Feature Learning for Image Data: from Dictionary Learning to Deep Learning

Jeremy Watt
jermwatt@gmail.com

Reza Borhani
borhani@u.northwestern.edu

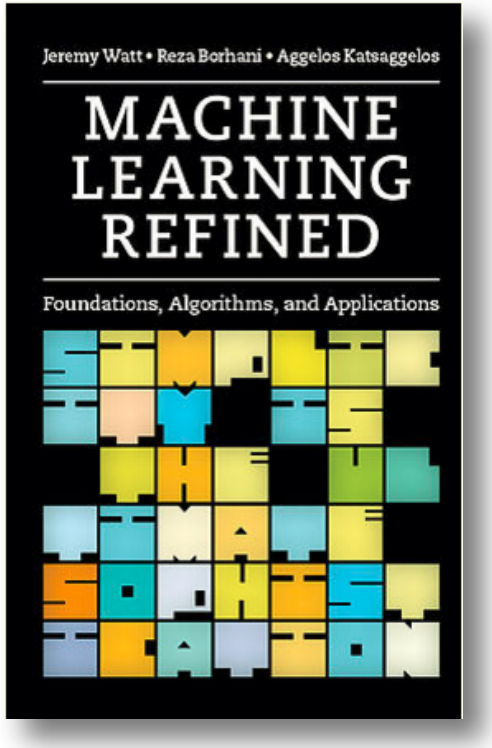
What is this talk about?

- Goal: give you a broad but rigorous introduction to feature learning

- This means we **will** focus on providing
  - **intuition:** what are these tools all about?
  - **context:** how do they fit into the larger picture of machine learning?
  - **perspective:** why do they works so well, and why have neural networks come back now?

- This means we **will not** focus on providing
  - a complete review of current research in the area, although we will discuss major discoveries
  - a user's guide for a particular deep learning software package (there are plenty of these already available online e.g., *Udacity* has an entire class on how to use TensorFlow)

What is this talk based on?

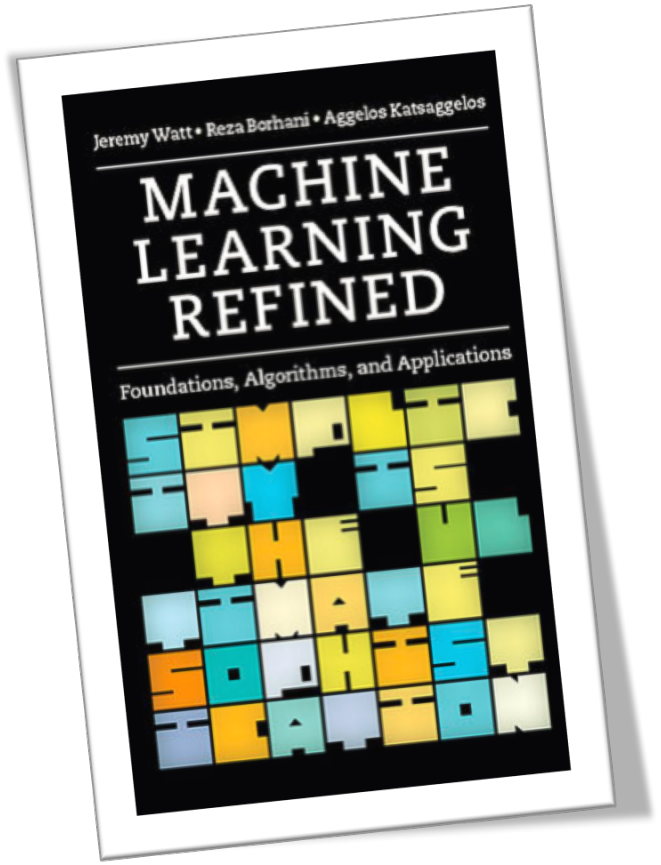


A new ML textbook!
Cambridge University Press
October 2016

Several large ML courses taught at
Northwestern University
Winter 2014, Winter 2015, Fall 2015

What makes our book unique?

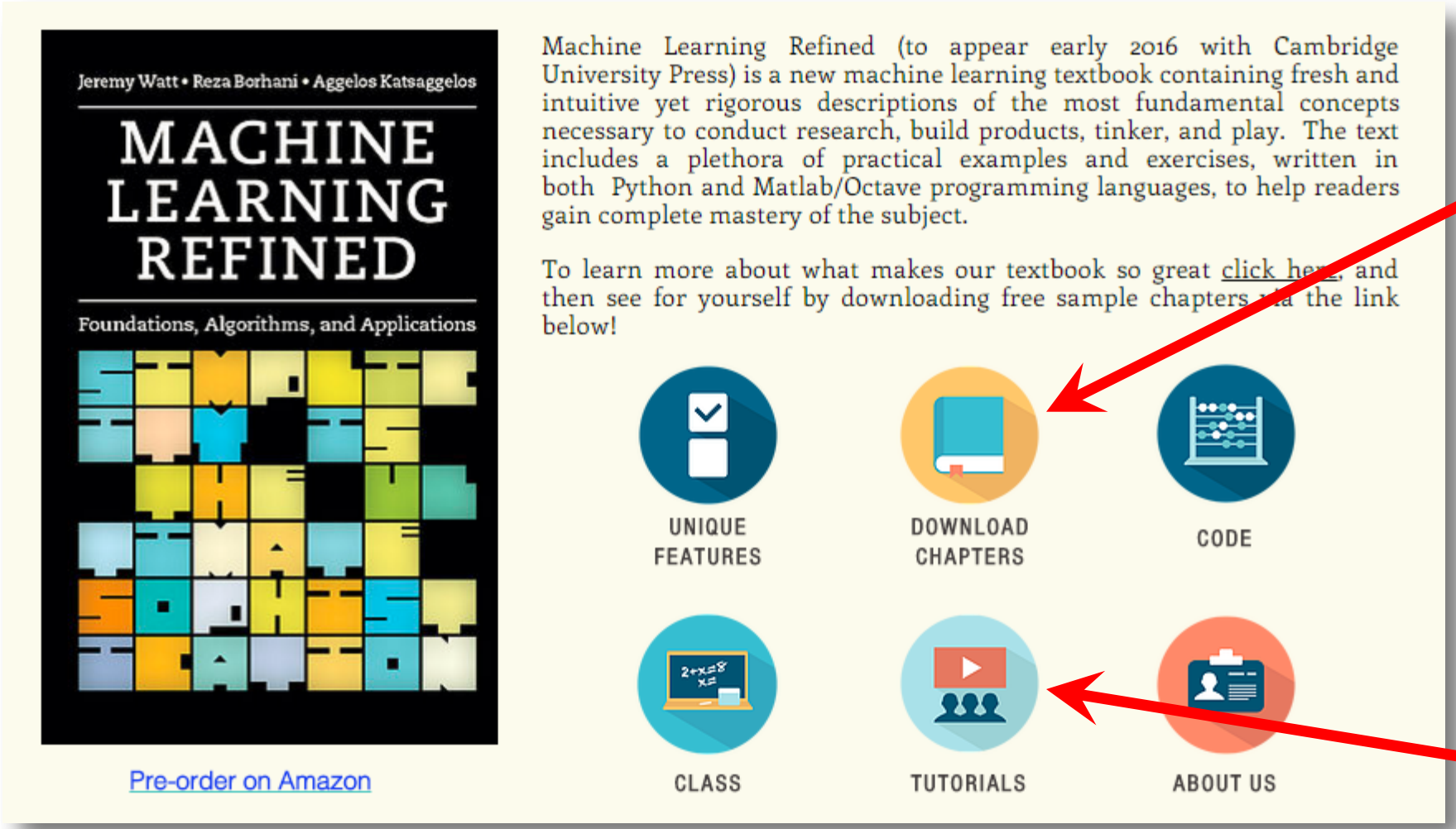
- A presentation built on lucid geometric intuition
  (w/ 200+ color illustrations)
- A learning experience where students learn by doing
  (w/ 50+ in depth coding exercises)
- Inclusion of real application to motivate concepts
  (computer vision, natural language processing, economics, neuroscience, recommender systems, physics, biology, etc.)



- A rigorous yet user friendly presentation of state-of-the-art numerical techniques
  (gradient descent, Newton's method, stochastic gradient descent, accelerated gradient descent, Lipschitz step-length rule, adaptive step-length selection, and more)

Where to download the slides?

www.MLrefined.com



Sample chapters of the book

Slides

Today's talk overview

Part I. An overview of features / unsupervised feature learning- principles of feature engineering, unsupervised feature learning, dictionary learning / sparse coding

Part II. Supervised feature learning – elements of function approximation, neural networks and fixed basis kernels, learning features for regression and classification

Part III. Deep learning and nonlinear optimization – a host of reasons, convolutional networks, a primer on the backpropogation algorithm

Part I

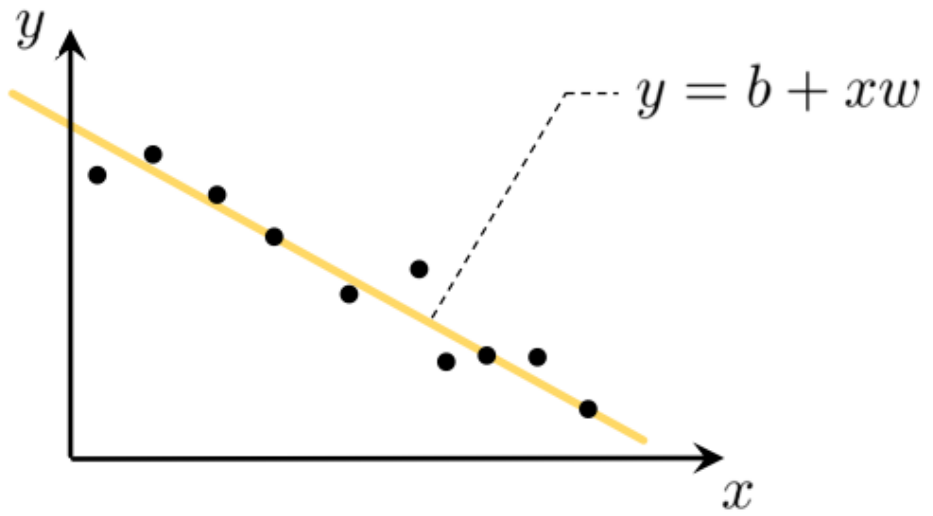
Linear Regression

Linear regression

Data: $\{(\mathbf{x}_p, y_p)\}_{p=1}^{P}$

Output: continuous scalar

Input: N-dimensional (here $N = 1, 2$)



Want to find b and $\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix}$ such that

$$b + \mathbf{x}_p^T \mathbf{w} \approx y_p, \quad p = 1, ..., P$$

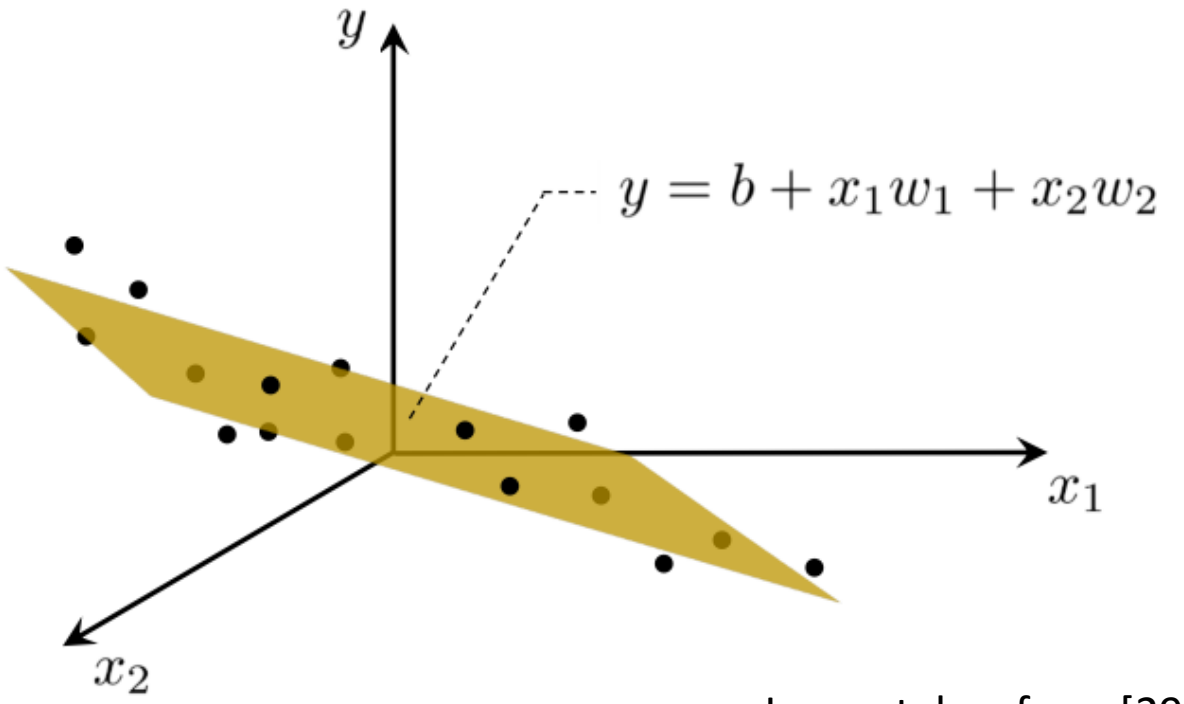
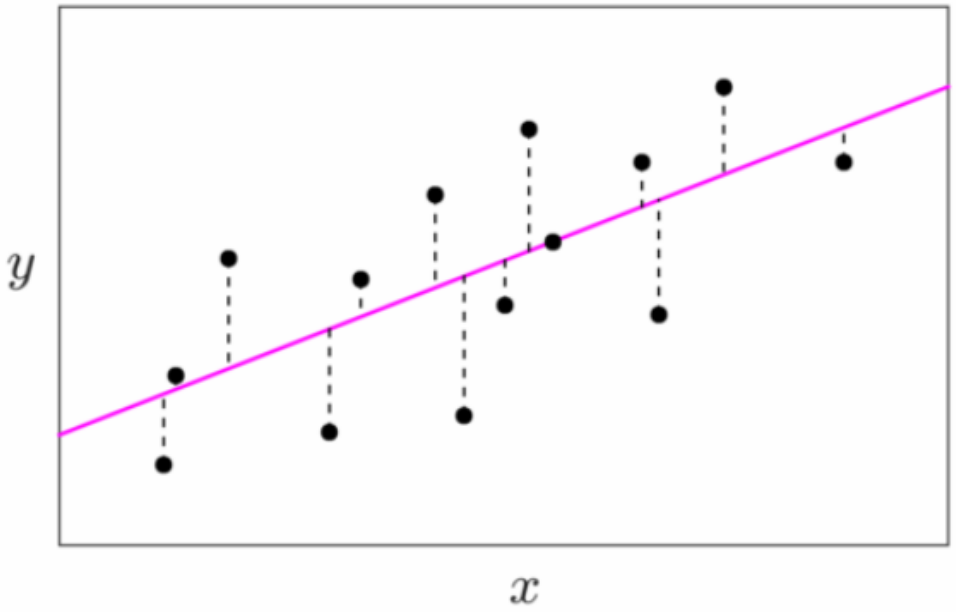


Image taken from [29]

The Least Squares solution

$$b + \mathbf{x}_p^T \mathbf{w} \approx y_p, \quad p = 1, ..., P$$



using some algebraic manipulations (see chap 3 of [29]) one can form a recovery problem whose solution gives parameters satisfying this desired relationship

best parameters $(b^\star, \mathbf{w}^\star)$

$$y_{\text{new}} = b^\star + \mathbf{x}_{\text{new}}^T \mathbf{w}^\star$$

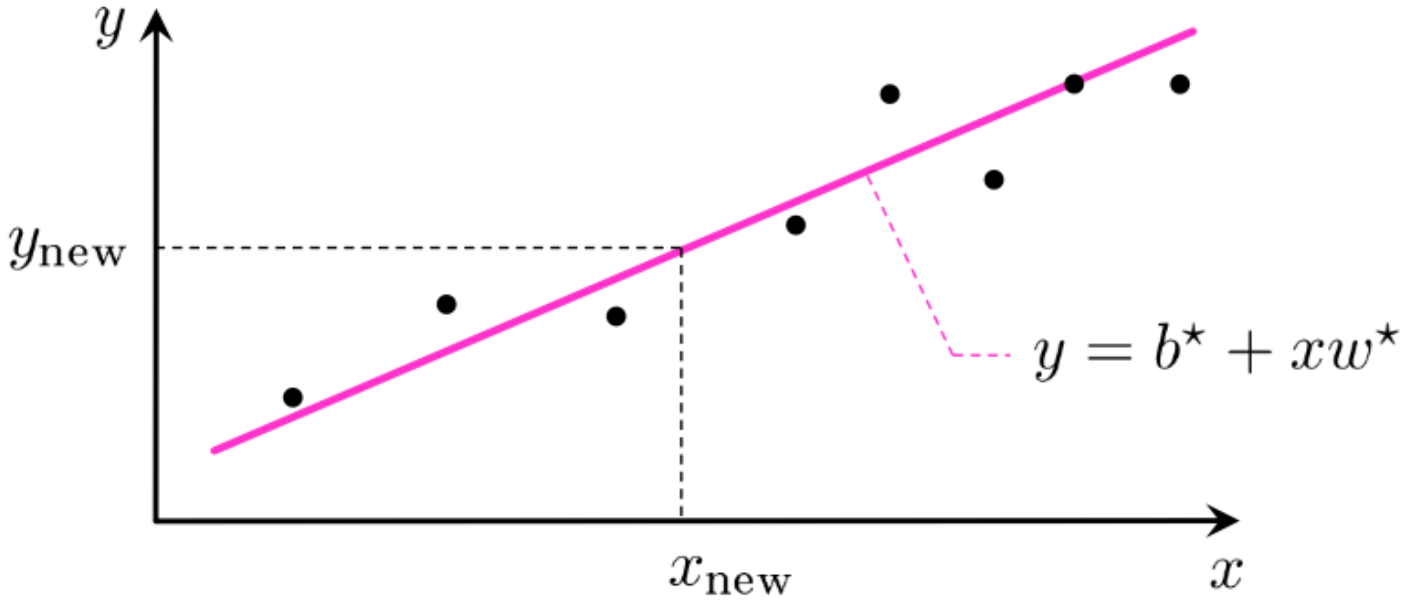


Image taken from [29]

Linear Classification

Logistic regression

$$y(\mathbf{x}) = \text{sign}\left(b + \mathbf{x}^T\mathbf{w}\right)$$

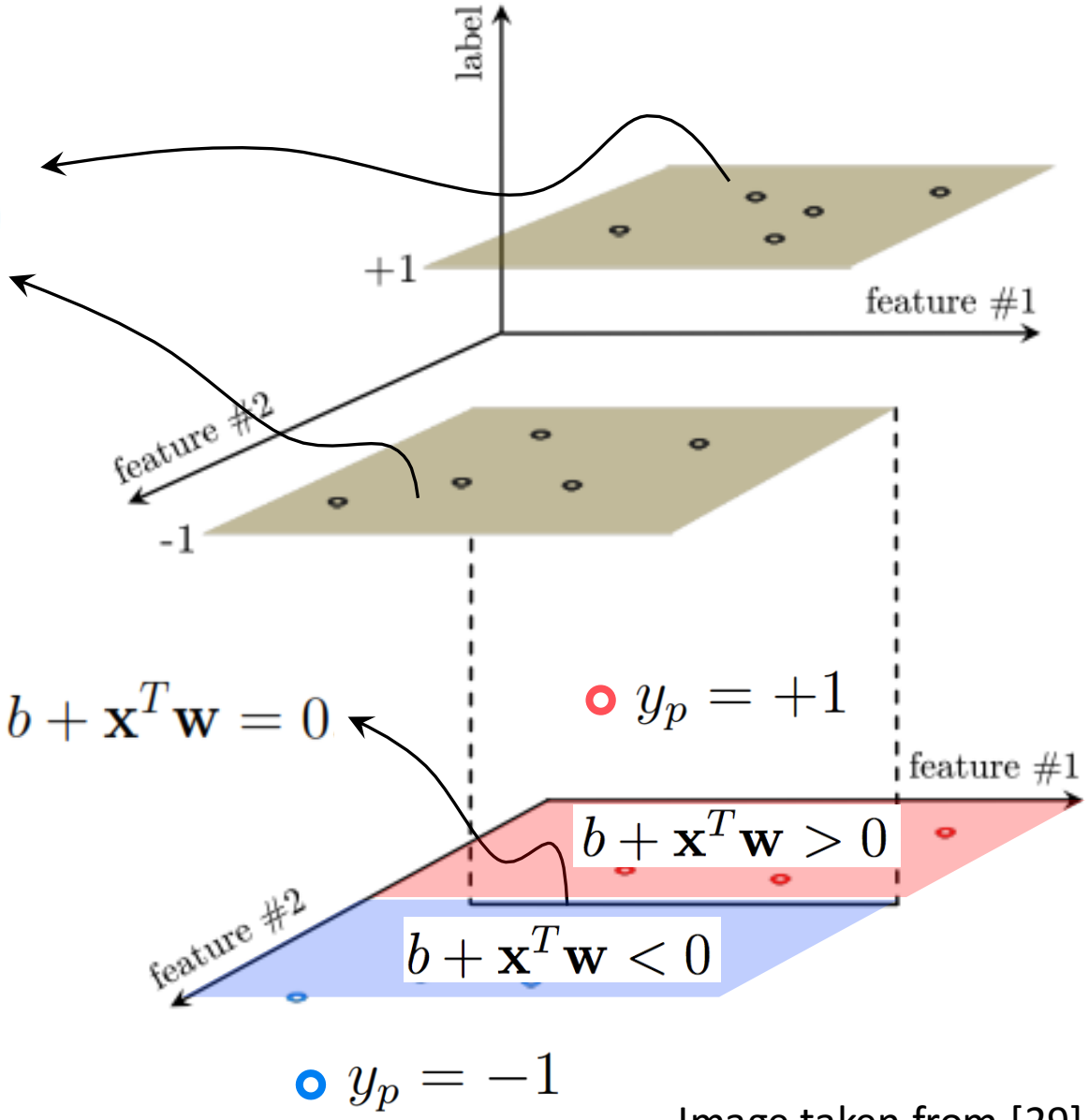


Image taken from [29]

Logistic regression

$$y(\mathbf{x}) = \tanh\left(b + \mathbf{x}^T\mathbf{w}\right)$$

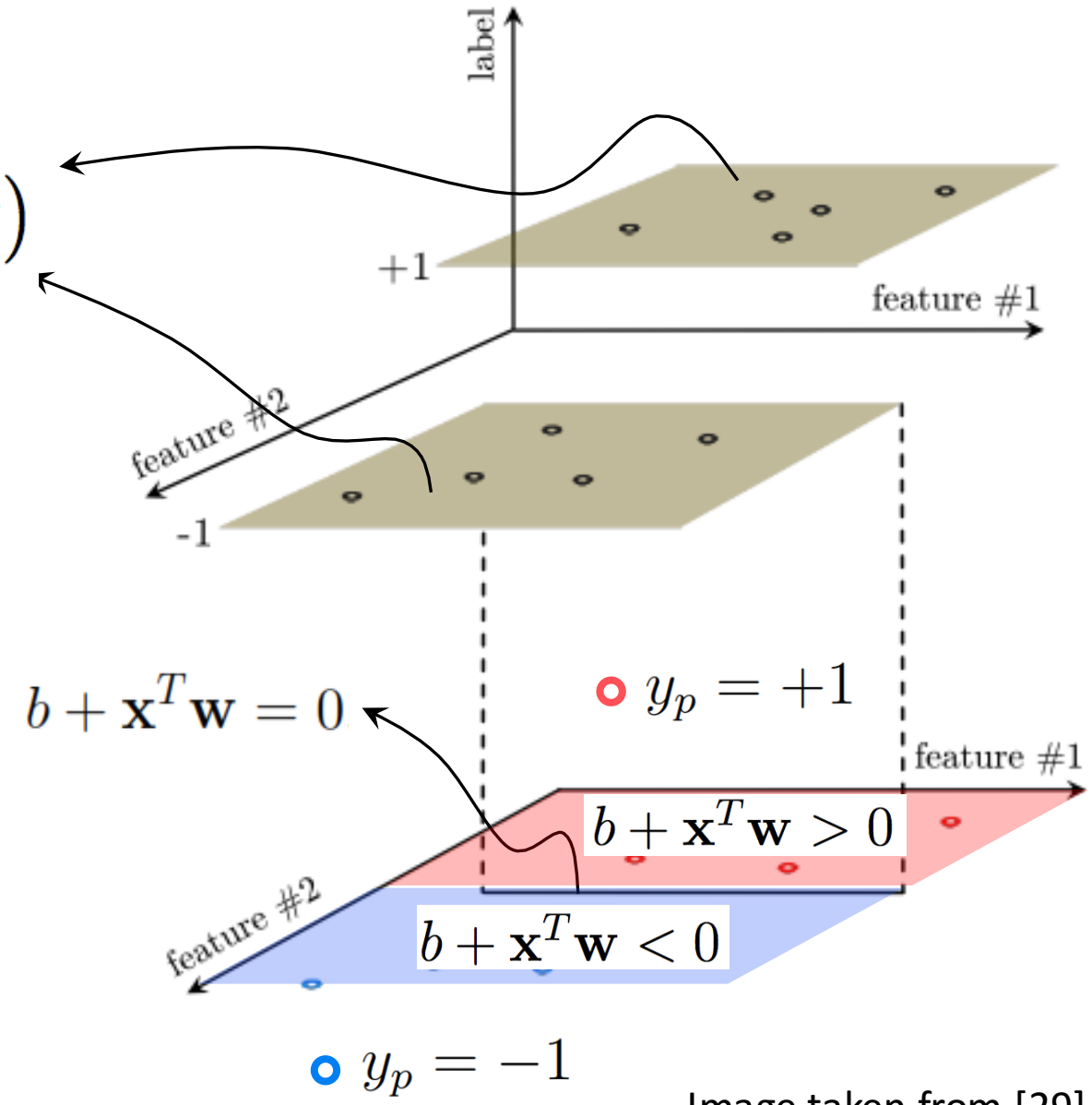


Image taken from [29]

Logistic regression

so we want to tune these parameters so that

$$y_p \approx \tanh\left(b + \mathbf{x}_p^T \mathbf{w}\right) \quad p = 1, ..., P$$

$$\Downarrow$$

using some algebraic manipulations (see chap 4 of [29]) one can form a recovery problem whose solution gives parameters satisfying this desired relationship

$$\Downarrow$$

best parameters $(b^\star, \mathbf{w}^\star)$

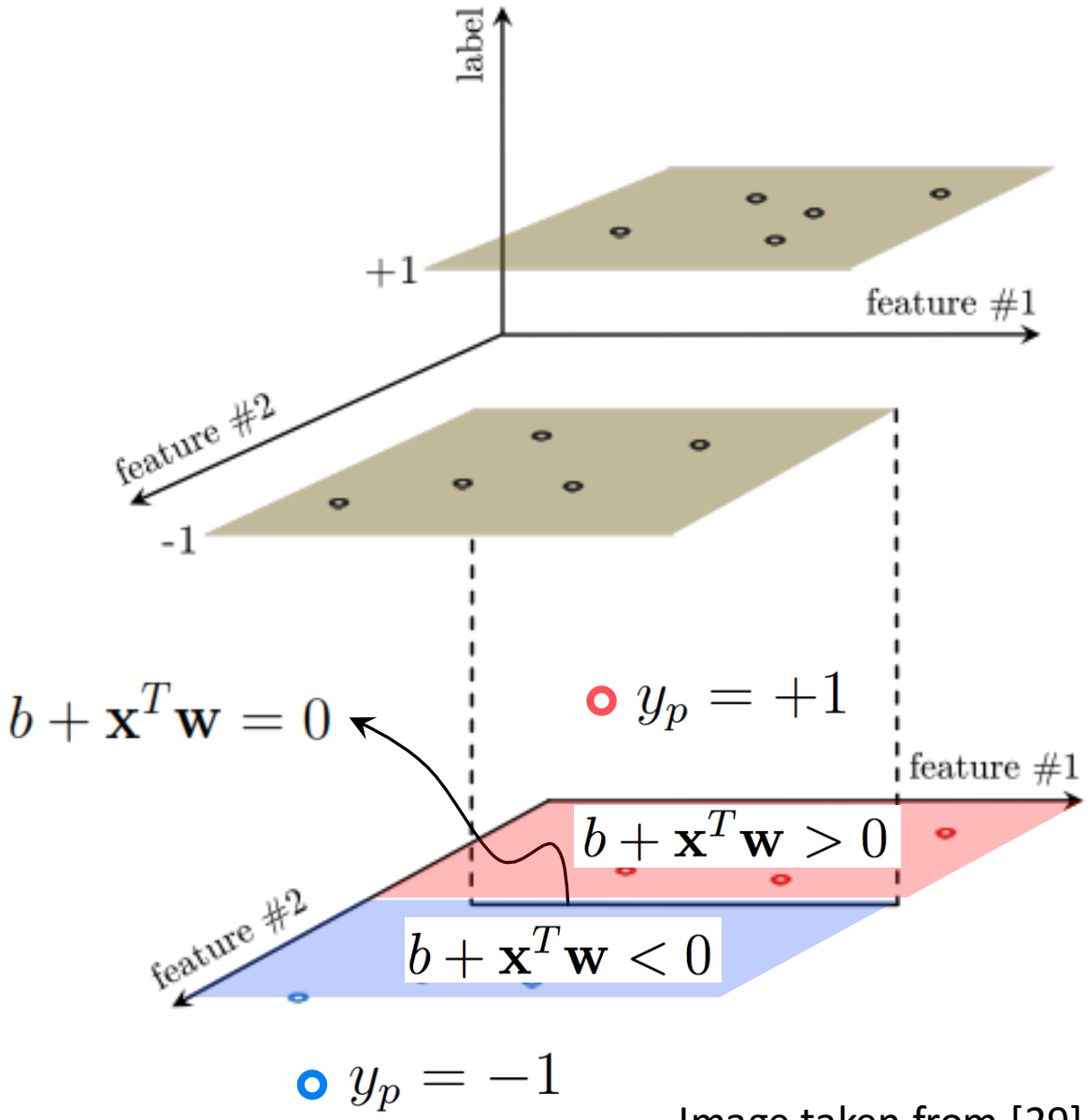


Image taken from [29]

What are “features” ?

Features

What are features?

- **Geometrically:** features are transformations of the input that provoke a good *nonlinear* fit/ separation

- **Philosophically:** features are those defining characteristics of the phenomenon underlying a dataset that allow for optimal learning

Feature design: regression

- Data appears distributed periodically, i.e., for each p it looks like

$$b + f(x_p)\, w = b + \sin(2\pi x_p)\, w \approx y_p$$

  where w needs to be tuned to make $\approx$ as tight as possible

- In ML terms: $f(x_p) = \sin(2\pi x_p)$ called a *feature transformation*
- Note the feature and output are *linearly* related in w

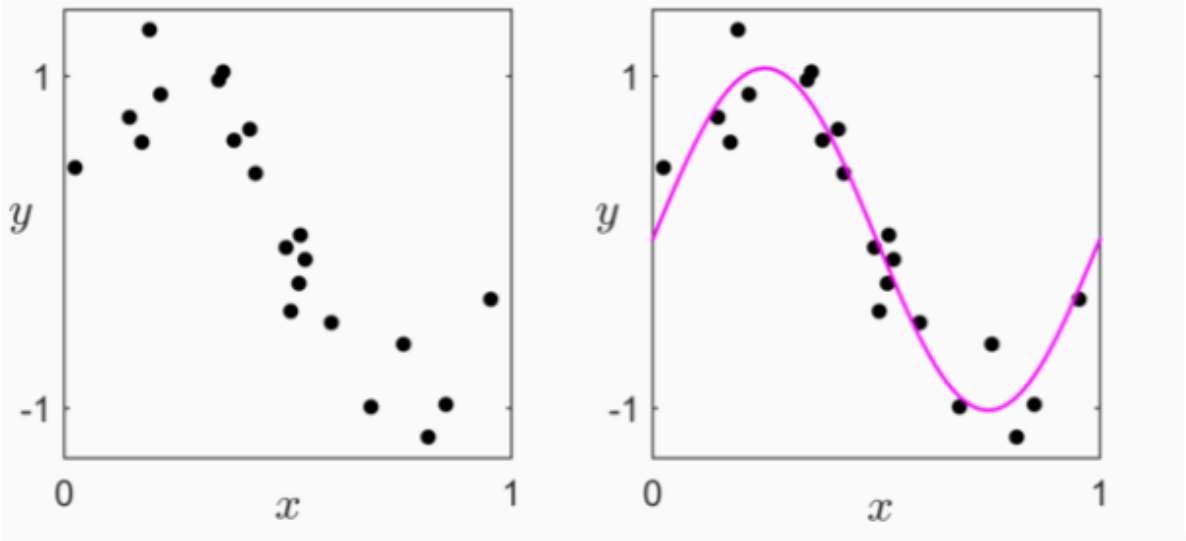


Image taken from [29]

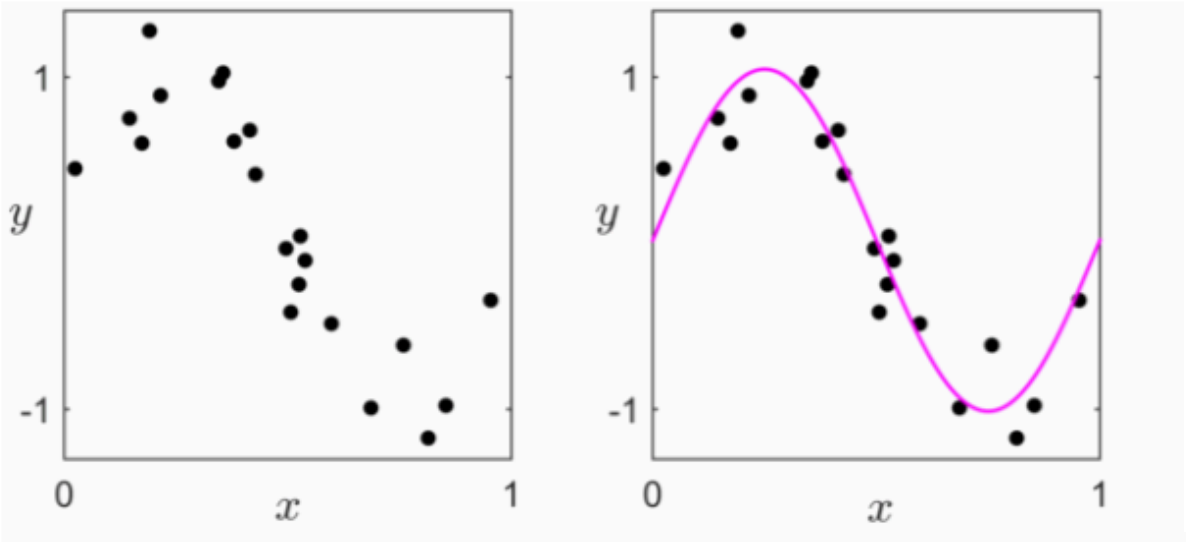
Feature design: regression

- Data appears distributed periodically, i.e., for each p it looks like

$$b + f\left(x_p\right) w = b + \sin\left(2\pi x_p\right) w \approx y_p$$

where w needs to be tuned to make $\approx$ as tight as possible

- Tune w so that it minimizes the squared between each feature and its corresponding output



$$\underset{w}{\text{minimize}} \sum_{p=1}^{P} \left(f\left(x_p\right) w - y_p\right)^2$$

can solve for optimal w in closed form or use iterative procedure like e.g., gradient descent [29]

Image taken from [29]

Feature design: regression

- Data appears distributed periodically, i.e., for each p it looks like

$$b + f\left(x_p\right) w = b + \sin\left(2\pi x_p\right) w \approx y_p$$

where w needs to be tuned to make $\approx$ as tight as possible

- Note the feature and output are *linearly* related in w

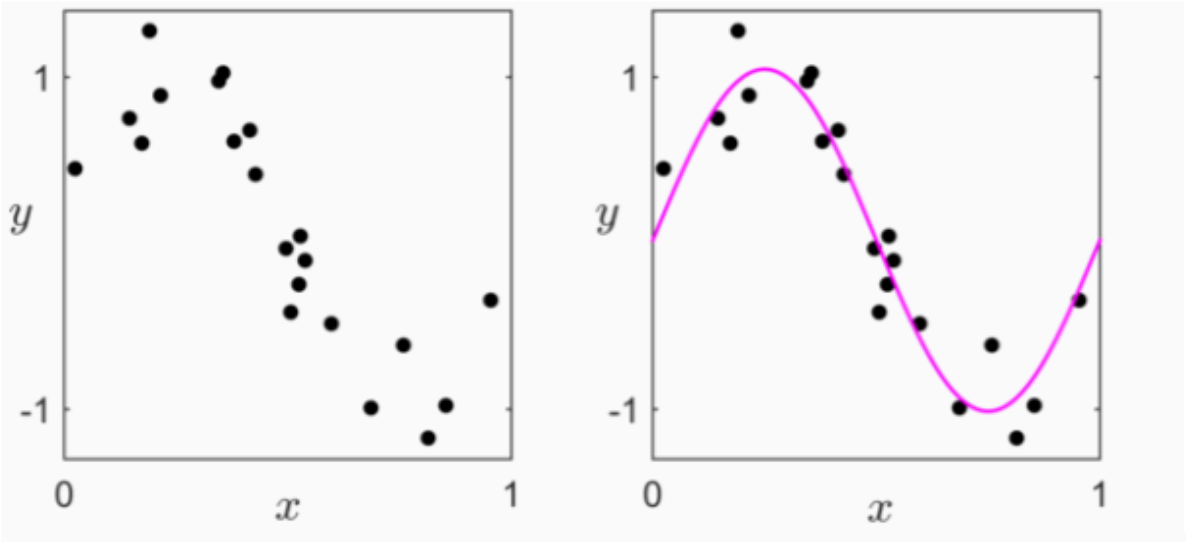


Image taken from [29]

Feature design: regression

A properly designed feature (or set of features) for linear regression provides a good *nonlinear* fit in the original feature space and, simultaneously, a good *linear* fit in the transformed feature space.

$$b + f(x)\,w = b + \sin(x)\,w \approx y$$

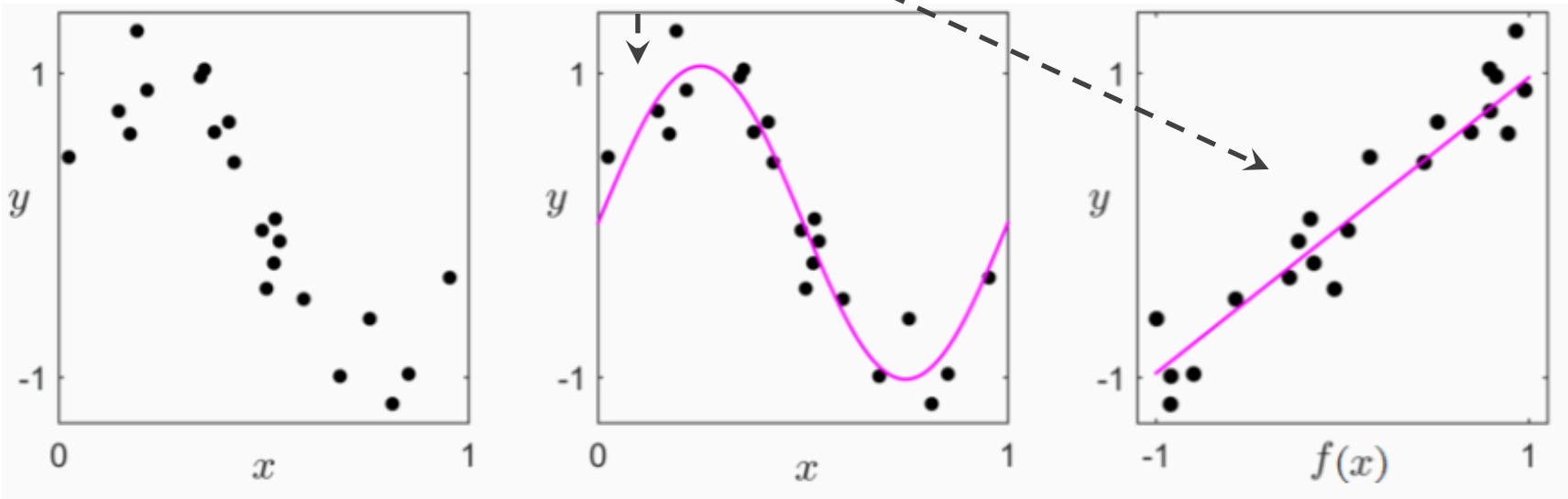


Image taken from [29]

Feature design: regression

A properly designed feature (or set of features) for linear regression provides a good *nonlinear* fit in the original feature space and, simultaneously, a good *linear* fit in the transformed feature space.

$$b + \sum_{m=1}^{M} f_m(x_p)\, w_m \approx y_p$$

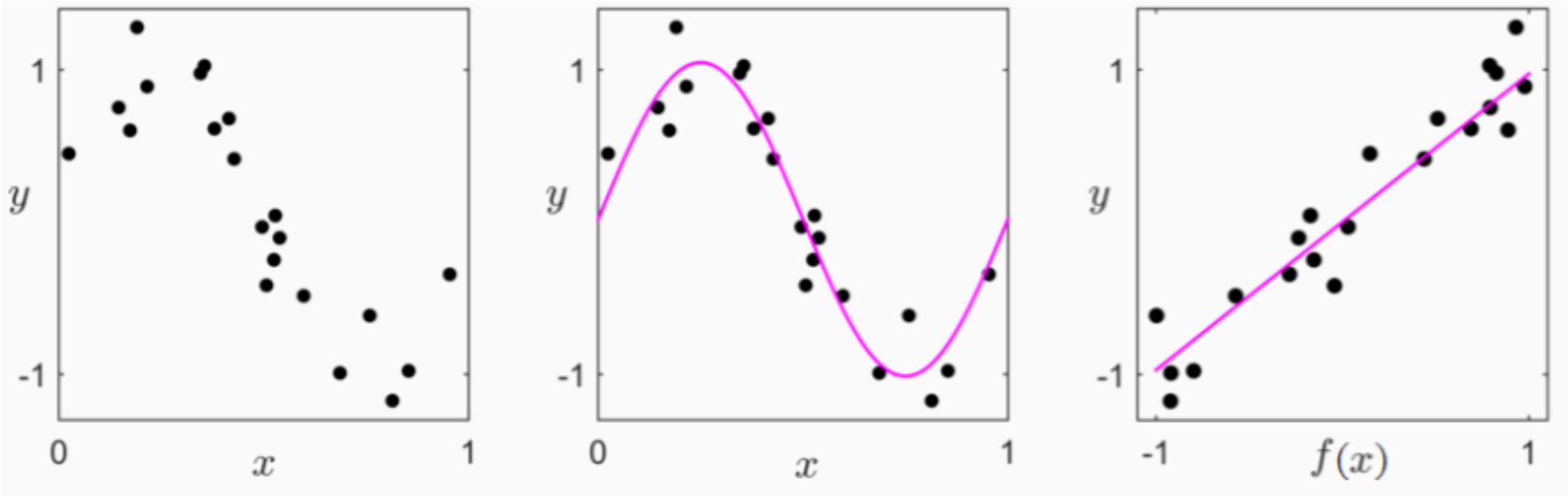


Image taken from [29]

Feature design: regression

What are we aiming to do with features?

- The data we receive are noisy samples from an underlying function, the phenomenon (e.g., gravity) we wish to understand/predict

- In determining correct features we aim to approximate this function from the data as well as possible

- Estimate this as $b + \sum_{m=1}^{M} f_m(\mathbf{x})\, w_m$

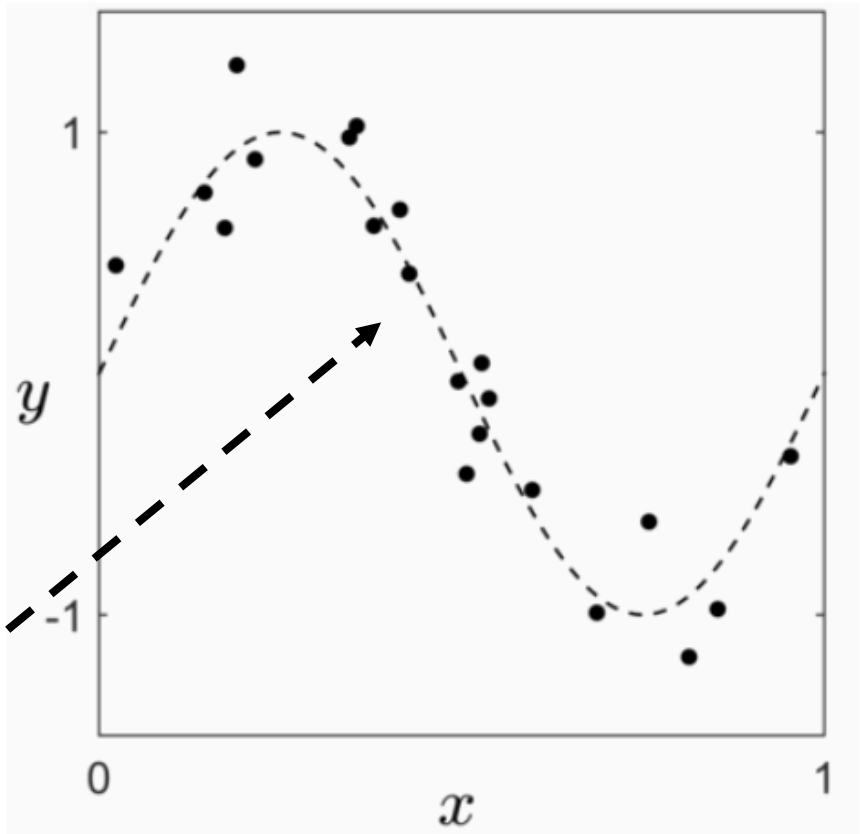


Image taken from [29]

Feature design: regression (re-visited)

- intuition can be hard to come by
- inputs are rarely one/two dimensional, can be hard to visualize data
- even so it can be very hard to determine proper feature transformation 'by eye'
- if we have enough (relatively clean) data we can *learn* the appropriate features automatically

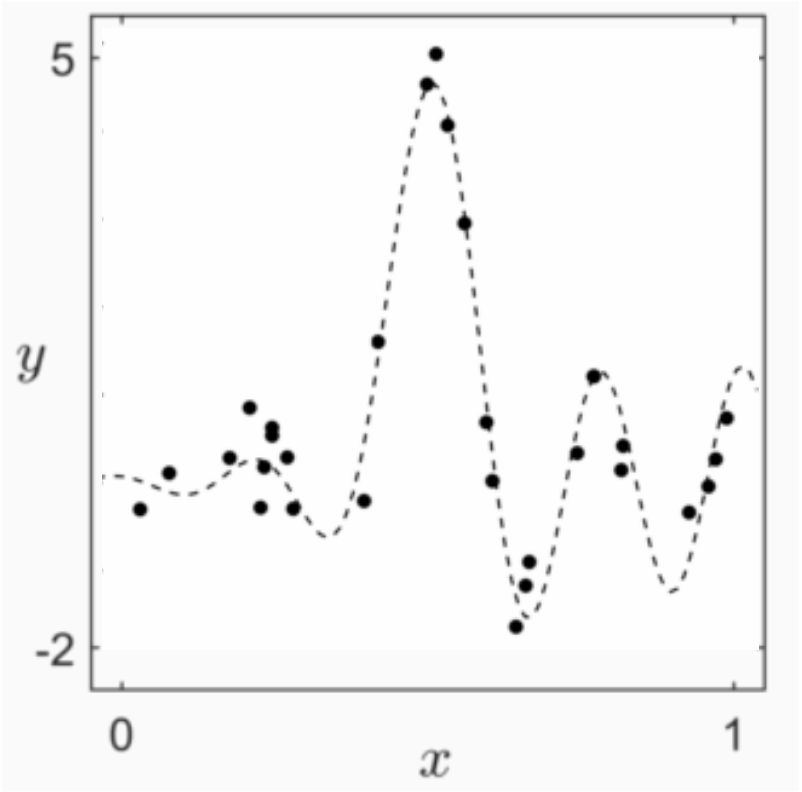
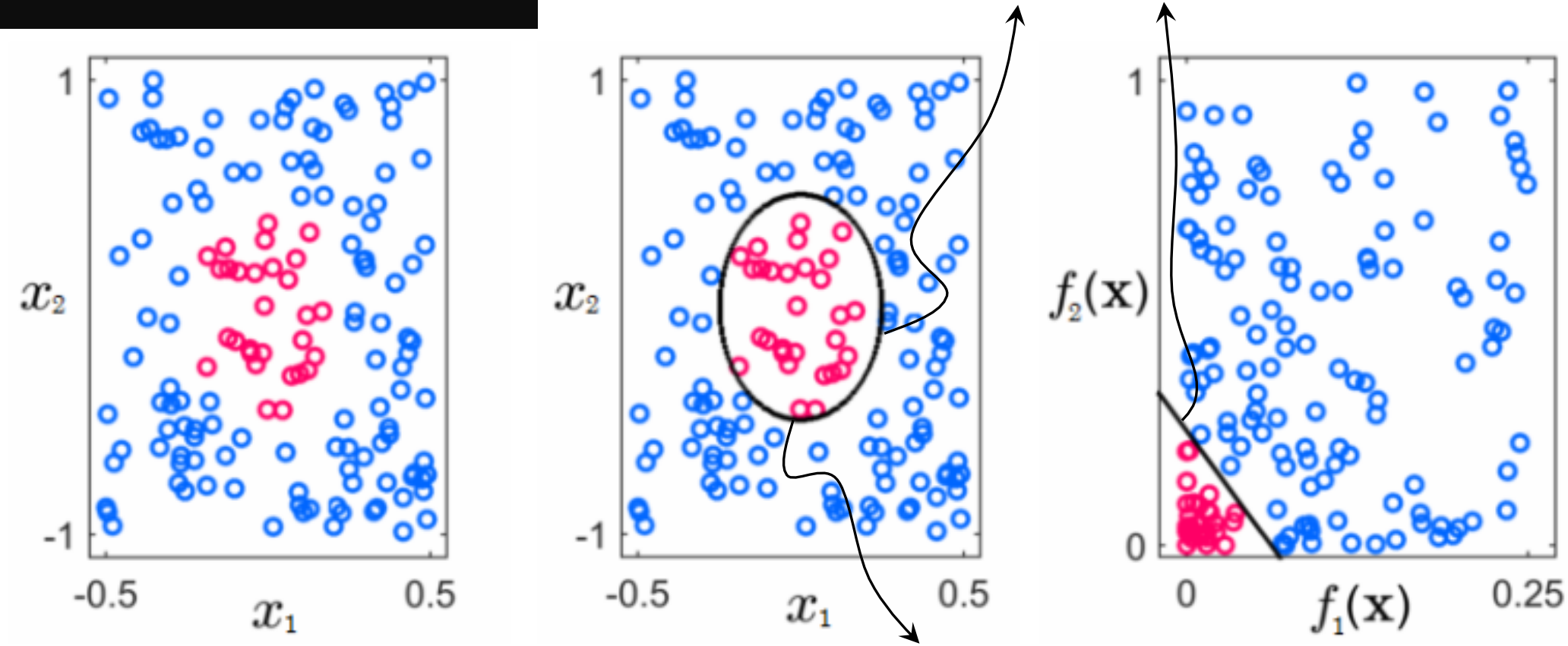


Image taken from [29]

Feature design: classification

$$1 + f_1(\mathbf{x})\, w_1 + f_2(\mathbf{x})\, w_2 = 0$$

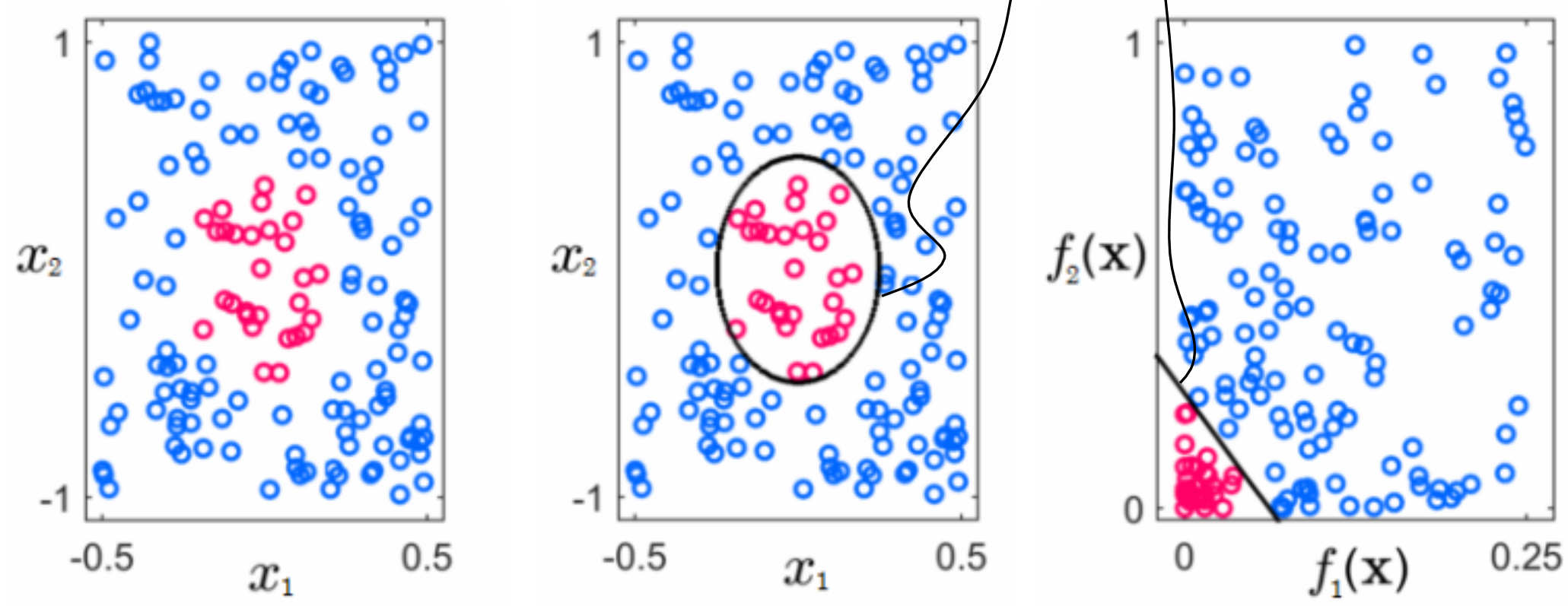


$$1 + x_1^2 w_1 + x_2^2 w_2 = 0$$

Note the feature and output are *linearly* related in w_1 and w_2

Feature design: classification

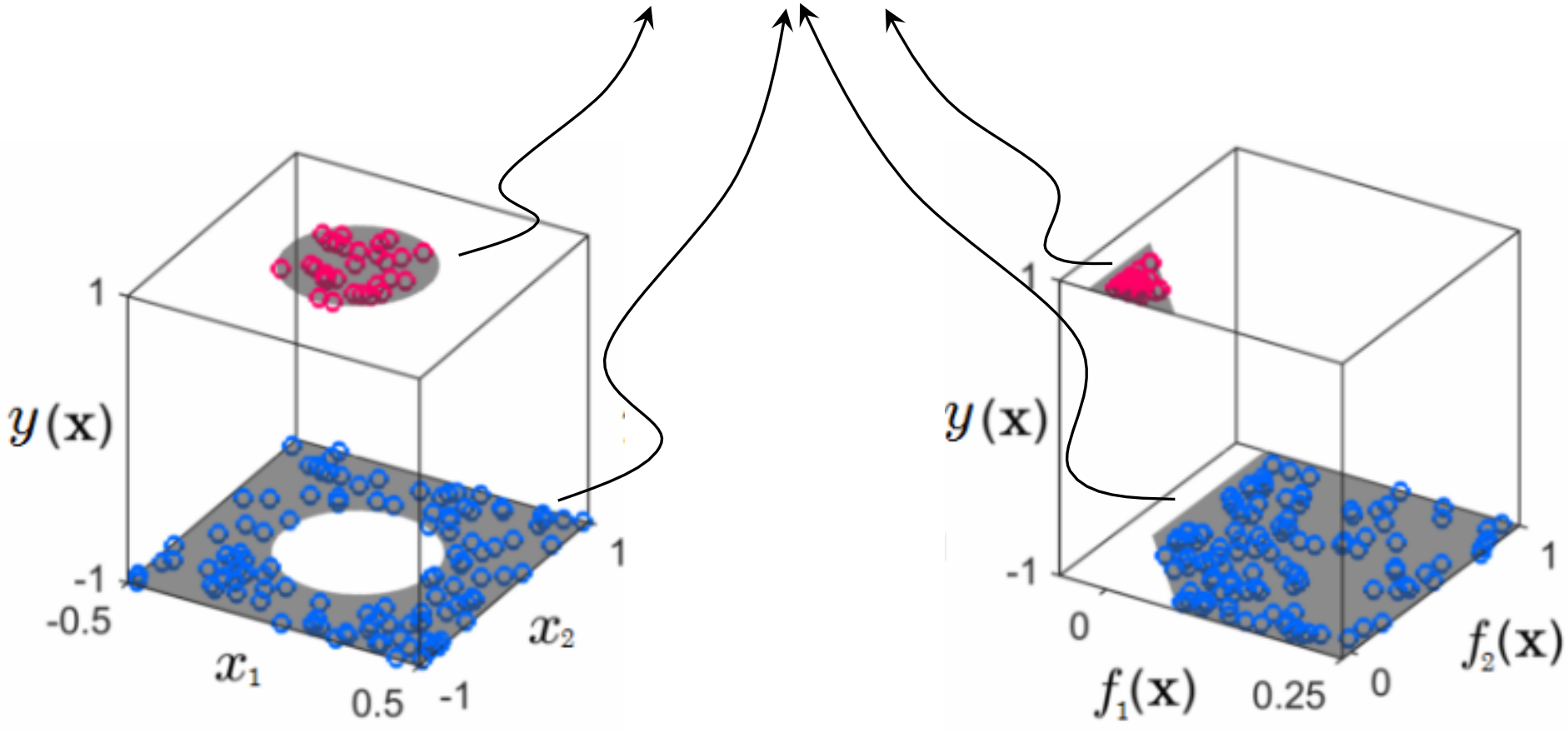
$$1 + f_1(\mathbf{x})\, w_1 + f_2(\mathbf{x})\, w_2 = 0$$



Properly designed features for linear classification provide good *nonlinear* separation in the original feature space and, simultaneously, good *linear* separation in the transformed feature space.

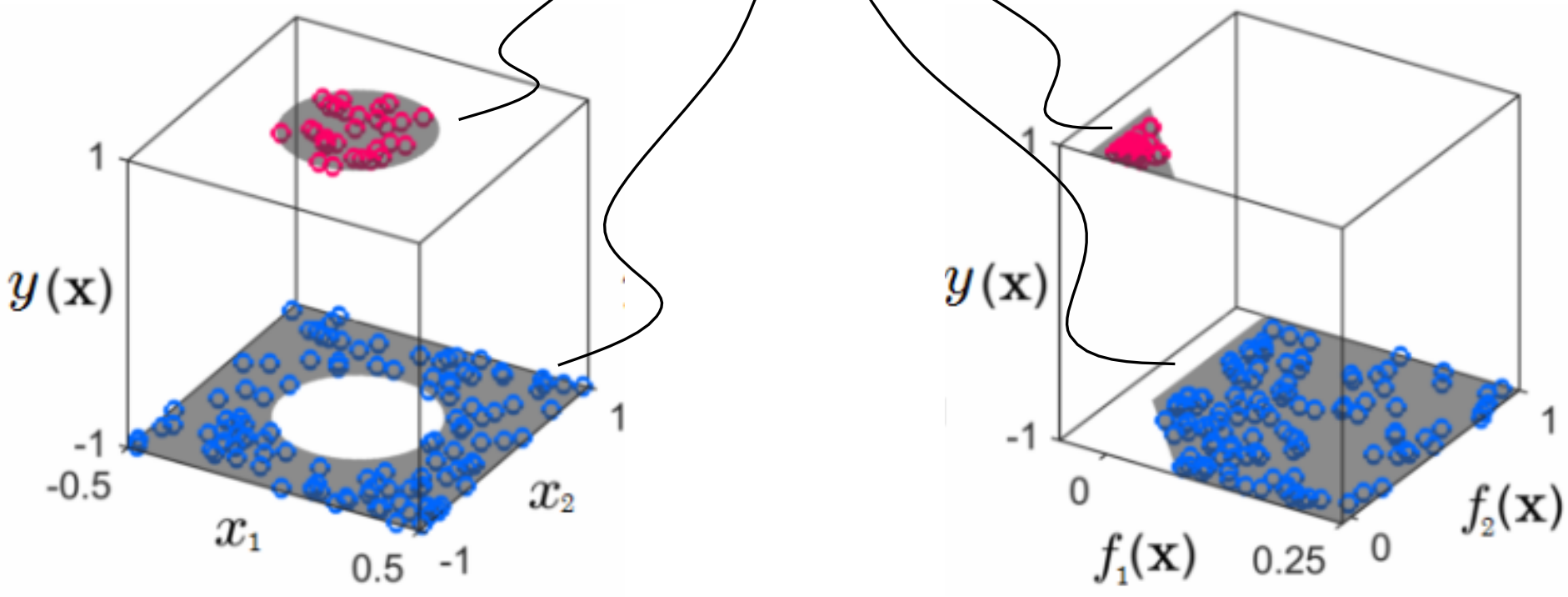
Full view of data

$$y(\mathbf{x}) = \text{sign}\left(1 + f_1(\mathbf{x})\, w_1 + f_2(\mathbf{x})\, w_2\right)$$



Full view of data

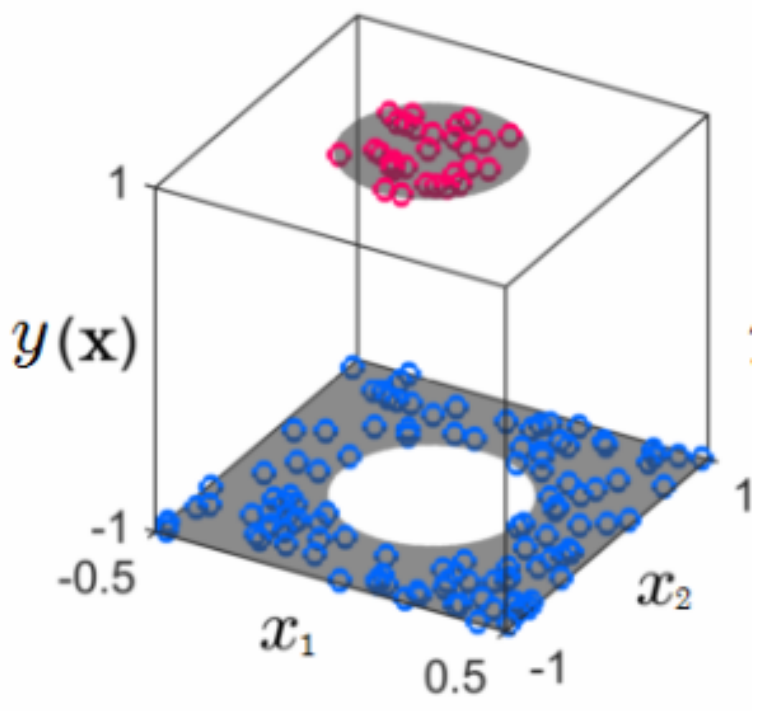
$$y(\mathbf{x}) = \text{sign}\left(b + \sum_{m=1}^{M} f_m(\mathbf{x})\, w_m\right)$$



Feature design: classification

What are we aiming to do with features?

- The data we receive are noisy samples from an underlying binary function, the phenomenon we wish understand / predict
- In determining correct features we aim to approximate this function from the data as well as possible



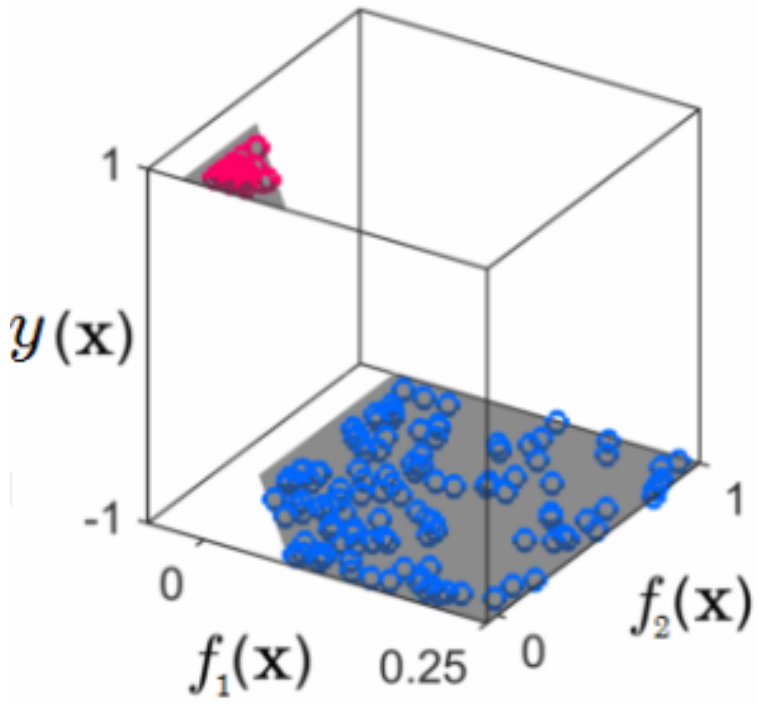
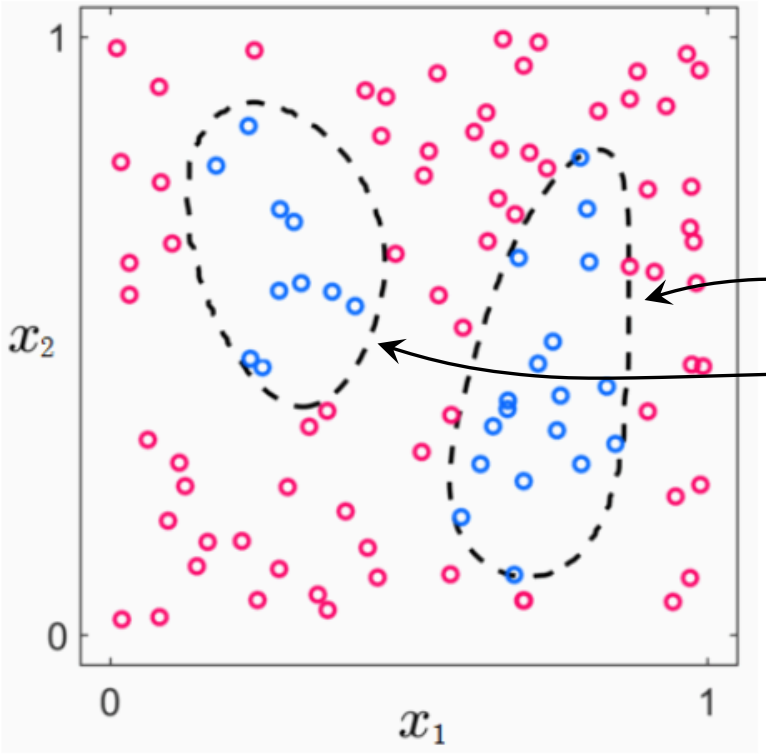


Image taken from [29]

Feature design: classification

- intuition can be hard to come by
- inputs are rarely one/two dimensional, can be hard to visualize data
- can be hard to determine proper feature transformation 'by eye'
- with enough (relatively clean) data we can *learn* good features



$$b + \sum_{m=1}^{M} f_m(\mathbf{x})\, w_m = 0$$

Image taken from [29]

Features: summary (re-visited)

What are features?

- **Geometrically:** features are transformations of the input that provoke a good *nonlinear* fit/separation

- **Philosophically:** features are those defining characteristics of the phenomenon underlying a dataset that allow for optimal learning

How do we design good features?

- **By hand (engineered):** translate understanding of a phenomenon into a set of geometric transformations of the input – often requires expertise
  - Requires strong knowledge to produce reasonable results
- **Learn from data (automatic):** use bases (e.g., neural networks) to determine directly from the data
  - Requires large datasets to produce reasonable results (more on this in part II of talk!)

Commonly-used engineered features

TEXT

- Sentiment analysis

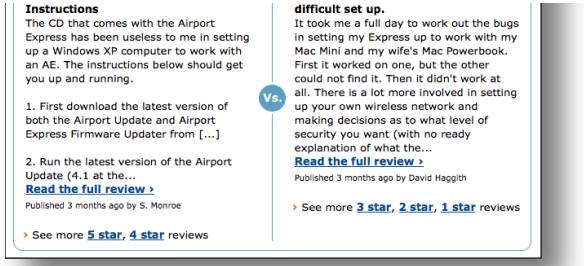
IMAGE

- Pedestrian detection

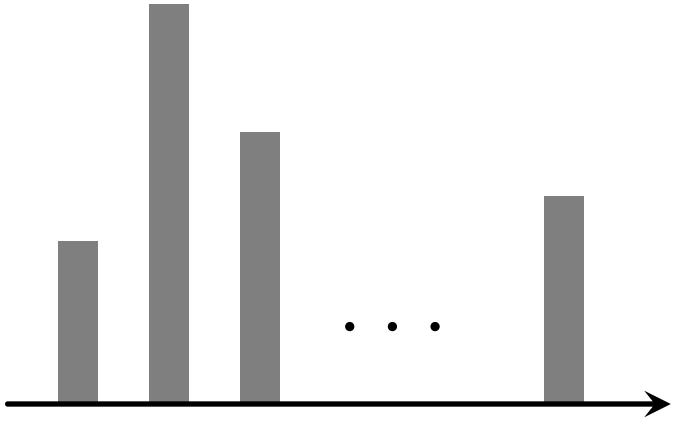
AUDIO

- Speech recognition

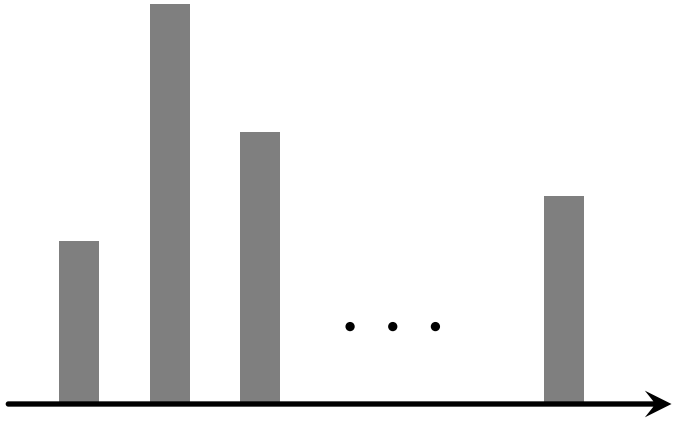
Feature transformations for real data often consist of discrete processing steps which aim at ensuring that instances within a single class are "similar" while those from different classes are "dissimilar". These processing steps are still feature transformations $f_1(\mathbf{x}), ..., f_M(\mathbf{x})$ of the input data, but they are not so easily expressed algebraically.



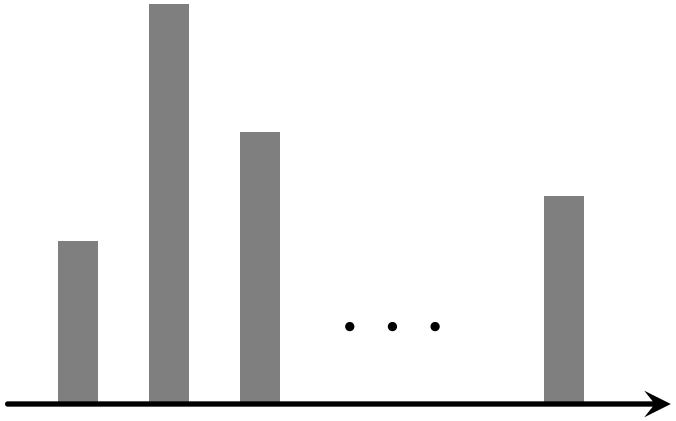
Histogram features



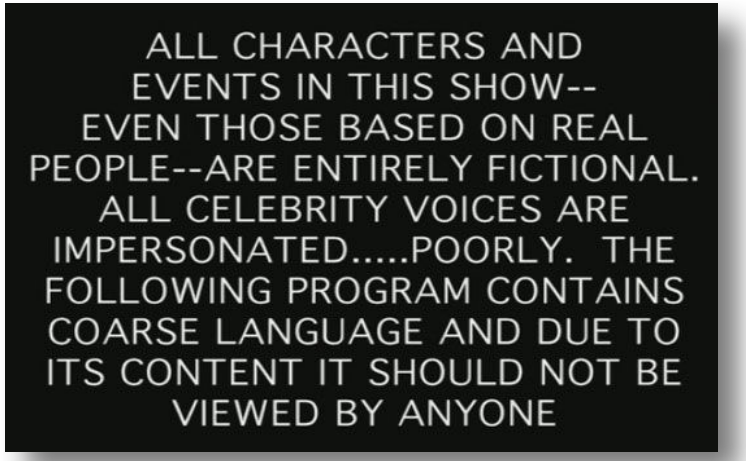
words

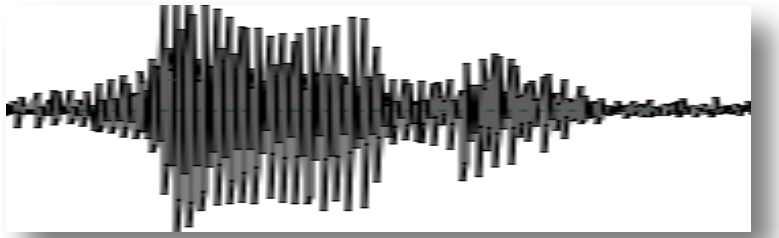


edge content



frequency





Bag of Words (BoW) histogram

1) dogs are the best
2) cats are the worst

- ❑ Parsing
- ❑ Stop word removal
- ❑ Stemming
- ❑ Merging
- ❑ Normalized vector representation

Parsing	dogs / are / the / best	cats / are / the / worst
Stop word removal	dogs / ~~are~~ / ~~the~~ / best	cats / ~~are~~ / ~~the~~ / worst
Stemming	dog~~s~~ best	cat~~s~~ worst
Merging	best cat dog worst	

$$\mathbf{x}_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \begin{pmatrix} \text{best} \\ \text{cat} \\ \text{dog} \\ \text{worst} \end{pmatrix} \qquad \mathbf{x}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \begin{pmatrix} \text{best} \\ \text{cat} \\ \text{dog} \\ \text{worst} \end{pmatrix}$$

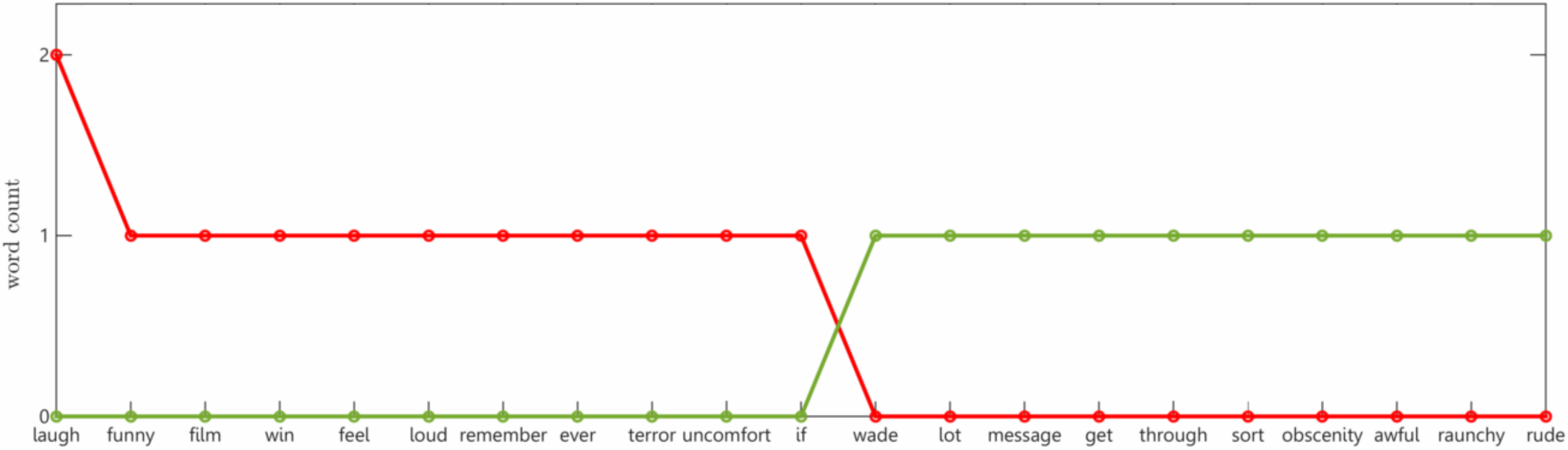
Sentiment analysis with BoWs

One of the funniest films you will ever feel this uncomfortable about laughing out loud at. Remember if you laugh, the terrorists win!

Kevin Ranson
MovieCrypt.com

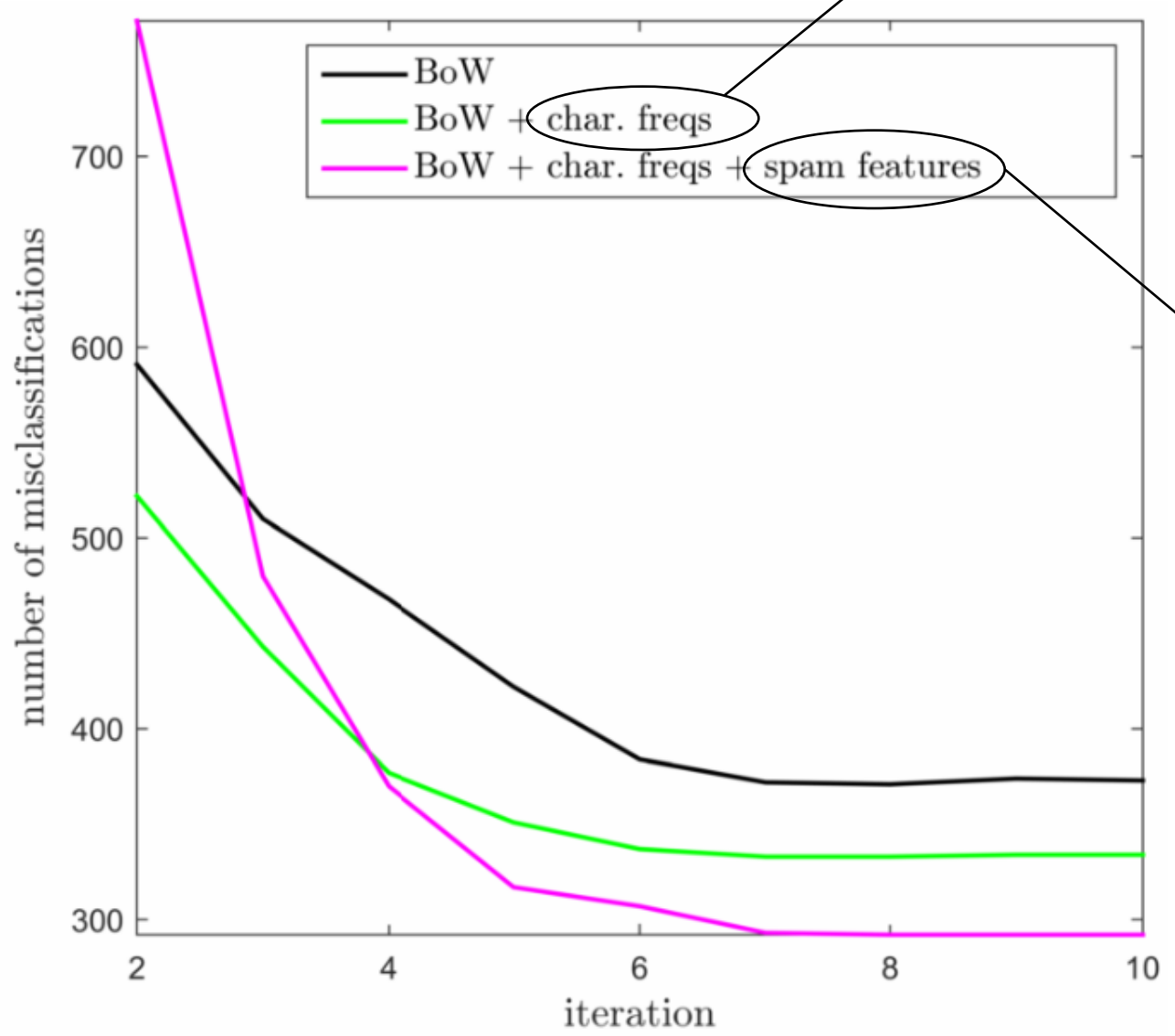
It is rude and raunchy, and it has a message, sort of. You have to wade through an awful lot of obscenity to get to it though.

Roger Moore
Orlando Sentinel



Spam detection

Frequency of certain characters like ‘ ! ’ and ‘ * ’



- Percentage of capital letters
- Length of longest uninterrupted sequence of capital letters
- …

Classification ⇒ distinguish between different classes of data

- **Object detection and recognition**:
  - for identification, organizing/taking better photos (e.g., [25])

Classification $\Rightarrow$ distinguish between different classes of data

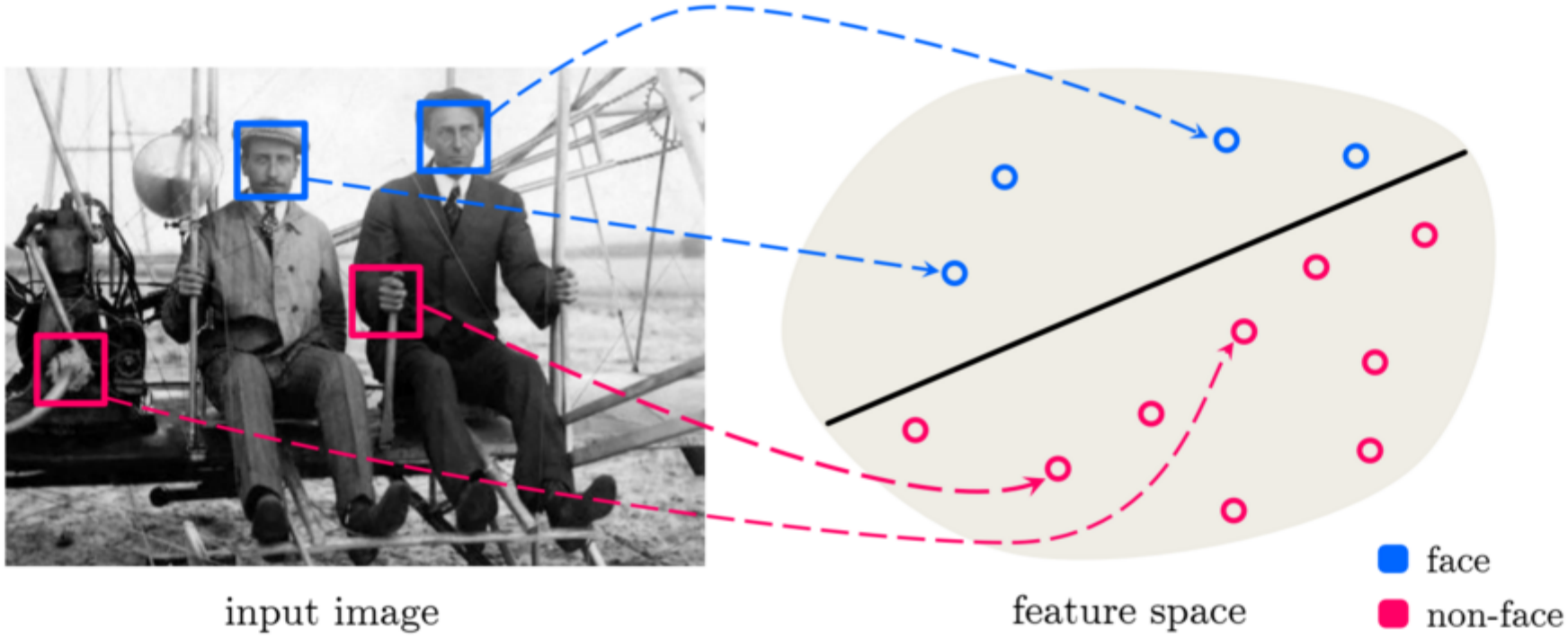


Image taken from [29]

Classification ⇒ distinguish between different classes of data

- **Object detection and recognition**:
  - for human-computer interaction (e.g., [2,9,18,32])

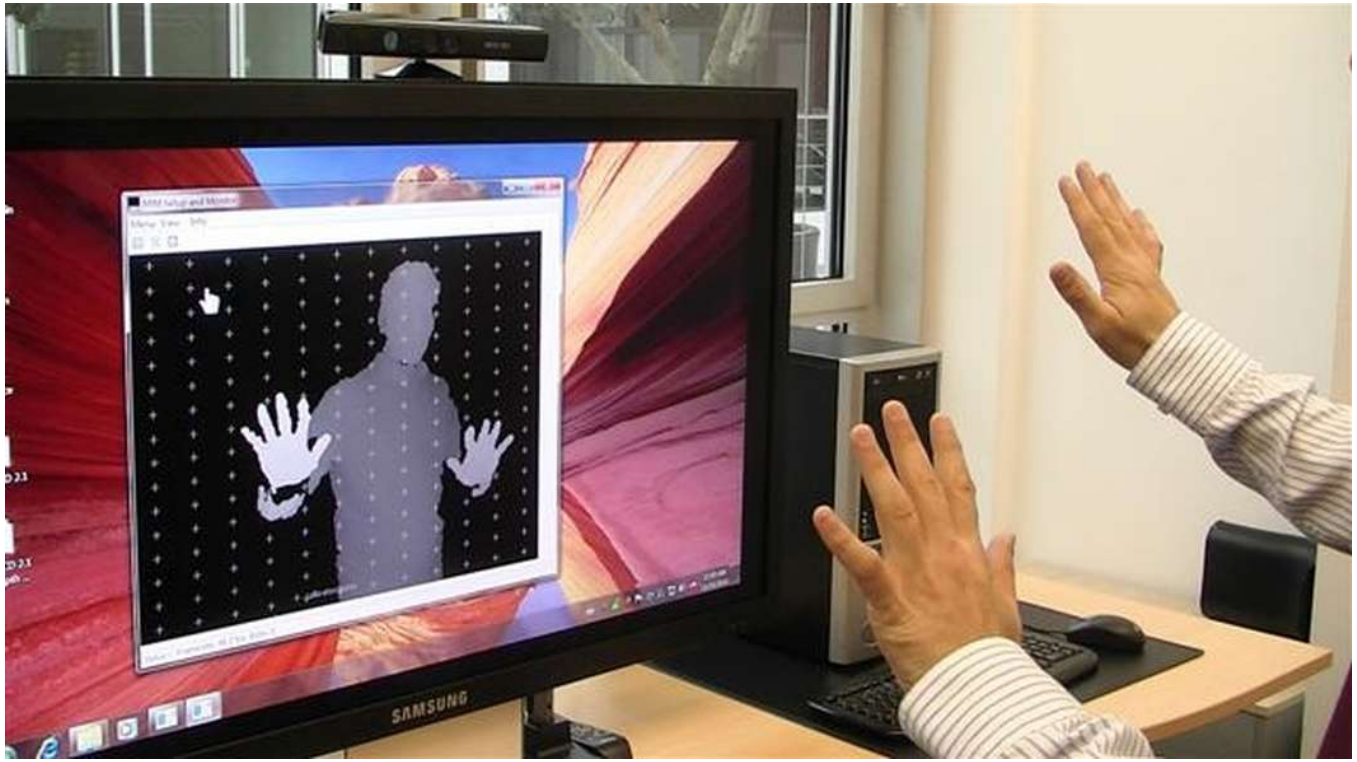
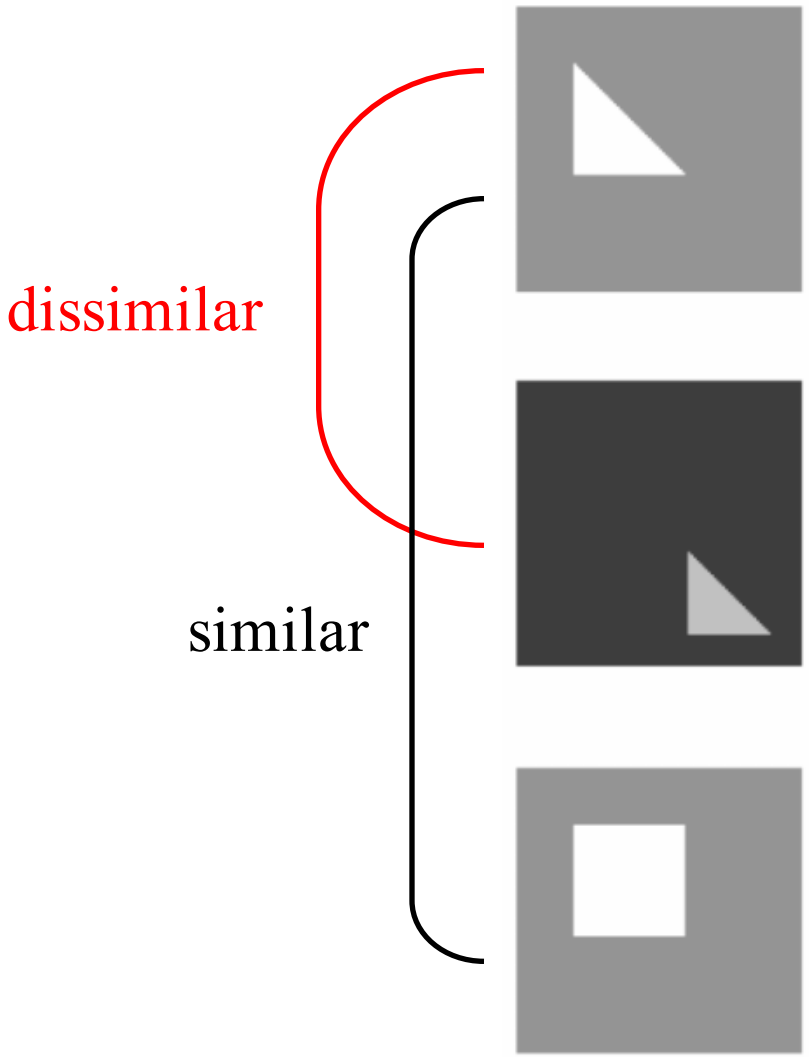


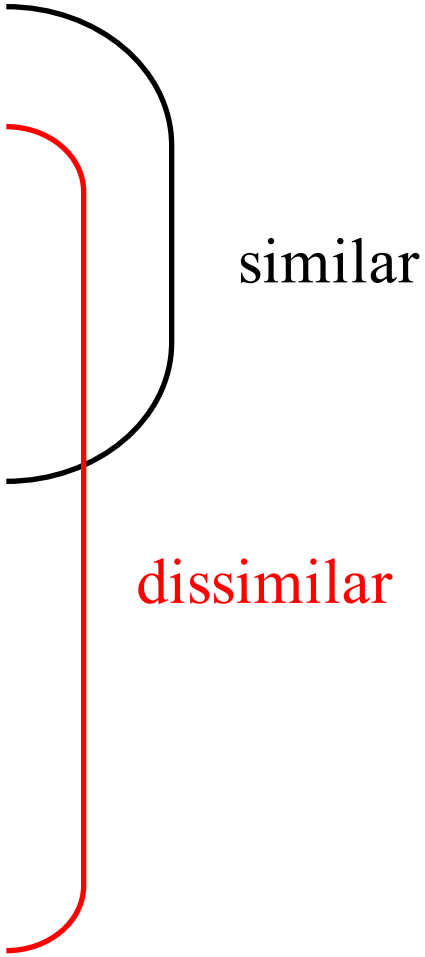
Image taken from http://www.gizmag.com/kinect-used-to-control-windows-7/16997/

Images and edges

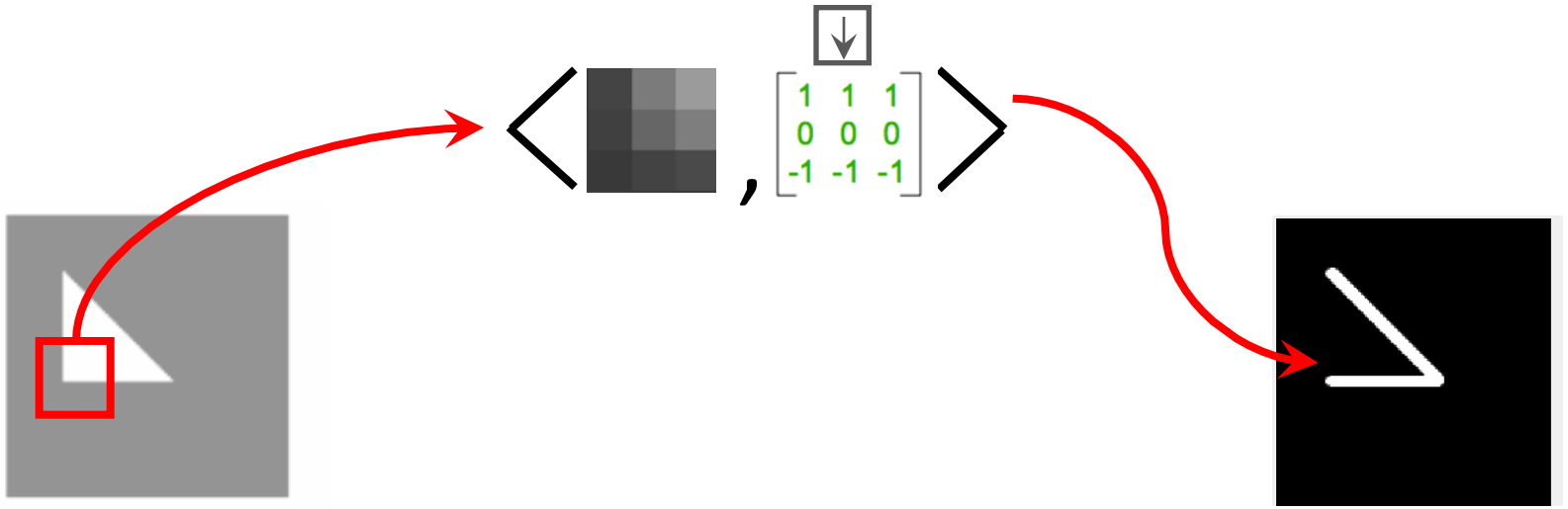
Image taken from [29]

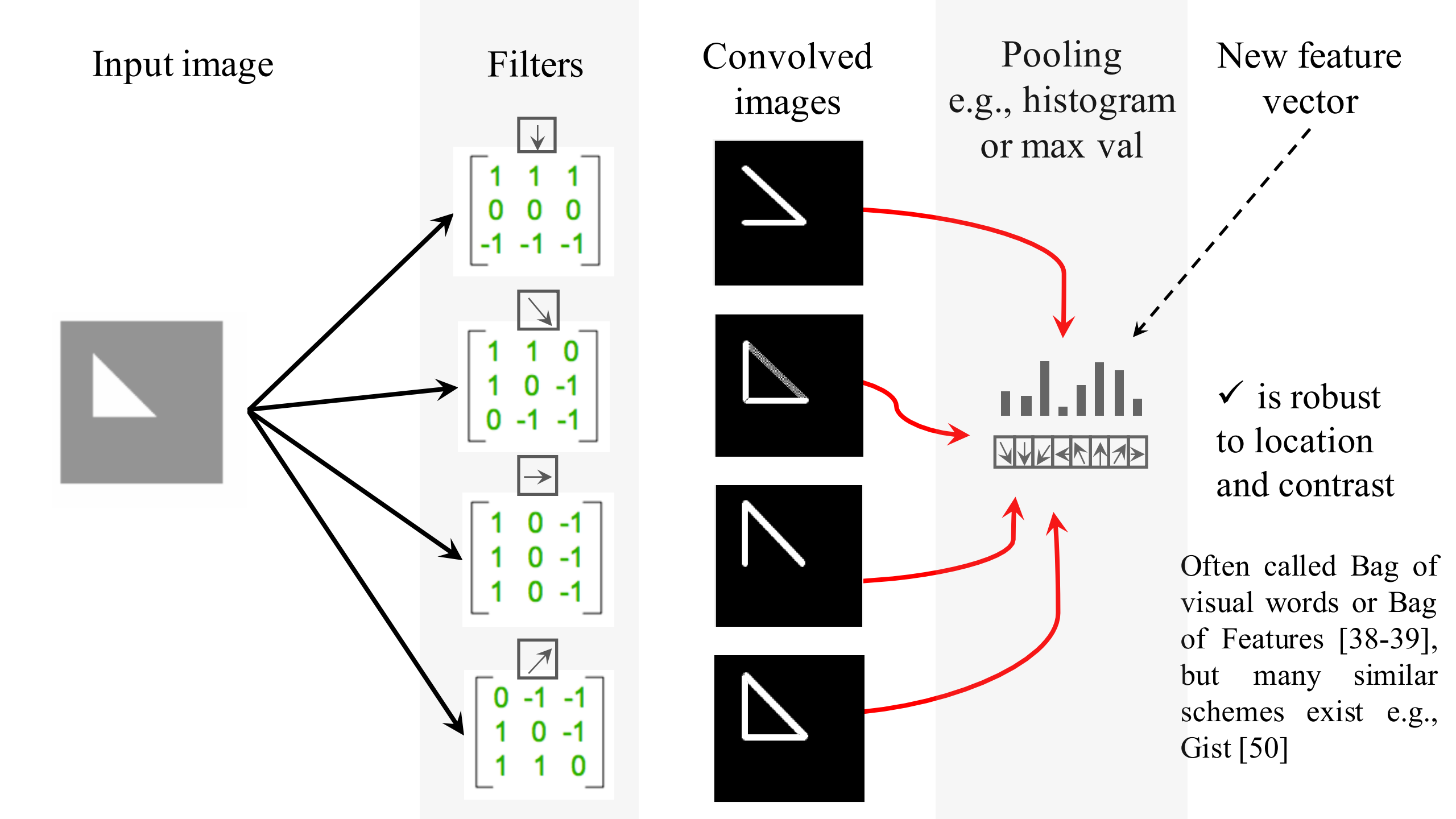
Image histogram features: pixel values or edge orientations?



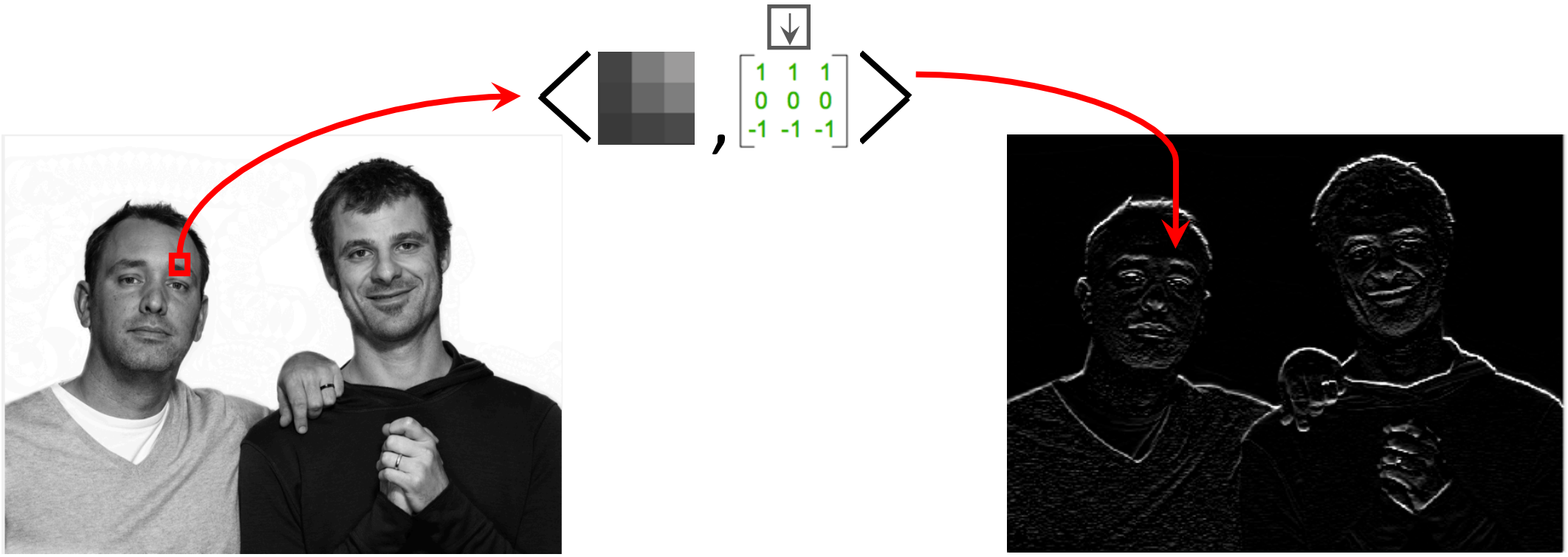


Discrete convolution



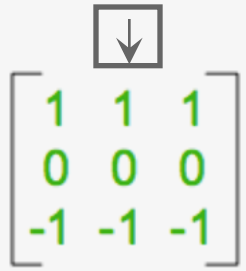

Input image
Filters
Convolved images
Pooling
e.g., histogram or max val
New feature vector
1 1 1
0 0 0
-1 -1 -1
1 1 0
1 0 -1
0 -1 -1
1 0 -1
1 0 -1
1 0 -1
0 -1 -1
1 0 -1
1 1 0
✓ is robust to location and contrast
Often called Bag of visual words or Bag of Features [38-39], but many similar schemes exist e.g., Gist [50]

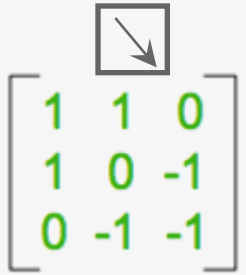
Discrete convoultion

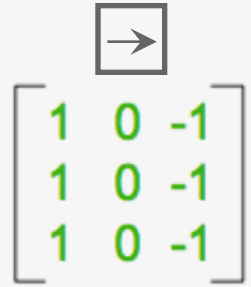


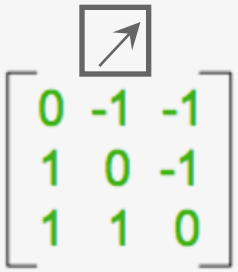
Input image

Filters

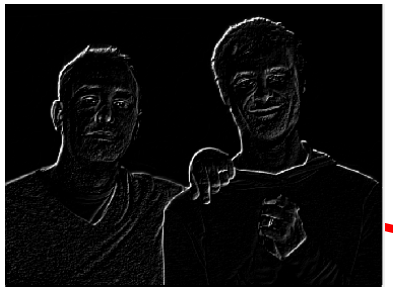








Convolved images



Pooling
e.g., histogram or max val

New feature vector

✓ is robust to contrast but not localization

Often called Bag of visual words or Bag of Features [38-39], but many similar schemes exist e.g., Gist [50]

Input image

Filters

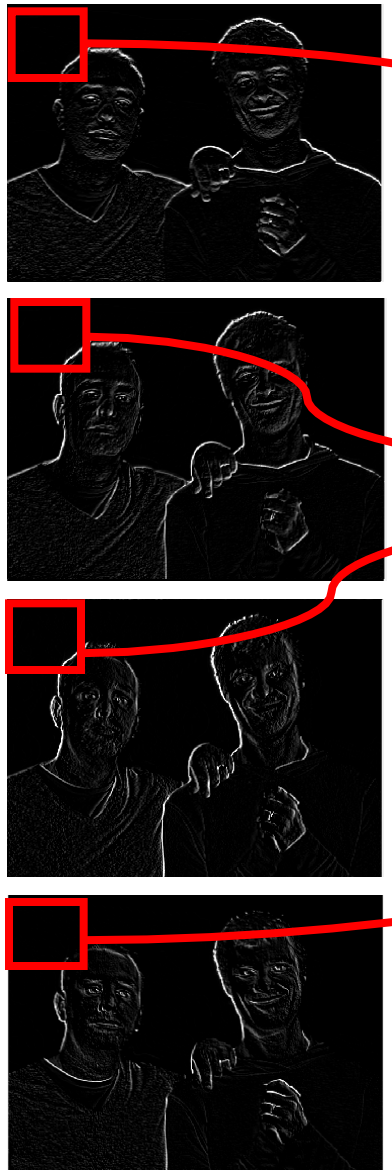
$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

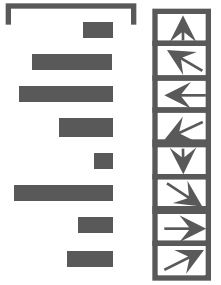
$$\begin{bmatrix} 0 & -1 & -1 \\ 1 & 0 & -1 \\ 1 & 1 & 0 \end{bmatrix}$$

Convolved images



Pooling
e.g., histogram or max val

New feature vector



Input image

Filters

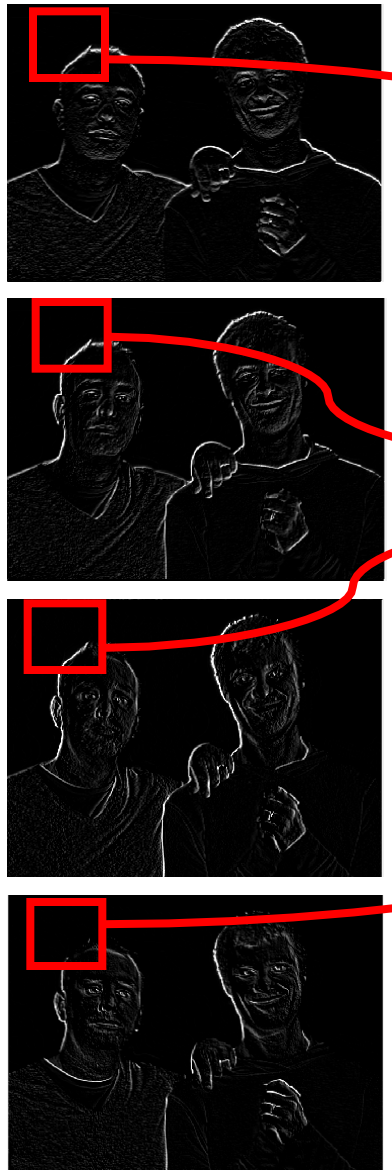
$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

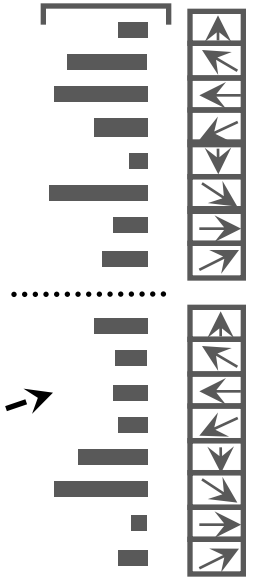
$$\begin{bmatrix} 0 & -1 & -1 \\ 1 & 0 & -1 \\ 1 & 1 & 0 \end{bmatrix}$$

Convolved images



Pooling
e.g., histogram or max val

New feature vector



Input image

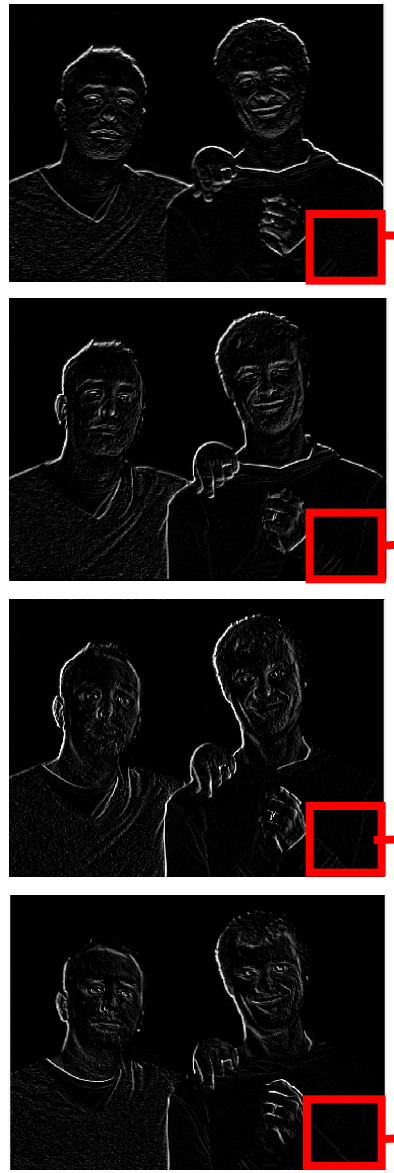
Filters

Convolved images

Pooling e.g., histogram or max val

New feature vector

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \quad \begin{bmatrix} 0 & -1 & -1 \\ 1 & 0 & -1 \\ 1 & 1 & 0 \end{bmatrix}$$



Often called Histogram of Oriented Gradients [22], many similar schemes exist (e.g., SIFT [43]) as well as extensions e.g., spatial pyramid matching [40-41]

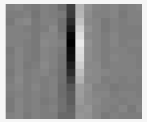
Input image

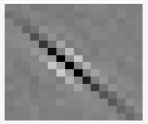
Filters

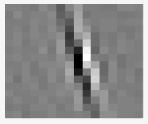
Convolved images

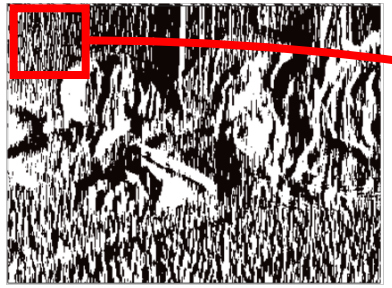
Pooling e.g., histogram or max val

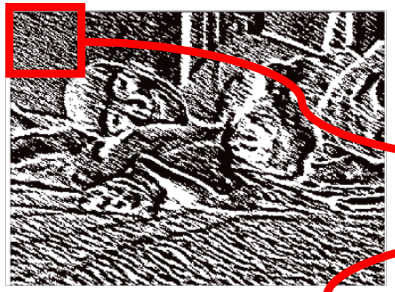
New feature vector

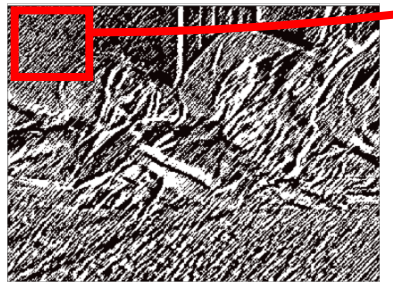


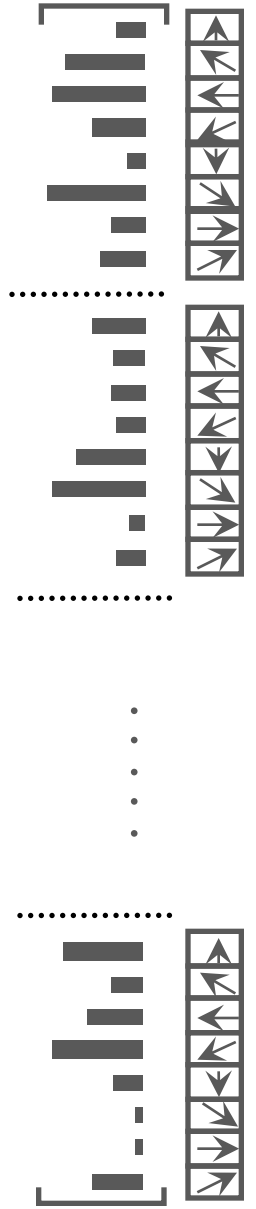












Often called Histogram of Oriented Gradients [22], many similar schemes exist (e.g., SIFT [43]) as well as extensions e.g., spatial pyramid matching [40-41]

Biological inspiration

- Visual cortex consists primarily of simple oriented edge detectors that act on small overlapping patches of visual field

- These detections are then combined (or “pooled”), see e.g., [34 – 36]

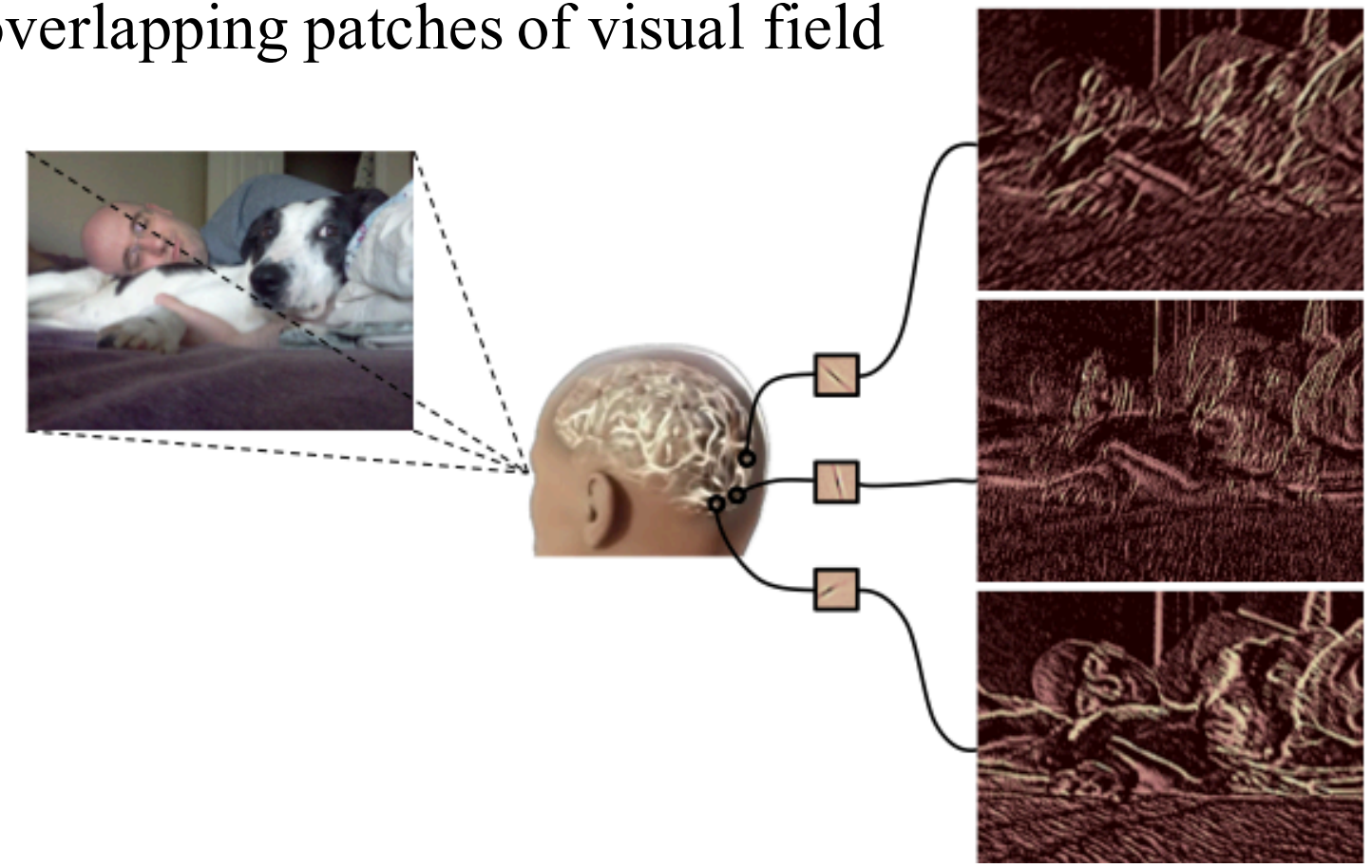
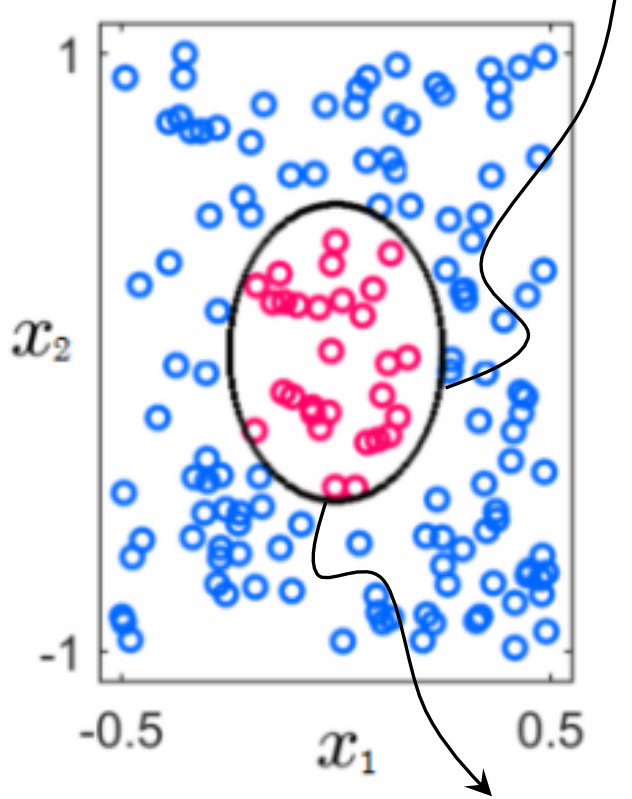


Image taken from [29]

Commonly-used engineered featu

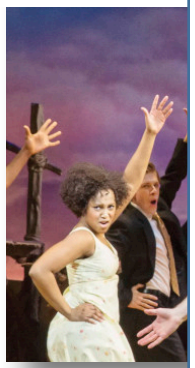
Feature transformations for real dat
aim at ensuring that instances witl
different classes are "dissimilar". T
mations $f_1(\mathbf{x}), ..., f_M(\mathbf{x})$ of the in
algebraically.

$$1 + f_1(\mathbf{x})\, w_1 + f_2(\mathbf{x})\, w_2 = 0$$



$$1 + x_1^2 w_1 + x_2^2 w_2 = 0$$

EVEN THOSE BASED ON REAL PEOPLE--ARE ENTIRELY FICTIONAL. ALL CELEBRITY VOICES ARE IMPERSONATED.....POORLY. THE FOLLOWING PROGRAM CONTAINS COARSE LANGUAGE AND DUE TO ITS CONTENT IT SHOULD NOT BE VIEWED BY ANYONE



Features: summary (re-visited)

What are features?

- **Geometrically:** features are transformations of the input that provoke a good *nonlinear* fit/separation

- **Philosophically:** features are those defining characteristics of the phenomenon underlying a dataset that allow for optimal learning

How do we design good features?

- **By hand (engineered):** translate understanding of a phenomenon into a set of geometric transformations of the input – often requires expertise
  - Requires strong knowledge to produce reasonable results
- **Learn from data (automatic):** use bases (e.g., neural networks) to determine directly from the data
  - Requires large datasets to produce reasonable results (more on this in part II of talk!)

Unsupervised Learning

Principal Component Analysis

- P datapoints $\mathbf{x}_1 \ldots \mathbf{x}_P$
- K basis vectors $\mathbf{c}_1 \ldots \mathbf{c}_K$
- want each point to lie as close to the span of K basis vectors

$$\sum_{k=1}^{K} \mathbf{c}_k w_{k,p} \approx \mathbf{x}_p$$

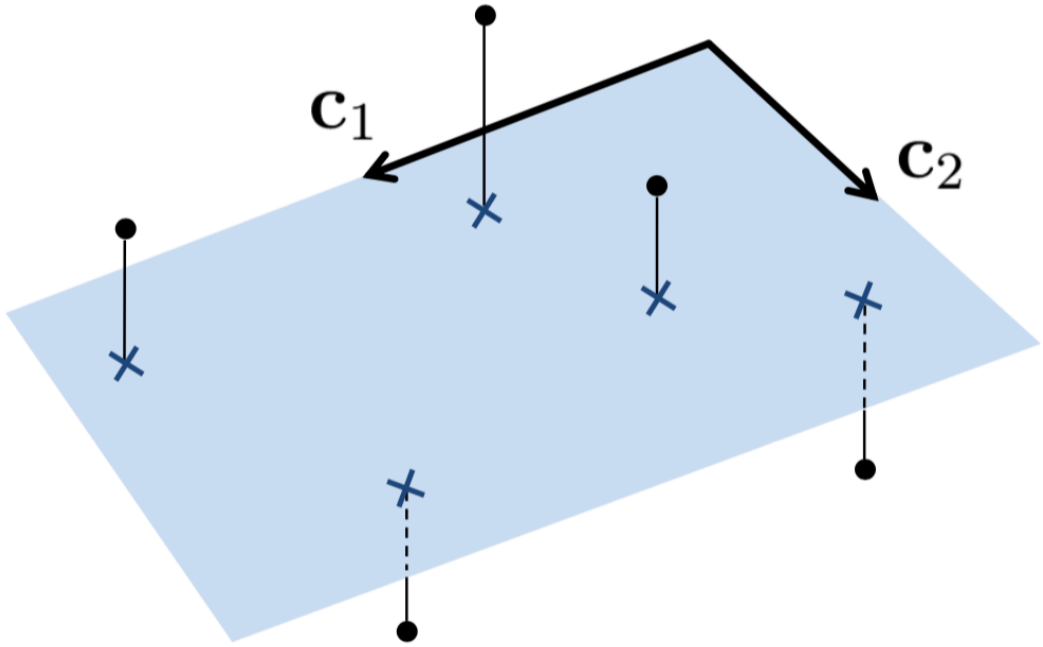


Image taken from [29]

Principal Component Analysis

- P datapoints $\mathbf{x}_1 \ldots \mathbf{x}_P$
- K basis vectors $\mathbf{c}_1 \ldots \mathbf{c}_K$
- want each point to lie as close to the span of K basis vectors

$$\mathbf{C}\mathbf{w}_p \approx \mathbf{x}_p$$

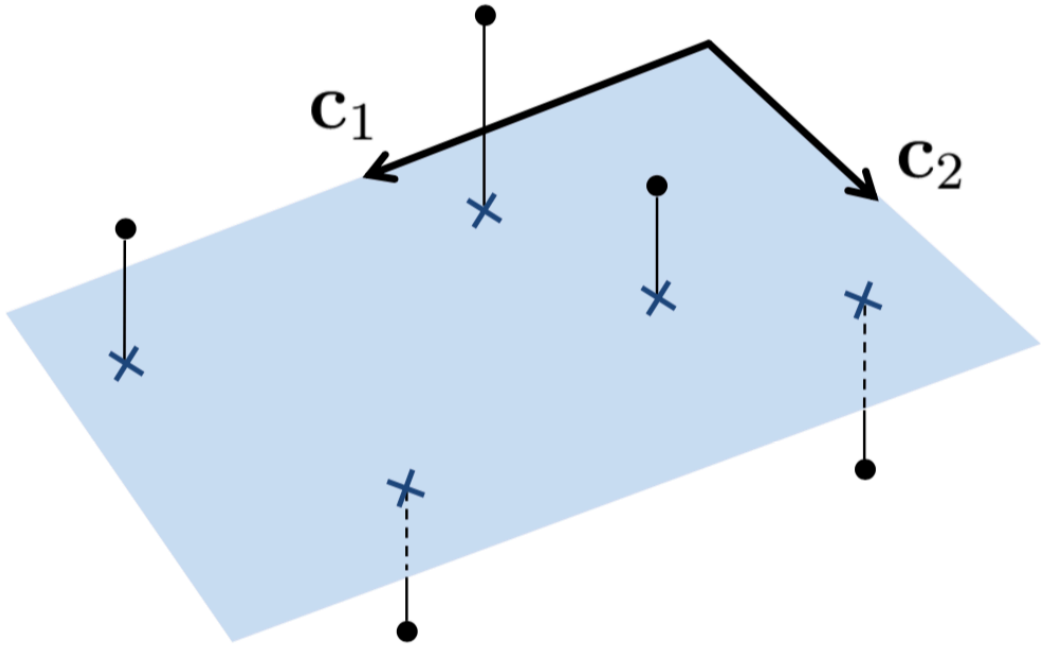
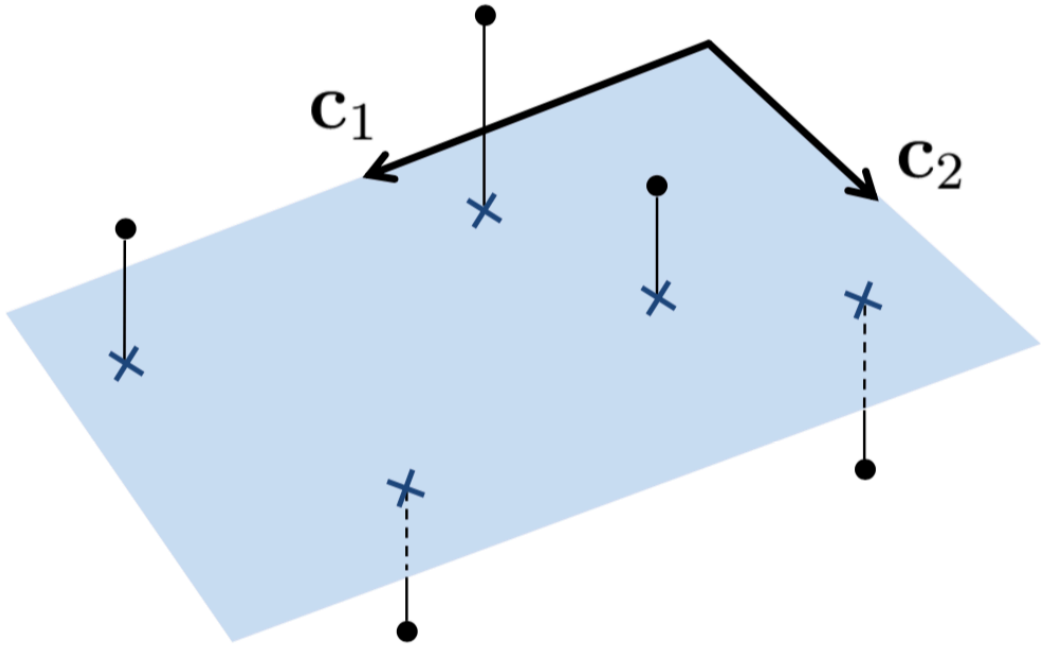


Image taken from [29]

Principal Component Analysis

- P datapoints $\mathbf{x}_1 \ldots \mathbf{x}_P$
- K basis vectors $\mathbf{c}_1 \ldots \mathbf{c}_K$
- want each point to lie as close to the span of K basis vectors



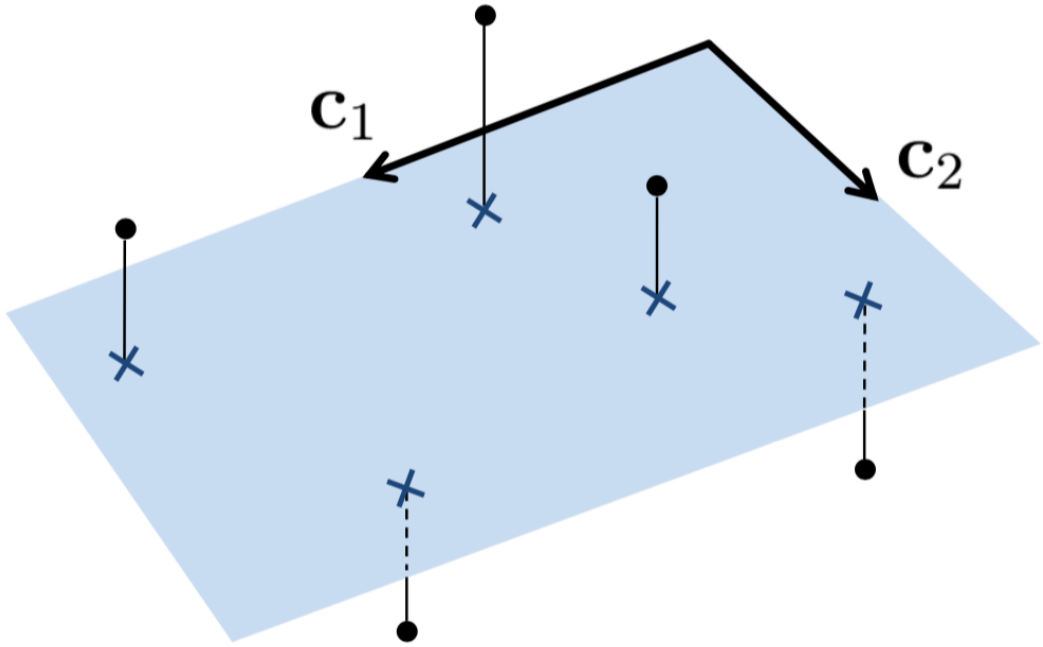
$$\mathbf{C}\mathbf{w}_p \approx \mathbf{x}_p$$

$$\mathbf{C} = [\mathbf{c}_1 | \mathbf{c}_2 | \cdots | \mathbf{c}_K] \qquad \mathbf{w}_p = \begin{bmatrix} w_{1,p} & w_{2,p} & \cdots & w_{K,p} \end{bmatrix}^T$$

Image taken from [29]

Principal Component Analysis

- P datapoints $\mathbf{x}_1 \ldots \mathbf{x}_P$
- K basis vectors $\mathbf{c}_1 \ldots \mathbf{c}_K$
- want each point to lie as close to the span of K basis vectors



$$\mathbf{CW} \approx \mathbf{X}$$

$$\mathbf{W} = [\mathbf{w}_1 | \mathbf{w}_2 | \cdots | \mathbf{w}_P] \qquad \mathbf{X} = [\mathbf{x}_1 | \mathbf{x}_2 | \cdots | \mathbf{x}_P]$$

Image taken from [29]

Principal Component Analysis

- P datapoints $\mathbf{x}_1 \ldots \mathbf{x}_P$
- K basis vectors $\mathbf{c}_1 \ldots \mathbf{c}_K$
- want each point to lie as close to the span of K basis vectors

$$\|\mathbf{CW} - \mathbf{X}\|_F^2$$

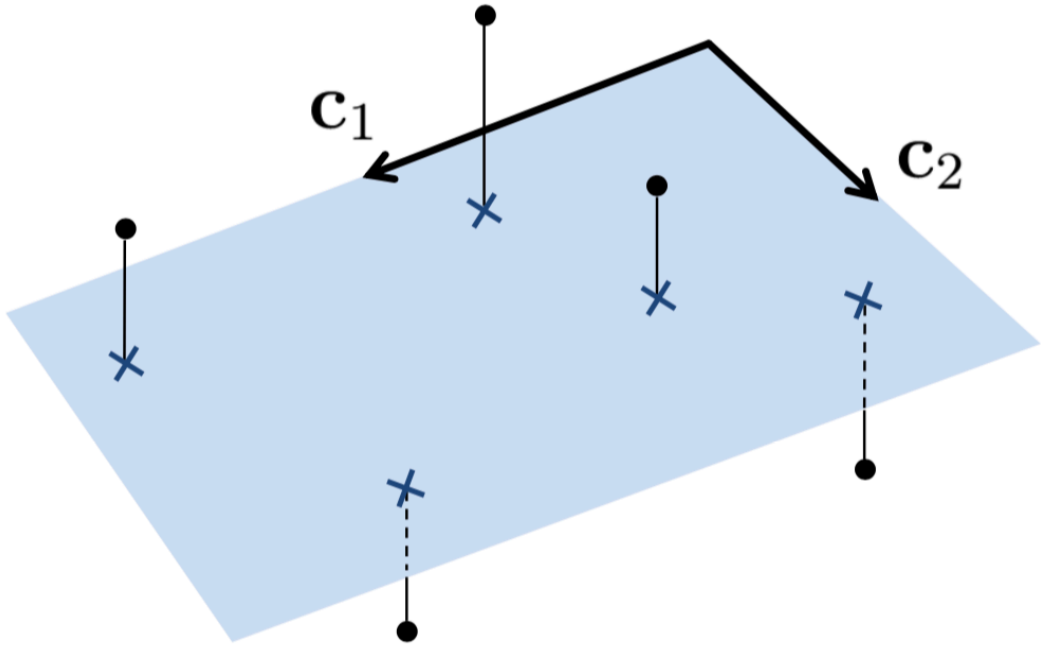


Image taken from [29]

Principal Component Analysis

- P datapoints $\mathbf{x}_1 \ldots \mathbf{x}_P$
- K basis vectors $\mathbf{c}_1 \ldots \mathbf{c}_K$
- want each point to lie as close to the span of K basis vectors

$$\underset{\mathbf{C},\mathbf{W}}{\text{minimize}} \|\mathbf{CW} - \mathbf{X}\|_F^2$$

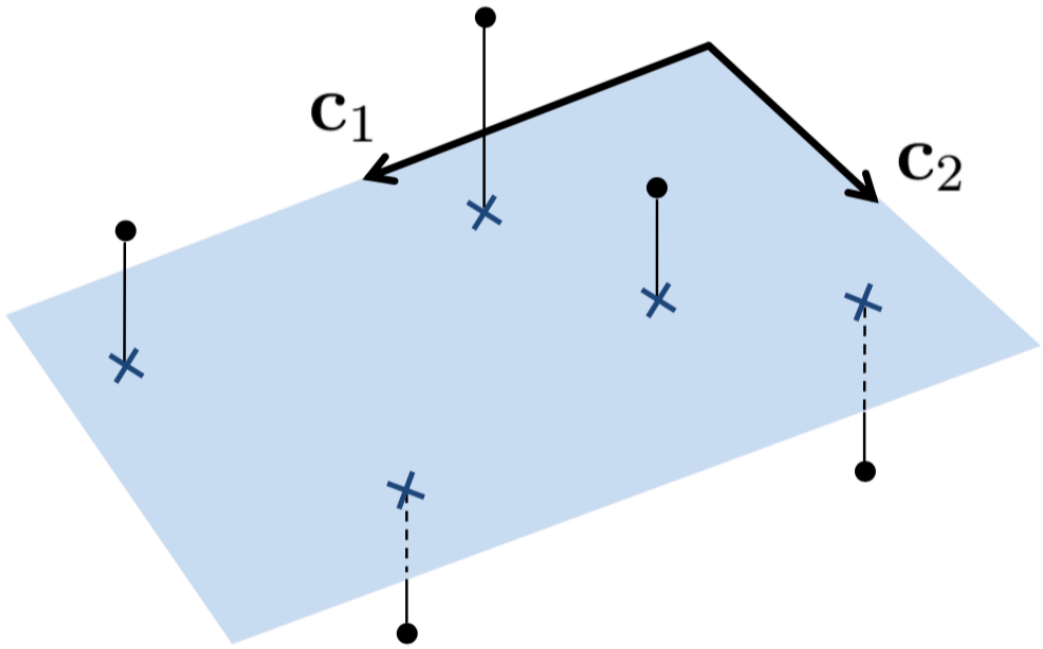


Image taken from [29]

K-means clustering

- P datapoints $\mathbf{x}_1 \ldots \mathbf{x}_P$
- K basis ~~vectors~~ centroids $\mathbf{c}_1 \ldots \mathbf{c}_K$
- want each point to lie as close to its corresponding centroid

$$\mathbf{CW} \approx \mathbf{X}$$

$$\mathbf{w}_p \in \{\mathbf{e}_k\}_{k=1}^{K} \quad \forall p = 1 \ldots P$$

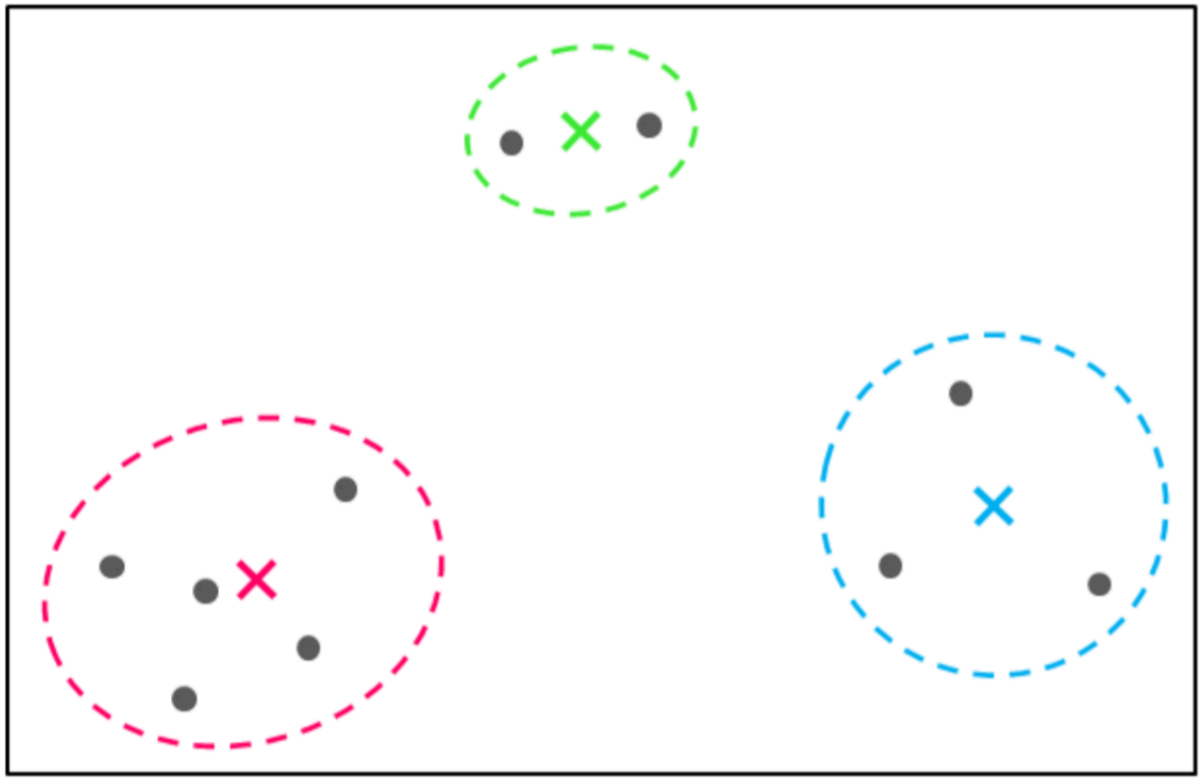
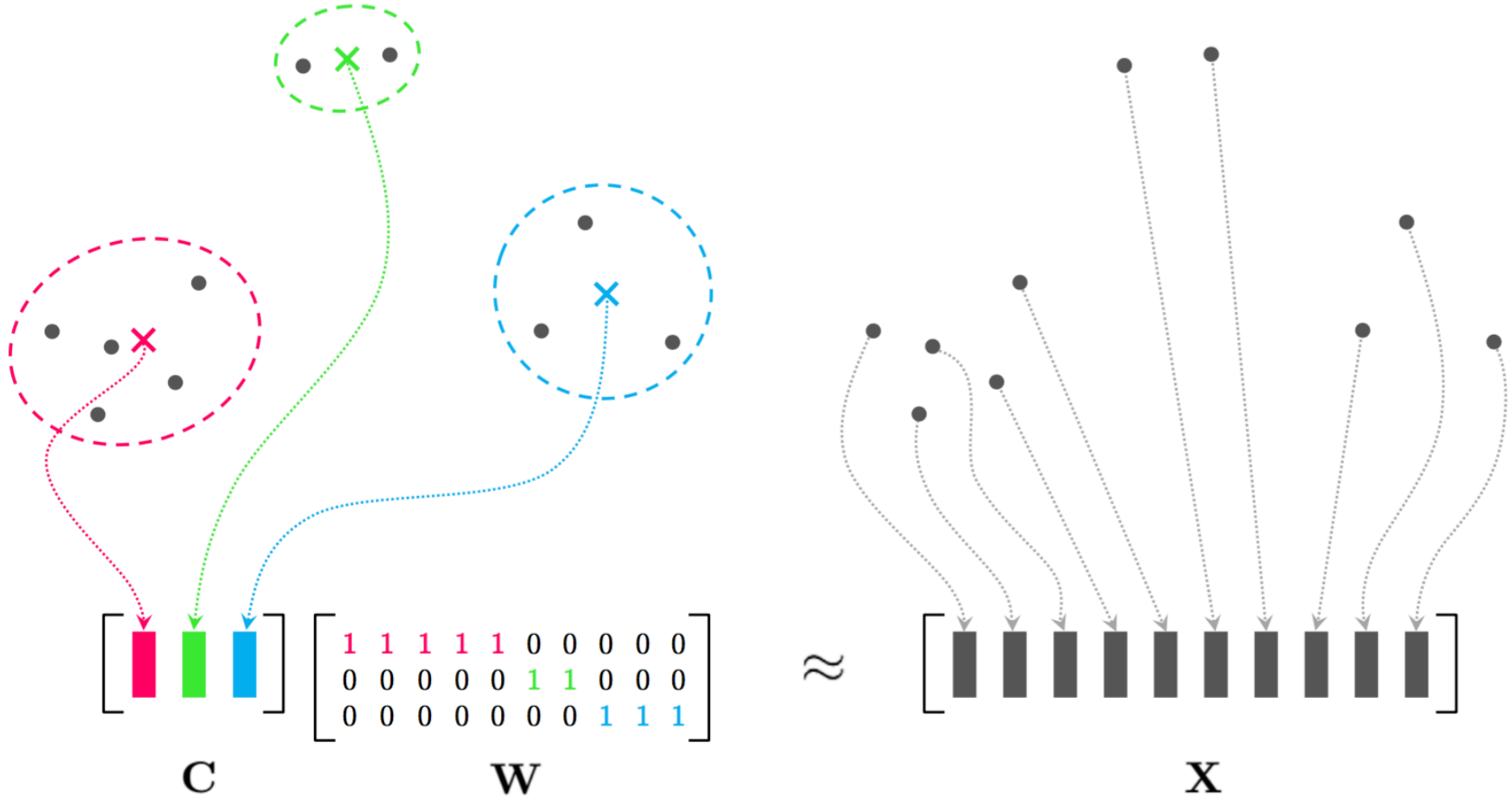


Image taken from [29]



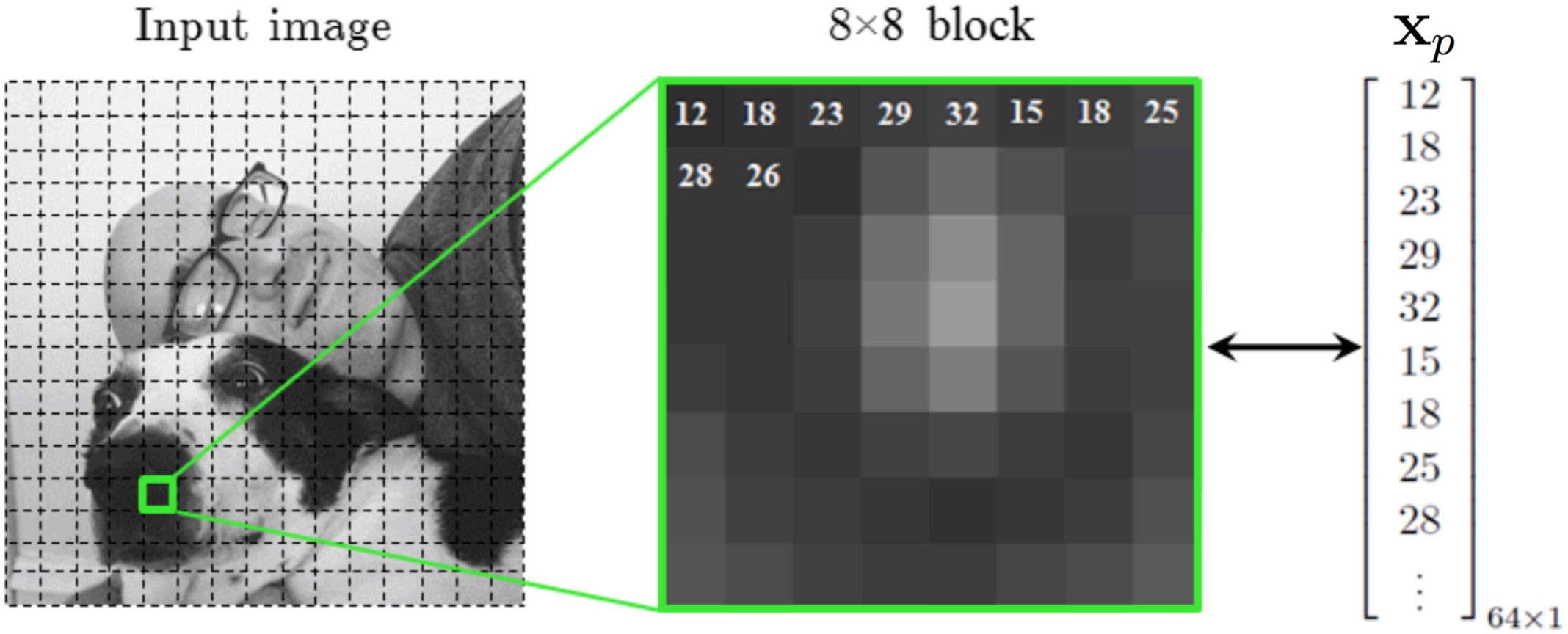
$$\|\mathbf{CW} - \mathbf{X}\|_F^2$$

Image taken from [29]

Common variations on the matrix factorization problem $\mathbf{CW} \approx \mathbf{X}$

Problem	Constraints
PCA/SVD	None
K-means	$\mathbf{w}_i$ a standard basis vector for all i

Classic signal processing

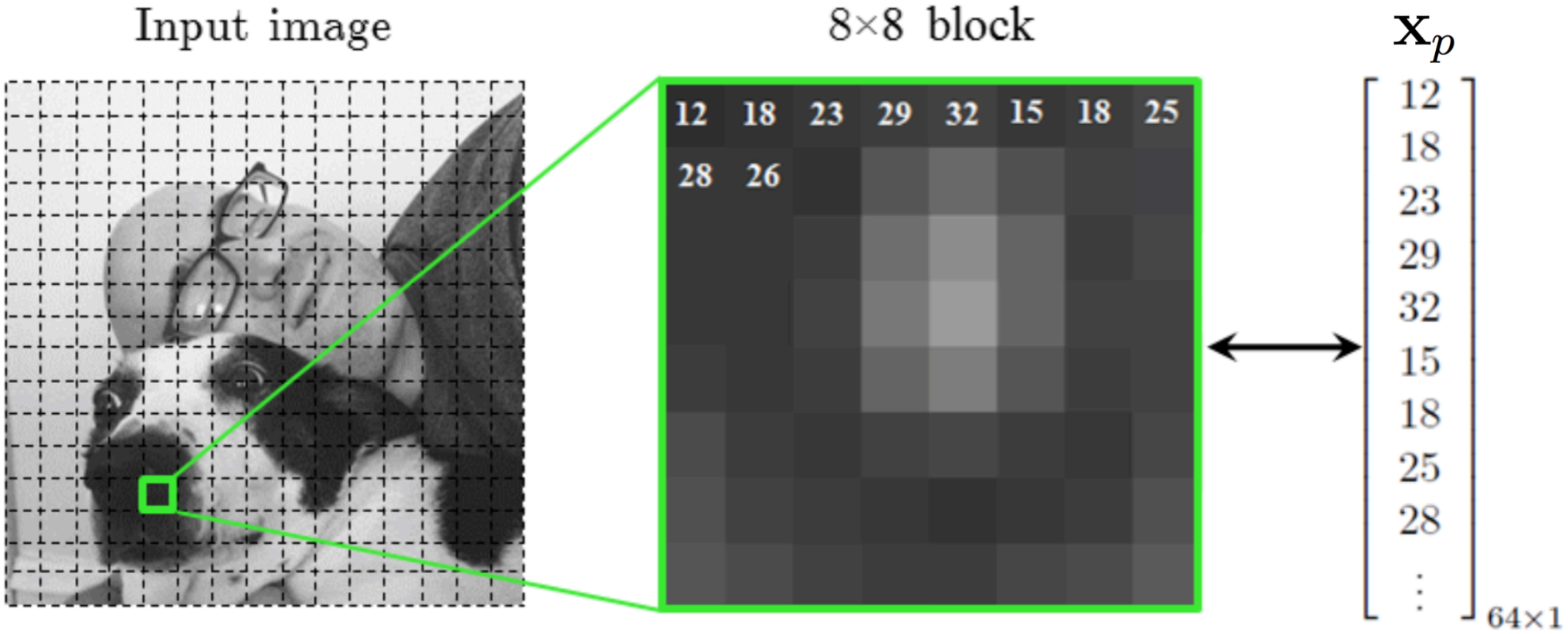


Very good bases $\mathbf{C}$ exist so that using only a few basis vectors

$$\mathbf{C}\mathbf{w}_p \approx \mathbf{x}_p$$

Image taken from [29]

Classic signal processing

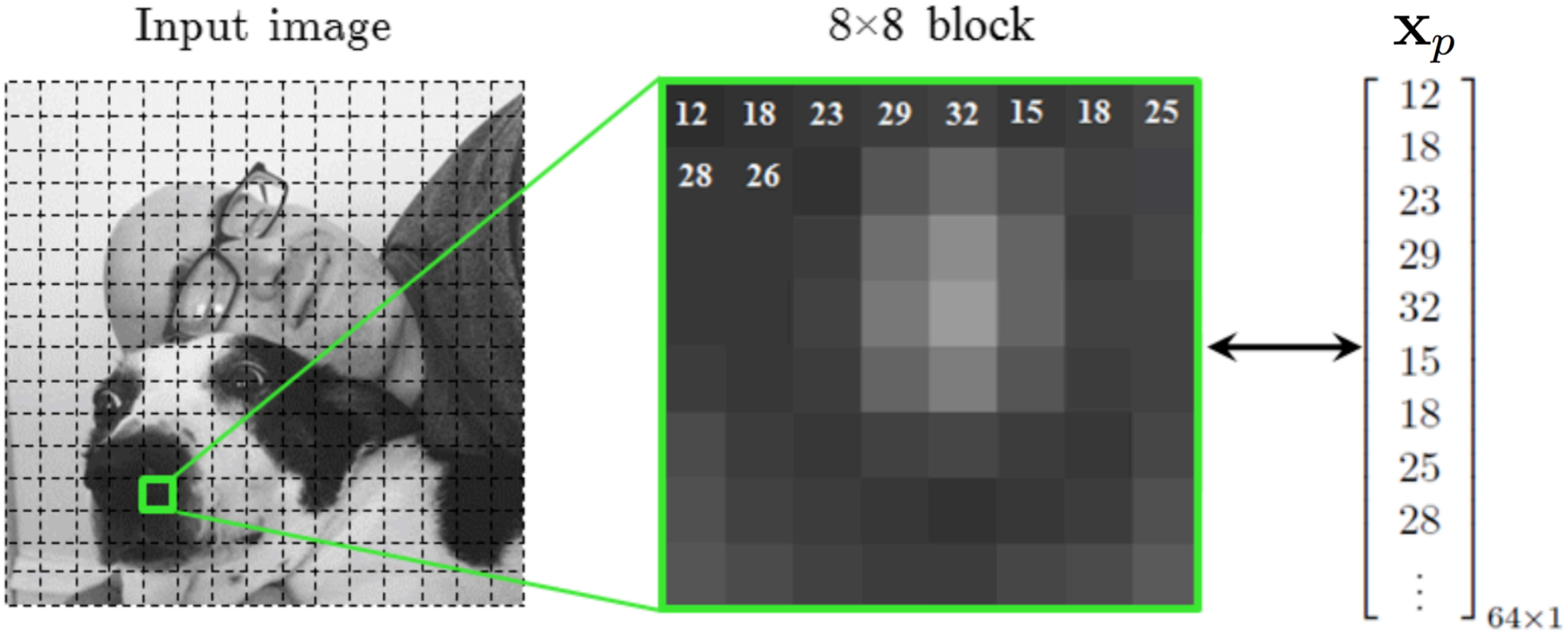


Very good bases $\mathbf{C}$ exist so that using only a few basis vectors

$\mathbf{C}\mathbf{w}_p \approx \mathbf{x}_p$ $\mathbf{w}_p$ has few non-zero entries

Image taken from [29]

Classic signal processing

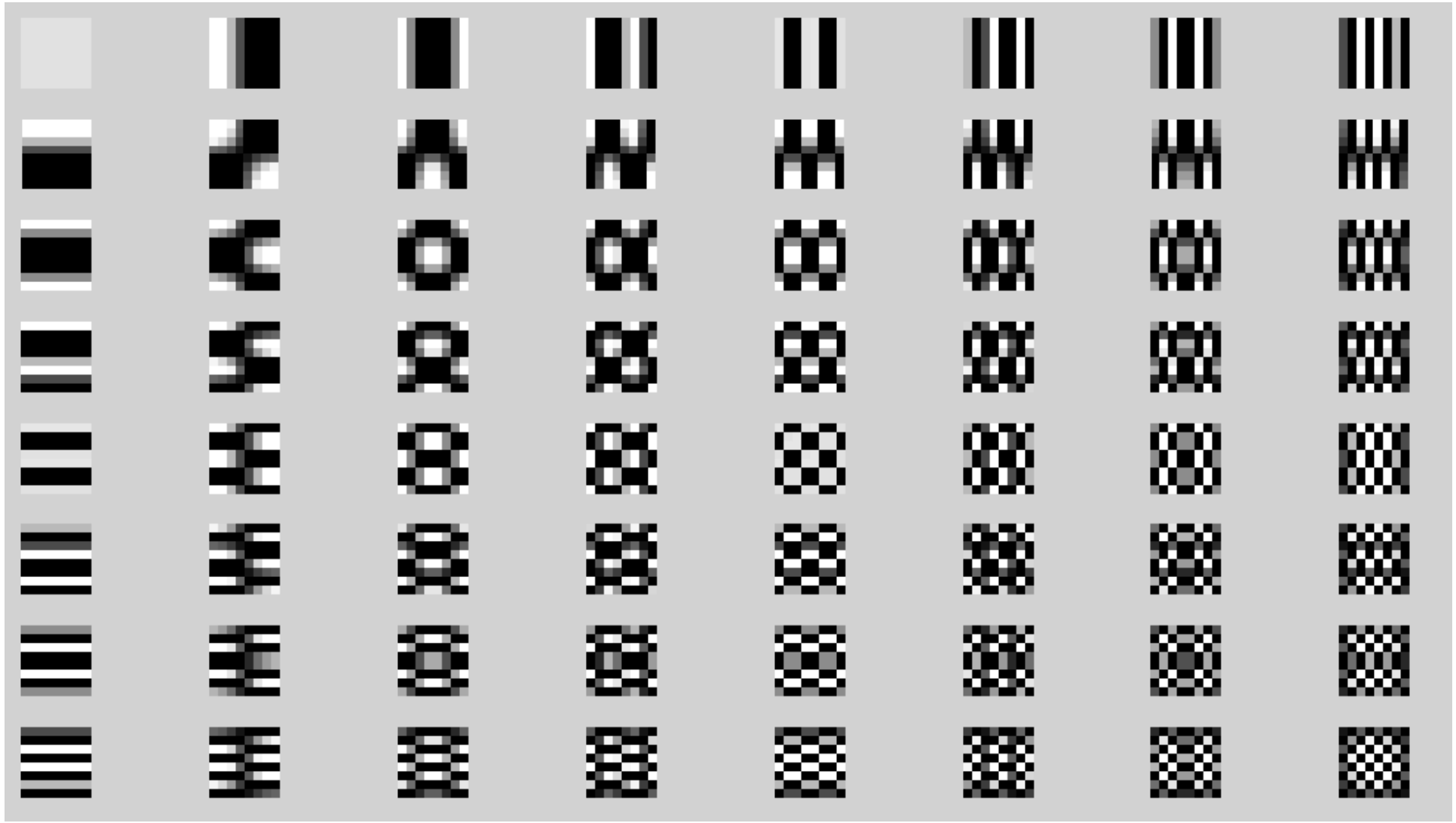


Very good bases $\mathbf{C}$ exist so that using only a few basis vectors

$$\mathbf{C}\mathbf{w}_p \approx \mathbf{x}_p \qquad \|\mathbf{w}_p\|_0 \text{ is small}$$

Image taken from [29]

The DCT basis



Compression using DCT

Original

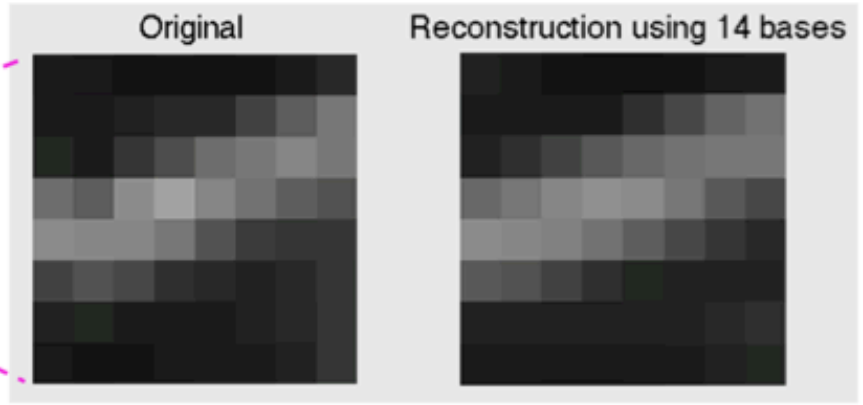
20%

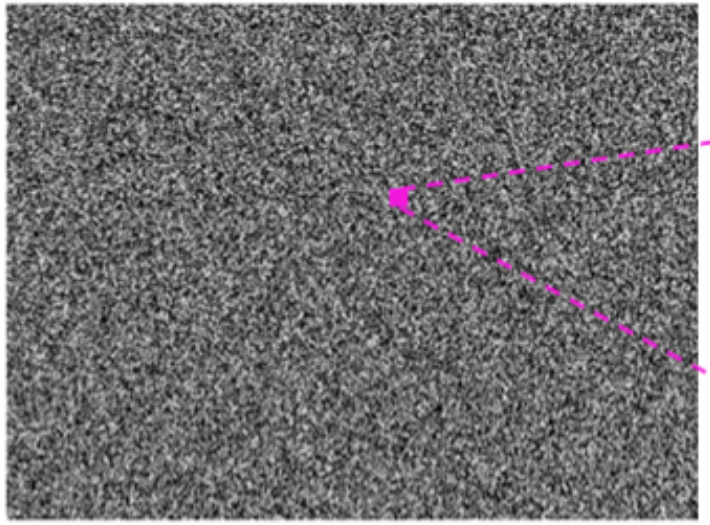
5%

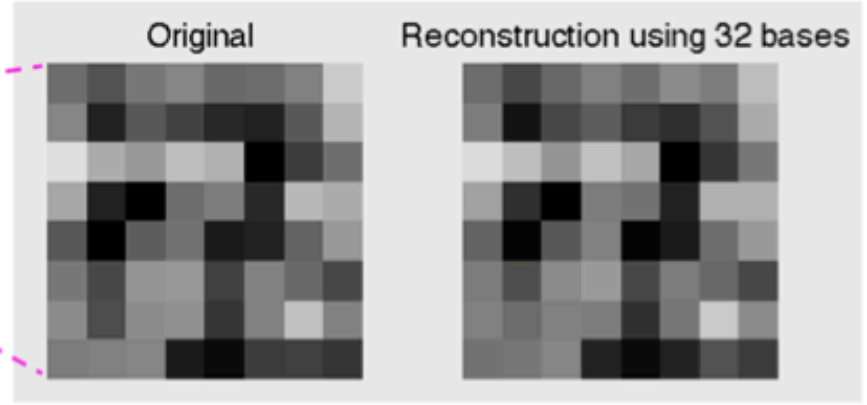
1%

Image taken from [29]

Single patch example







Classic Neuroscience

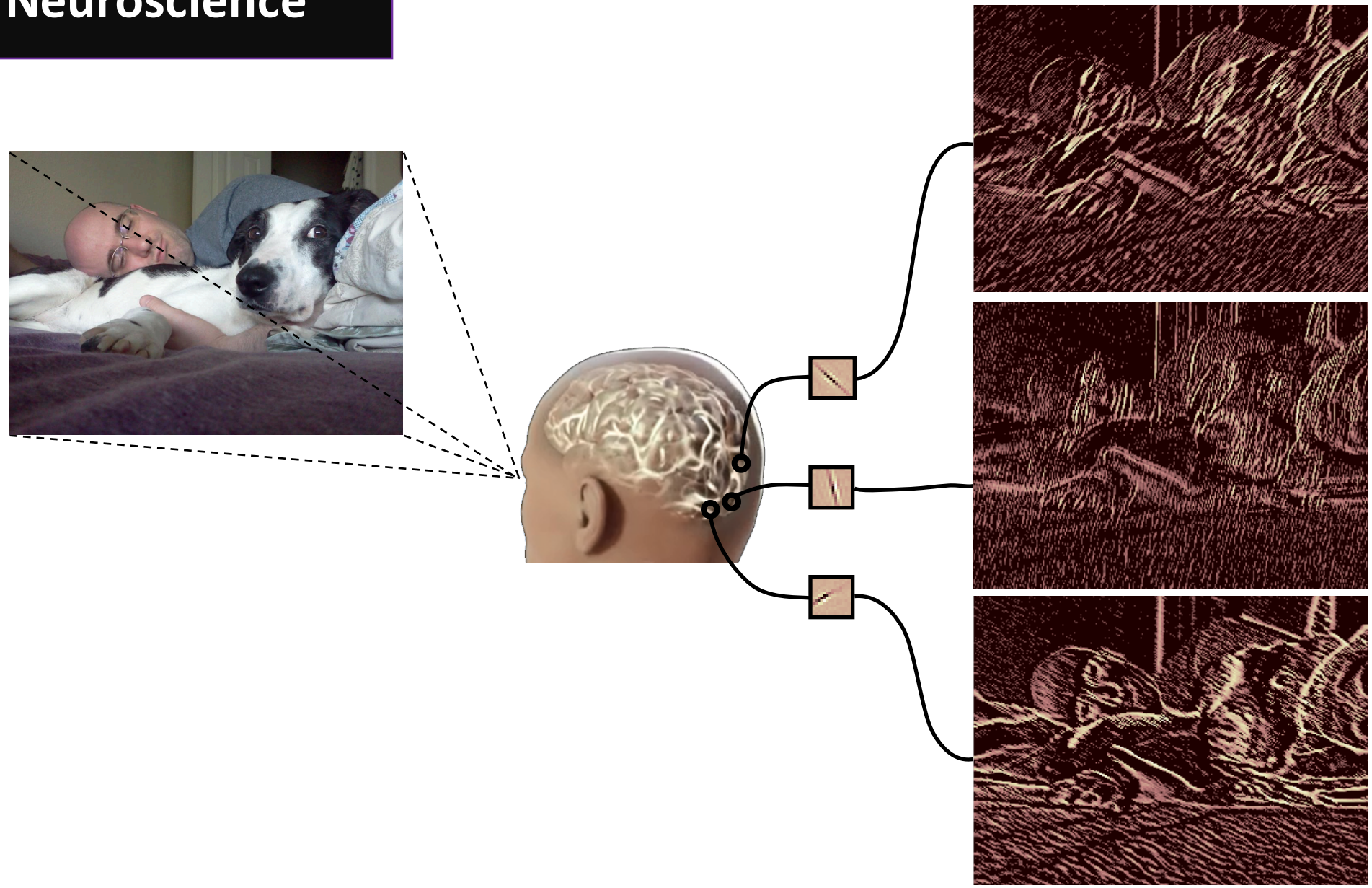


Image taken from [29]

Neuroscience: Sparse modeling

Neuroscience

See e.g., [52]

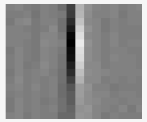
Input image

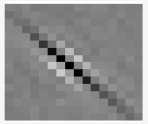
Filters

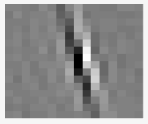
Convolved images

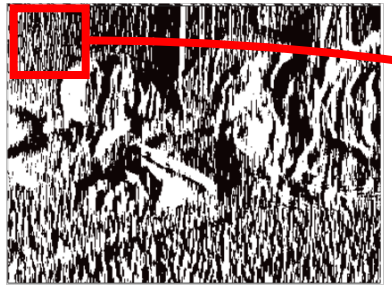
Pooling e.g., histogram or max val

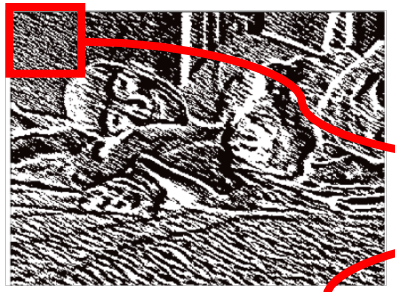
New feature vector

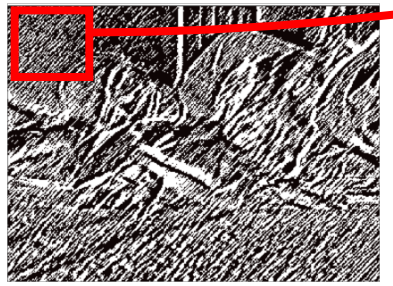


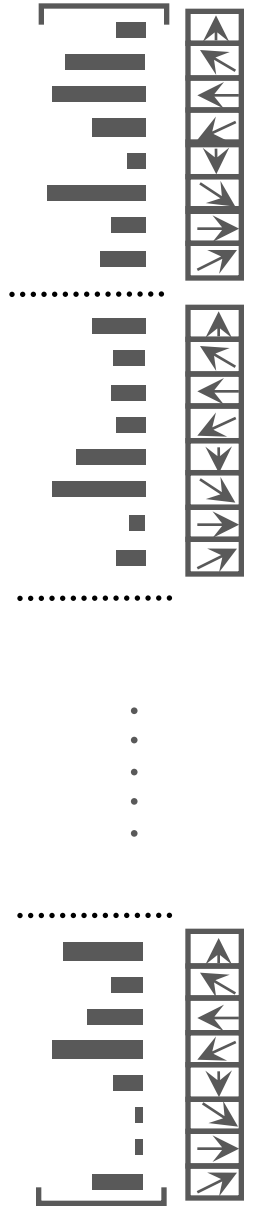




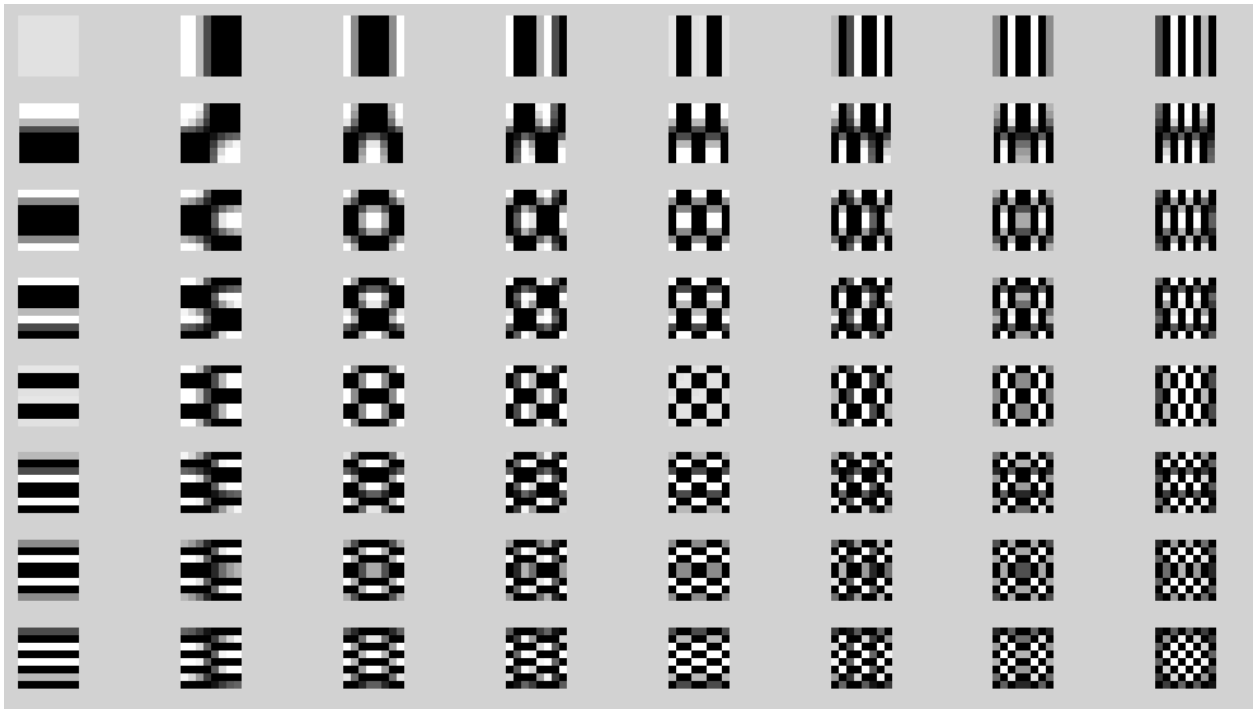


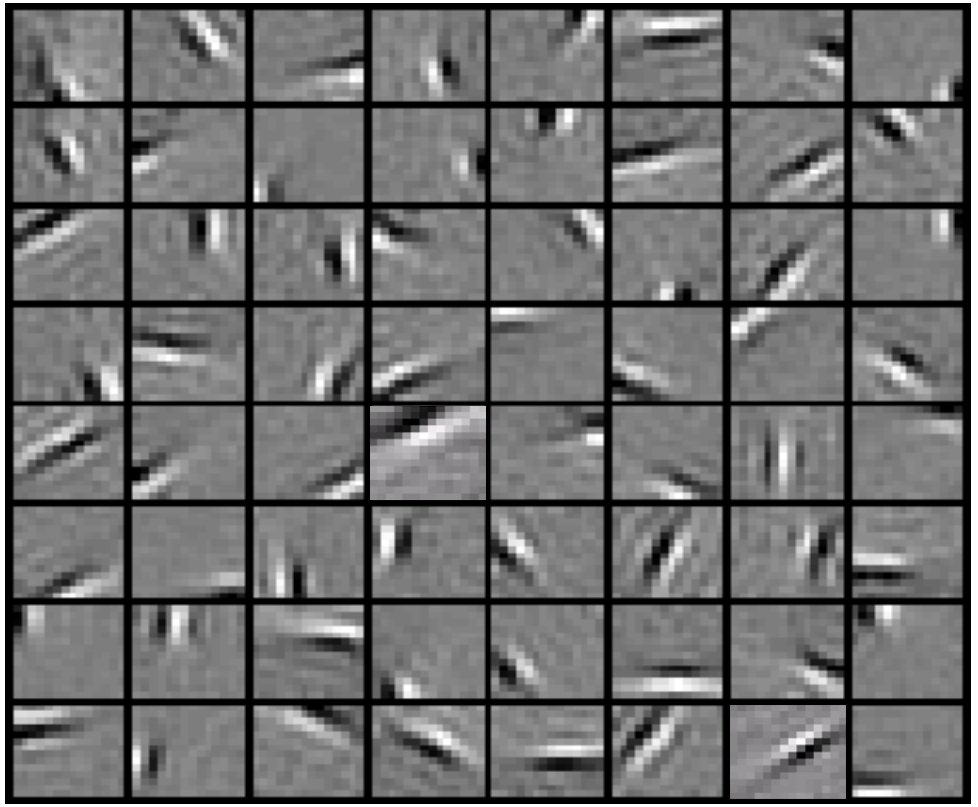






Often called Histogram of Oriented Gradients [22], many similar schemes exist (e.g., SIFT [43]) as well as extensions e.g., spatial pyramid matching [40-41]





Very good bases $\mathbf{C}$ exist so that using only a few basis vectors

$$\mathbf{C}\mathbf{w}_p \approx \mathbf{x}_p$$

$\|\mathbf{w}_p\|_0$ is small

Very good bases $\mathbf{C}$ exist so that using only a few basis vectors

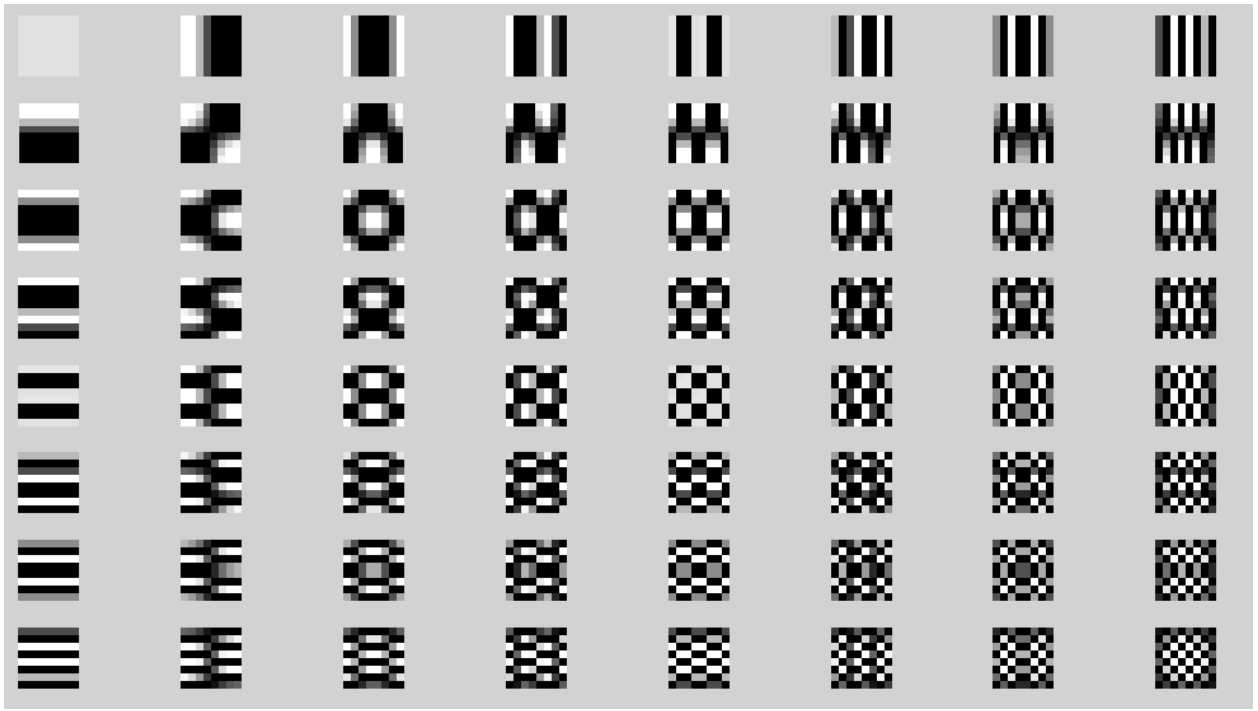
$$\mathbf{C}\mathbf{w}_p \approx \mathbf{x}_p$$

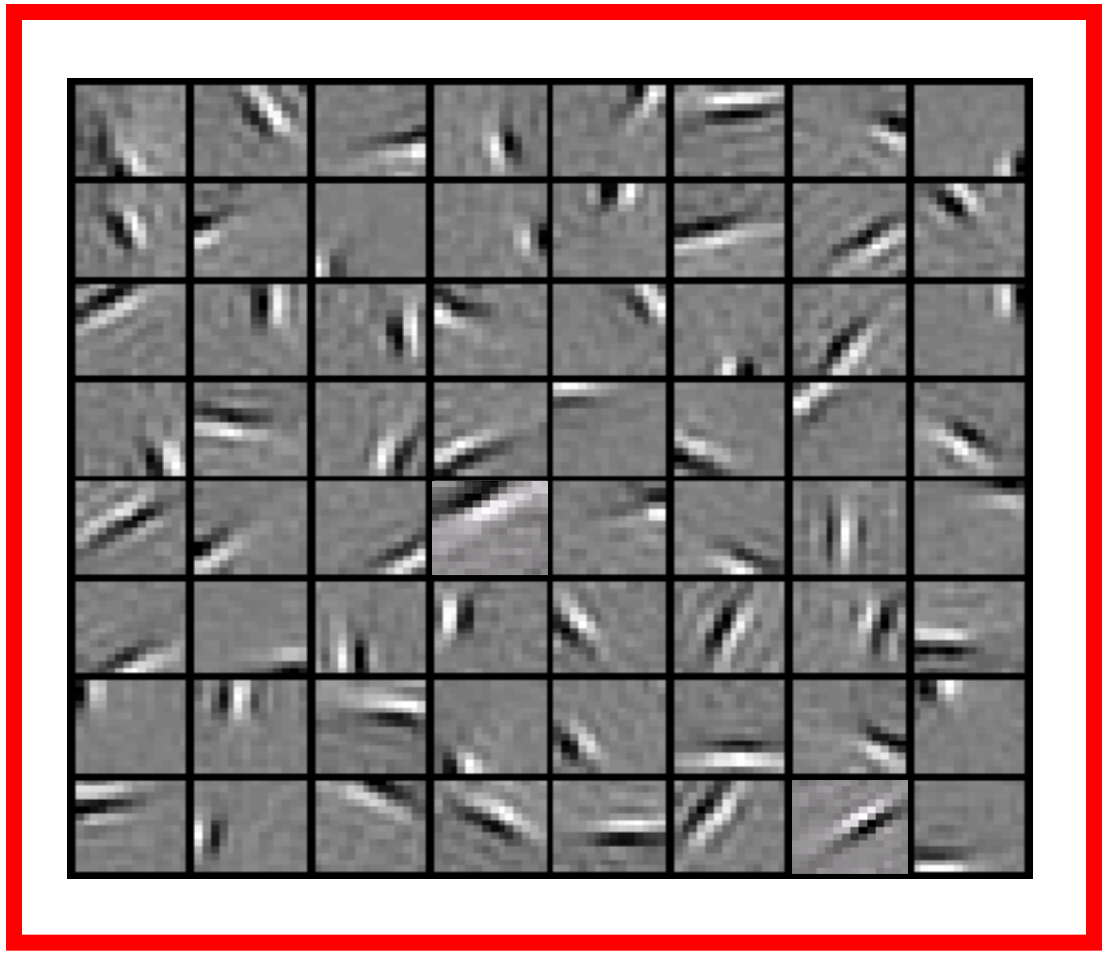
$\|\mathbf{w}_p\|_0$ is small

$$\mathbf{CW} \approx \mathbf{X}$$

$\|\mathbf{w}_p\|_0$ is small

$$\underset{\mathbf{C},\mathbf{W}}{\text{minimize}} \quad \|\mathbf{CW} - \mathbf{X}\|_F^2$$
$$\text{subject to} \quad \|\mathbf{w}_p\|_0 \leq S$$





$$\underset{\mathbf{C},\mathbf{W}}{\text{minimize}} \ \|\mathbf{CW} - \mathbf{X}\|_F^2$$
$$\text{subject to} \ \|\mathbf{w}_p\|_0 \leq S$$

Common variations on the matrix factorization problem $\mathbf{CW} \approx \mathbf{X}$

Problem	Constraints
PCA/SVD	None
K-means	$\mathbf{w}_i$ a standard basis vector for all i

End of Part I