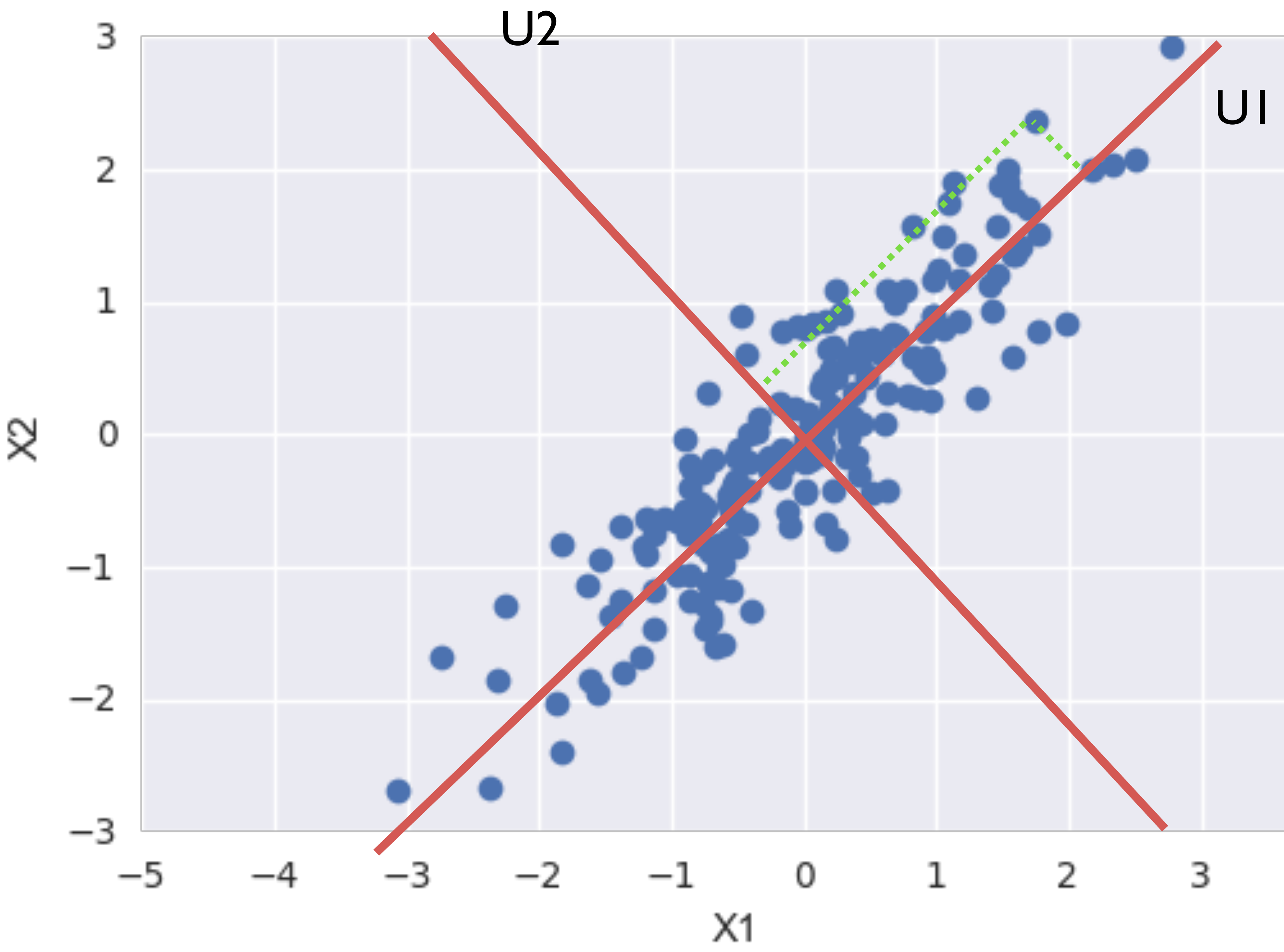
correlated features to uncorrelated

$$(x_1, x_2, x_3, ..., x_n) \rightarrow (u_1, u_2, u_3, ..., u_n)$$

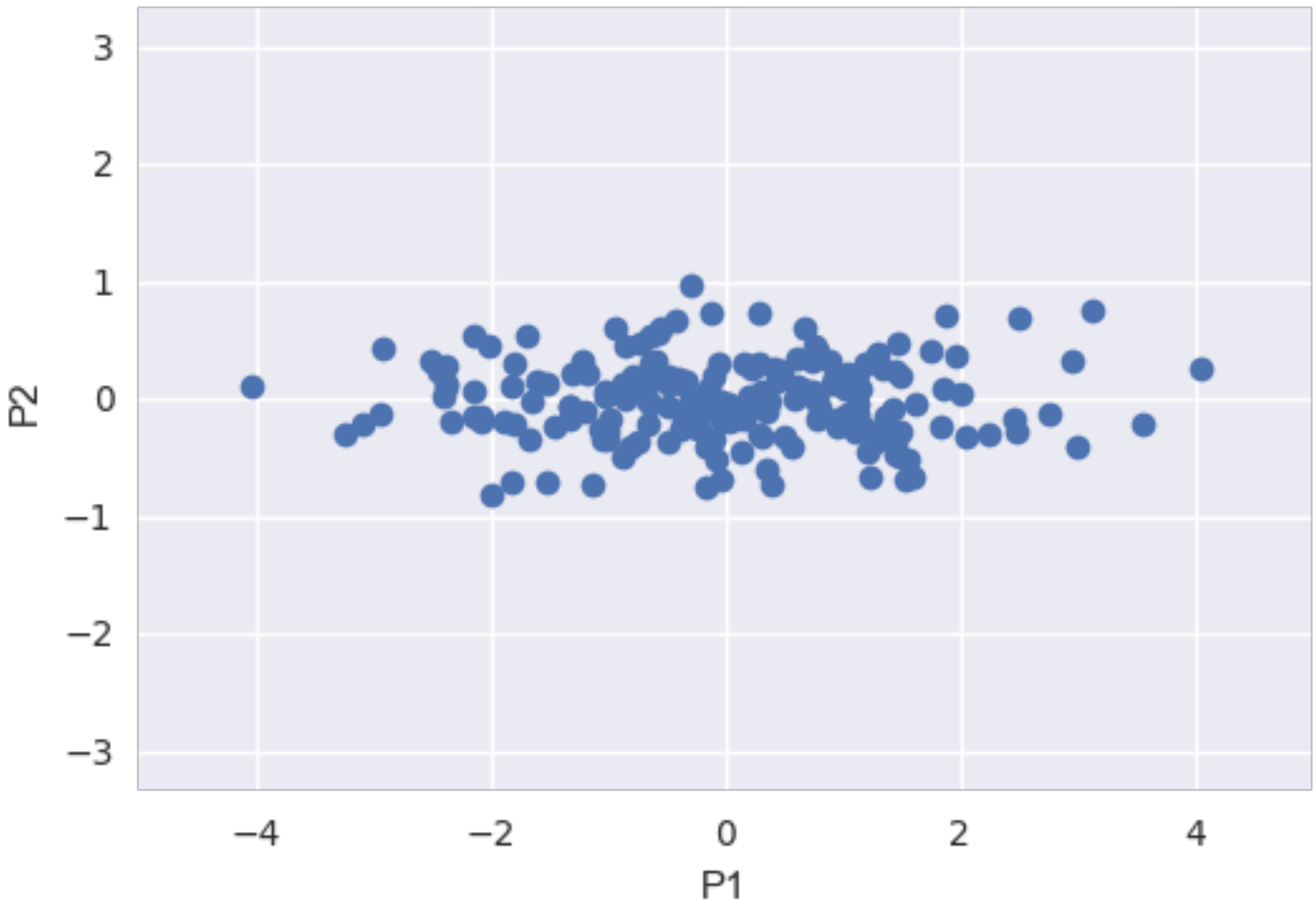
Original data



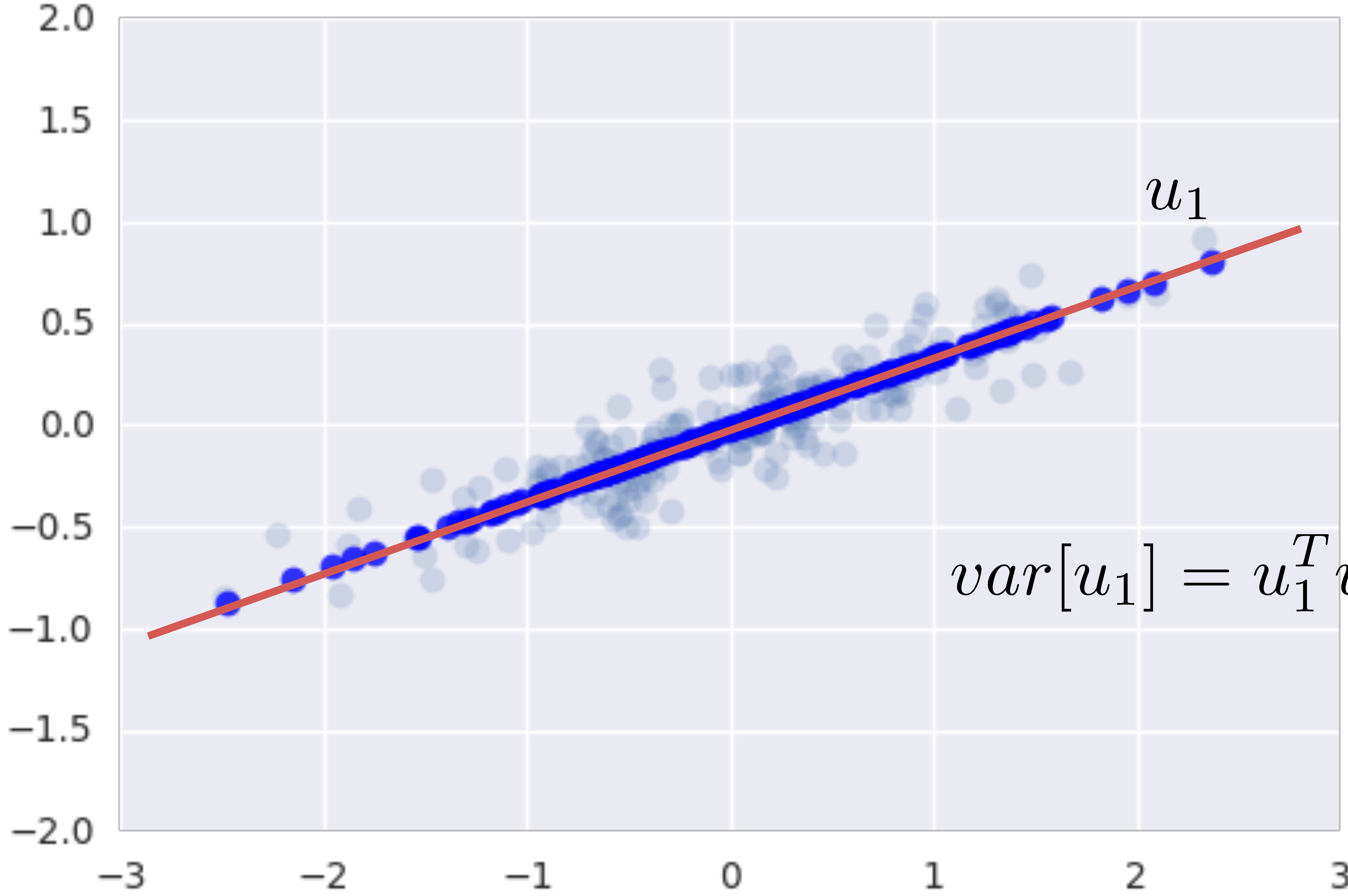
Original data - new system of coordinates



Uncorrelated data (rotation)



Almost the same information



$$var[u_1] = u_1^T u_1 \rightarrow max$$

Principal components - maths

Given the standardized data $X = \{x_i^j, i = 1..n, j = 1..N\}$

Find uncorrelated latent factors U (or P)

$$u_j = x_1 v_j^1 + x_2 v_j^2 + ... + x_n v_j^n$$

Matrix for for the rotation transform

$$u_i = X v_i \qquad U = XV$$

V - n x p
U - N x p

Look for linear combinations of factors one-by-one

Principal components - optimization objective

Looking for $u_j = x_1 v_j^1 + x_2 v_j^2 + \ldots + x_n v_j^n$

Start with $u_1 = X v_1$

Such that $var[u_1] = u_1^T u_1 \to max$

$$var[u_1] = u_1^T u_1 = (X v_1)^T (X v_1) = v_1^T X^T X v_1$$

Find $v_1 = argmax_{v_1 : v_1^T v_1 = 1} v_1^T X^T X v_1$

$$v_i = argmax_{v_i : v_i^T v_i = 1, v_i^T v_j = 0, j<i} v_i^T X^T X v_i$$

Principal components - answer

So which vectors v maximize the quantity below?

$$v_i = argmax_{v_i : v_i^T v_i = 1, v_i^T v_j = 0, j<i} v_i^T X^T X v_i$$

Eigenvectors $\lambda_i v_i = X^T X v_i$ $\quad v_i^T v_i = 1 \quad v_i^T v_j = 0$

$$Var[u_i] = v_i^T X^T X v_i = \lambda_i v_i^T v_i = \lambda_i \quad \lambda_1 > \lambda_2 > \ldots > \lambda_n > 0$$

Projection for the leading PC v_1

Is the leading eigenvector with the max eigenvalue

Recall the concept of eigenvectors/eigenvalues

$$\lambda v = Av \qquad \lambda - eigenvalue, v - eigenvector$$

$$(\lambda I - A)v = 0 \qquad det(\lambda I - A) = 0$$

Find

$\lambda_1, \lambda_2, ...\lambda_n$

$v_1, v_2, ...v_n$

Define up to a scaling factor, can require unit length

$$v_i \rightarrow Cv_i \quad |v_i| = 1$$

When $\lambda_i \neq \lambda_j \Rightarrow v_i^T v_j = 0$

$A^T = A$

Proof $v_i^T A v_j = \lambda_j v_i^T v_j$

$$v_i^T A v_j = (Av_i)^T v_j = \lambda_i v_i^T v_j$$

Principal components - proof

$$v_i = argmax_{v_i : v_i^T v_i = 1, v_i^T v_j = 0, j<i} v_i^T X^T X v_i$$

Consider eigenvectors:

$$\lambda_i v_i = X^T X v_i \qquad v_i^T v_i = 1 \qquad \lambda_1 > \lambda_2 > \ldots > \lambda_n > 0$$

$$v_i^T X^T X v_i = \lambda_i v_i^T v_i = \lambda_i$$

$$w = e_1 v_1 + e_2 v_2 + \ldots + e_n v_n \qquad w^T w = e_1^2 + e_2^2 + \ldots + e_n^2 = 1$$

$$w^T X^T X w = \lambda_1 e_1^2 + \lambda_2 e_2^2 + \ldots + \lambda_n e_n^2 \to \max$$

$$w = v_1, e_1 = 1, e_2 = e_3 = \ldots = e_n = 0$$

Principal components - select by variation

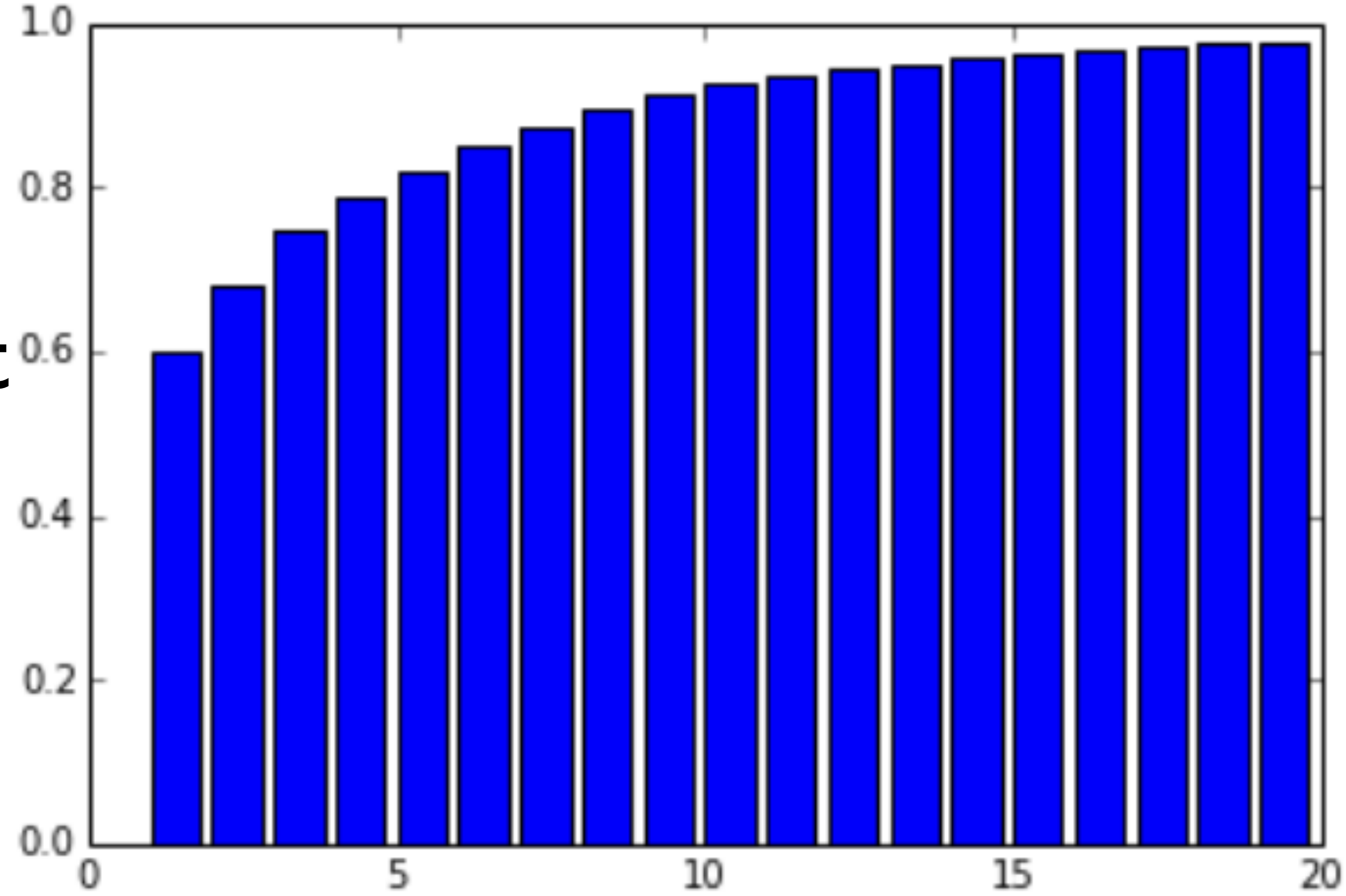
Information explained by a PC u_i

$$Var[u_i] = \lambda_i$$

% of total $\lambda_i / \sum_j \lambda_j$

Take first k, such that

$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{n} \lambda_i} \geq \alpha$$



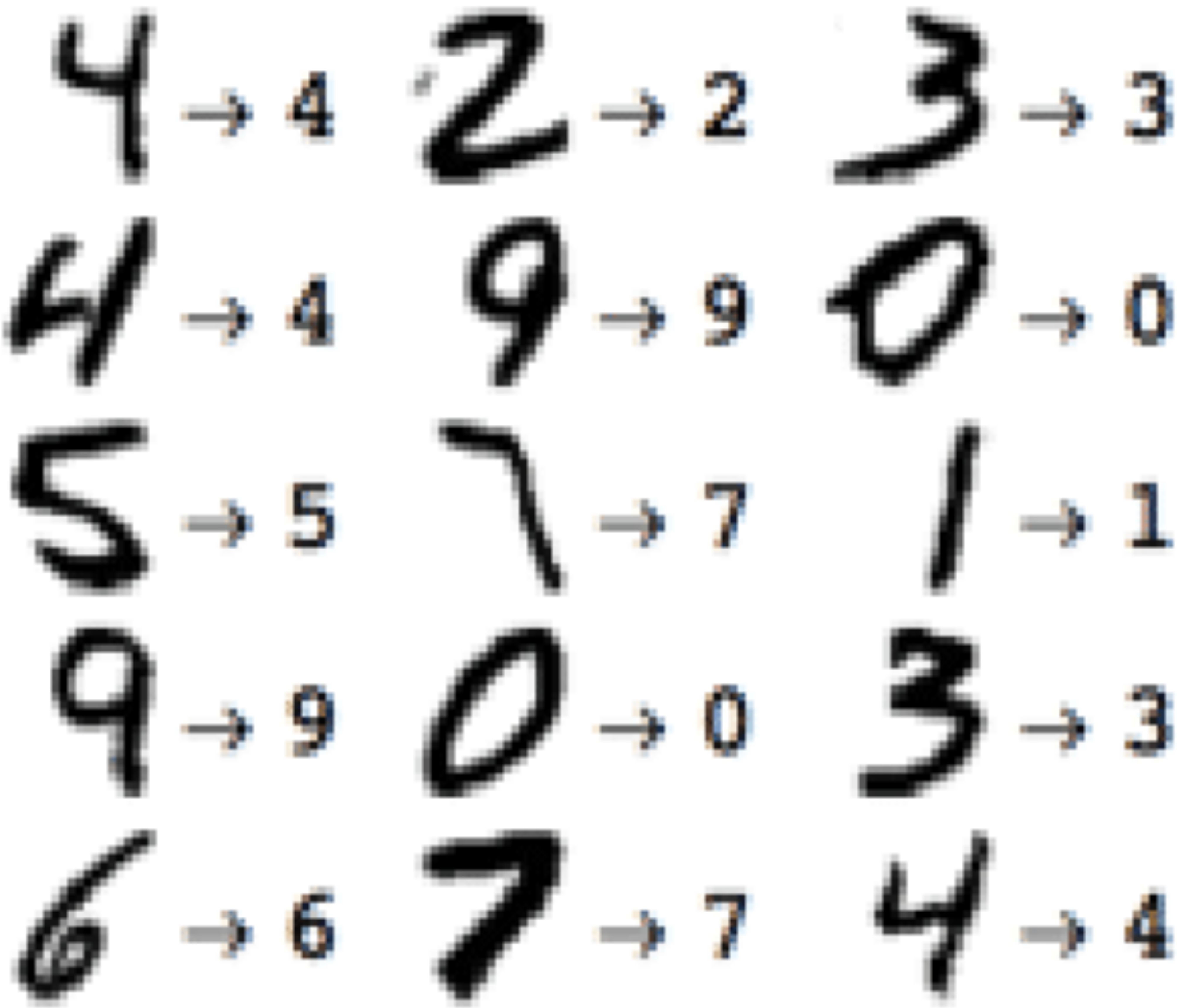
Information explained by leading PC's

Supervised learning: classification

Data/input	Discrete labels	Dependence
x_1	y_1	$y = f(x)$
		$x^* \rightarrow y^*$
x_2	y_2	y-discrete
...	...	$P(y\|x^*)$
x_N	y_N	

$$X = \{x_i, i = 1..N\} = \{x_i^j, i = 1..N, j = 1..n\}$$

$$Y = \{y_i, i = 1..N\}$$



wolfram.com

Binary/multiple classification

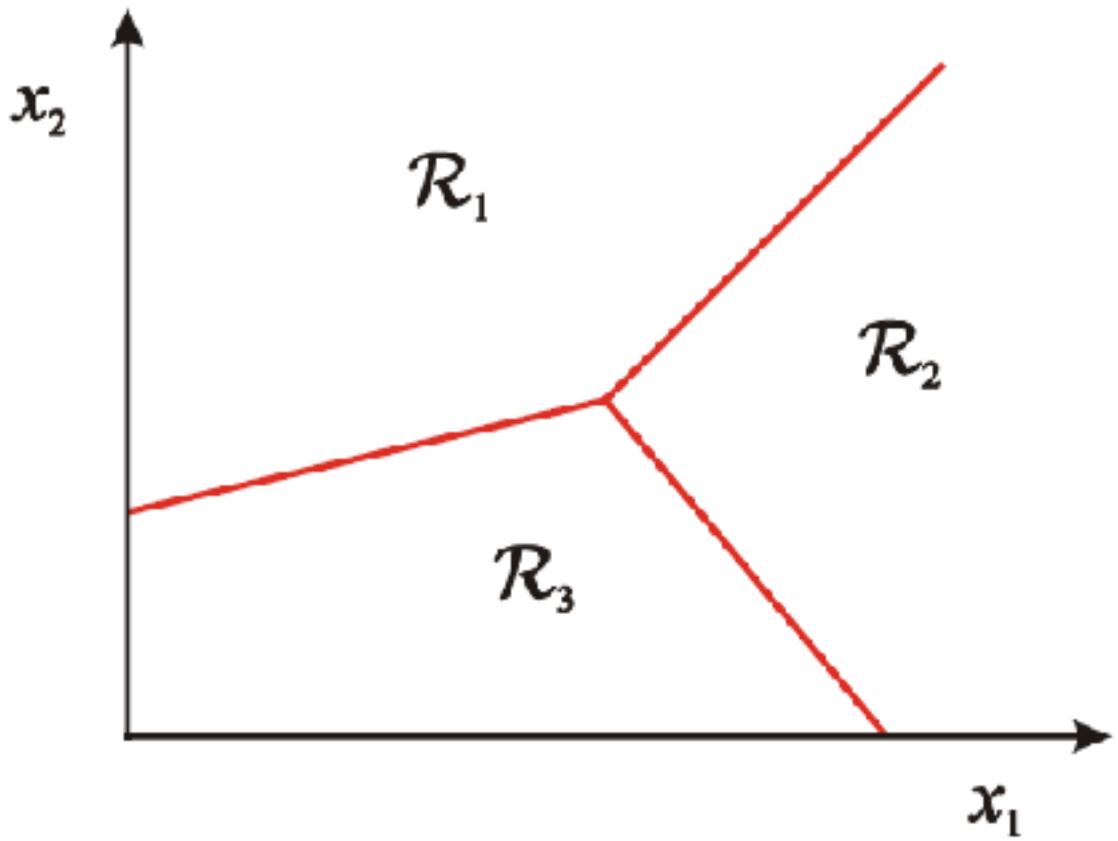
$$P(y|x^*) = Bern(y|\mu(x^*))$$

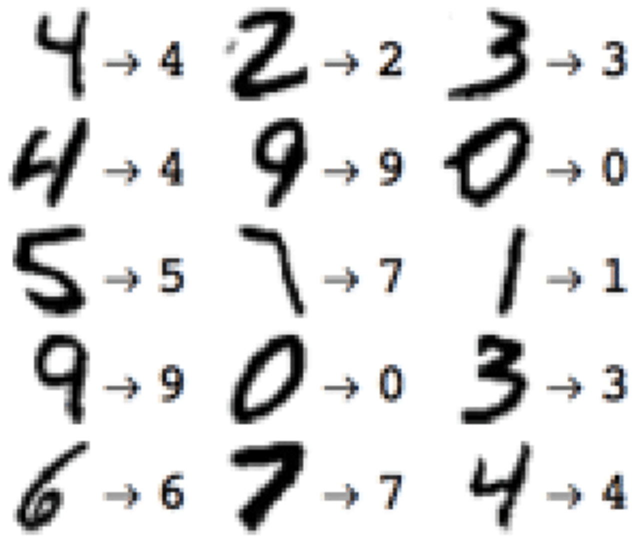
$$y = \begin{cases} 1 & \text{event happened} \\ 0 & \text{event not happened} \end{cases}$$

Multiple classification y=k

Multiple classification to binary:
c=k
c~=k

Multi-class classification

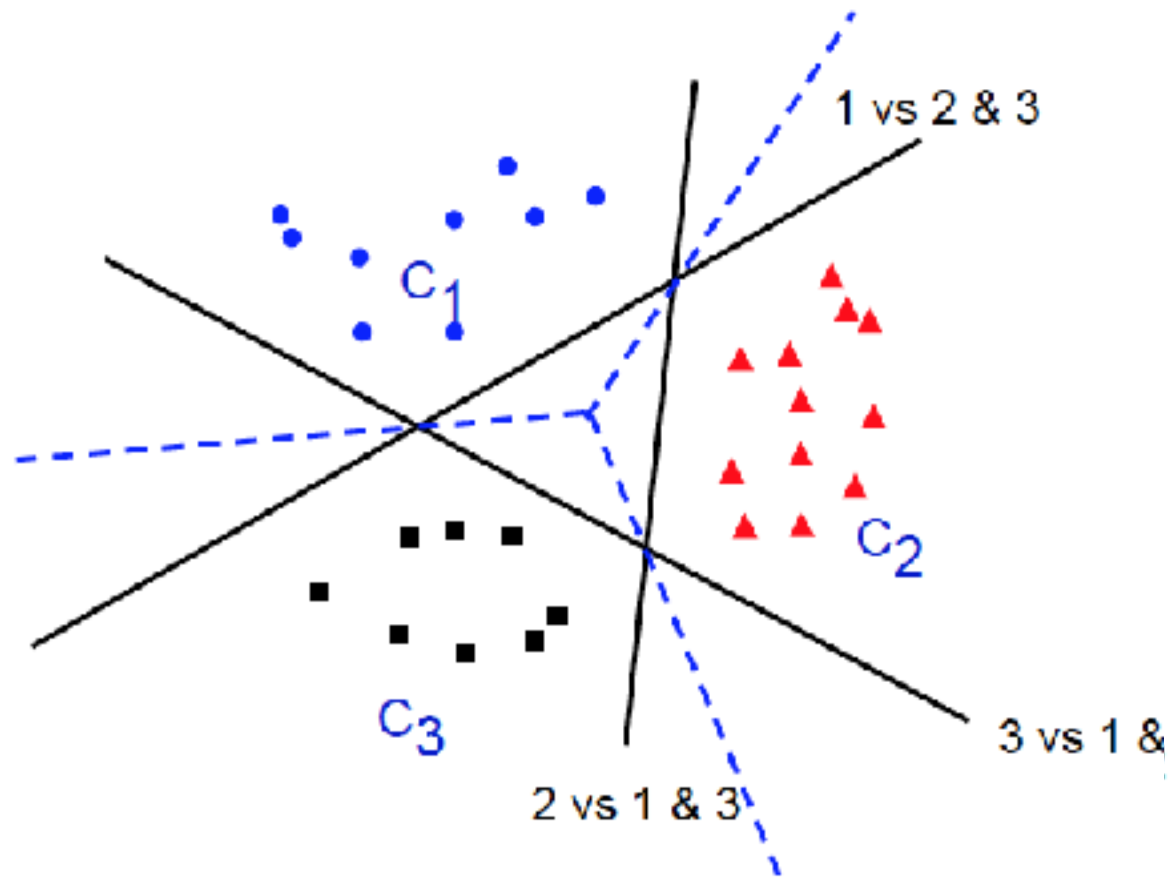




Is R1? Rather than R2 or R3

Is R2? Rather than R1 or R3

Is R3? Rather than R1 or R2

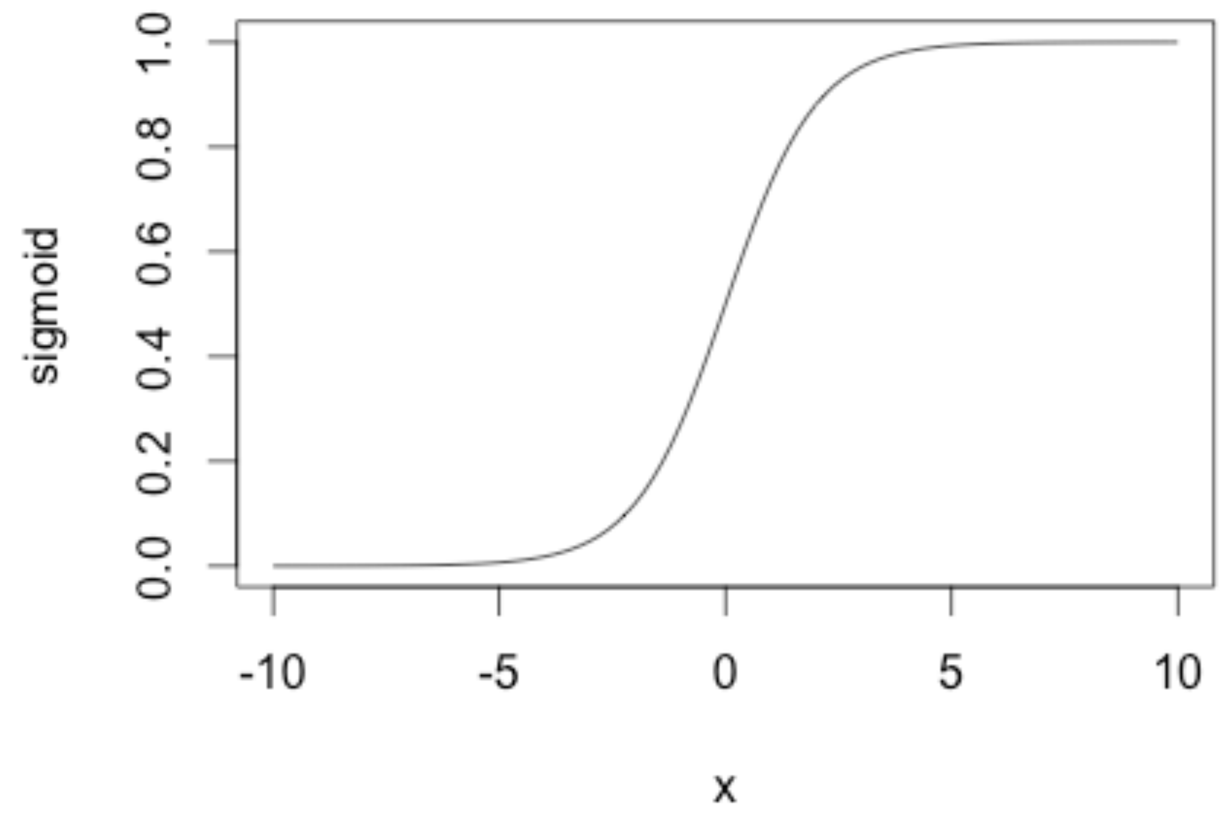


Logistic regression

$$P(y|x,\beta) = Bern(y|\mu(x,\beta))$$

$$\mu(x,\beta) = f(x\beta)$$

$$f(x) = \sigma(x) = \frac{e^x}{1+e^x} = \frac{1}{1+e^{-x}}$$



Logistic regression

$$P(y|x,\beta) = Bern(y|\mu(x,\beta))$$

$$f(x) = \sigma(x) = \frac{e^x}{1+e^x} = \frac{1}{1+e^{-x}}$$

$$\mu(x,\beta) = f(x\beta)$$

$$P(y=1) = \sigma(x\beta) = \frac{\exp(x\beta)}{1+\exp(x\beta)} = \frac{1}{1+\exp(-x\beta)}$$

$$P(y=0) = 1 - P(y=1) = \frac{1}{1+\exp(x\beta)}$$

Choosing a classifier to best fit the data

$$P(y|x^*)$$

Data/input	Discrete labels	Probability
x_1	y_1	P1
x_2	y_2	P2
$\dots$	$\dots$	
x_N	y_N	PN

Choose the model based on P1*P2*…*PN

Logistic regression - log-likelihood

$$P(y=1)=\sigma(x\beta)=\frac{\exp(x\beta)}{1+\exp(x\beta)}=\frac{1}{1+\exp(-x\beta)}$$

$$L=\prod_i P(y=y_i|x_i,\beta)$$

$$P(y=0)=1-P(y=1)=\frac{1}{1+\exp(x\beta)}$$

$$log(L)=\sum_i log\left(P(y=y_i|x_i,\beta)\right)$$

$$=\sum_i y_i log\left(P(y=1|x_i,\beta)\right)+\sum_i(1-y_i)log\left(P(y=0|x_i,\beta)\right)$$

$$=-\sum_i log\left(1+exp((2y_i-1)x_i\beta)\right)$$

$$\beta=argmin_\beta\sum_i log\left(1+exp((2y_i-1)x_i\beta)\right)$$

Accuracy

$$acc = \frac{|\{i : y_i^{est} = y_i^{true}\}|}{|\{i\}|}$$

Not all errors are the same!

Missing spam vs important e-mail as spam

False fire alarm vs missing a fire

Discussion

Fire alarm errors

Types of outcomes

$$y = \begin{cases} 1 & \text{event happened} \\ 0 & \text{event not happened} \end{cases}$$

True

True positive: $y_i^{est} = y_i^{true} = 1$

True negative: $y_i^{est} = y_i^{true} = 0$

Errors

False positive: $y_i^{est} = 1, y_i^{true} = 0$

False negative: $y_i^{est} = 0, y_i^{true} = 1$

Confusion matrix

$$\left[\begin{array}{c|c} TP & FN \\ \hline FP & TN \end{array}\right]$$

precision $PPV = \frac{TP}{TP + FP}$

true fires among reported fires

sensitivity or recall $TPR = \frac{TP}{TP + FN}$

reported fires among all fires

accuracy $ACC = \frac{TP + TN}{TP + TN + FP + FN}$

fraction of true classifications