

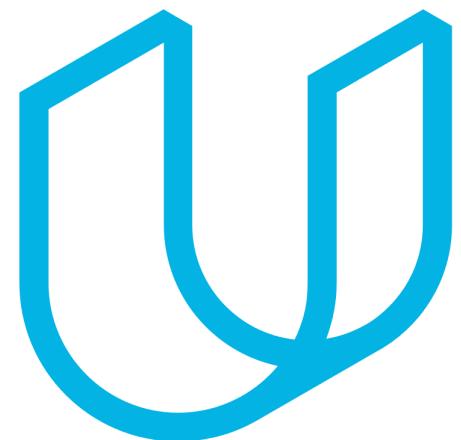
Computer Vision in Python: Object Detection Systems



Jeremy Watt
jermwatt@gmail.com



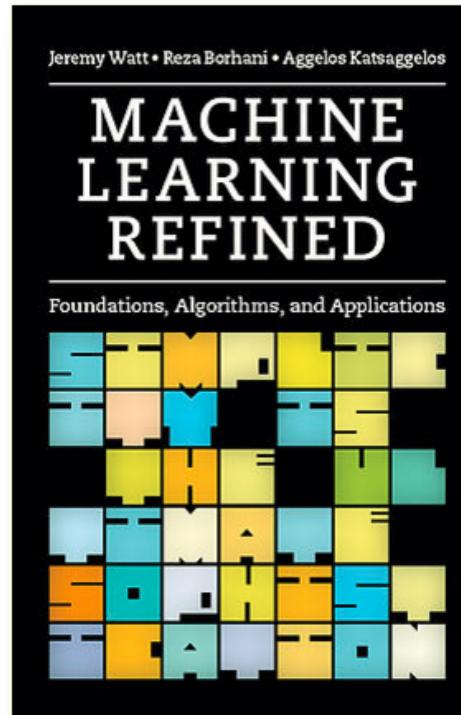
Northwestern
University



UDACITY



What is this talk based on?



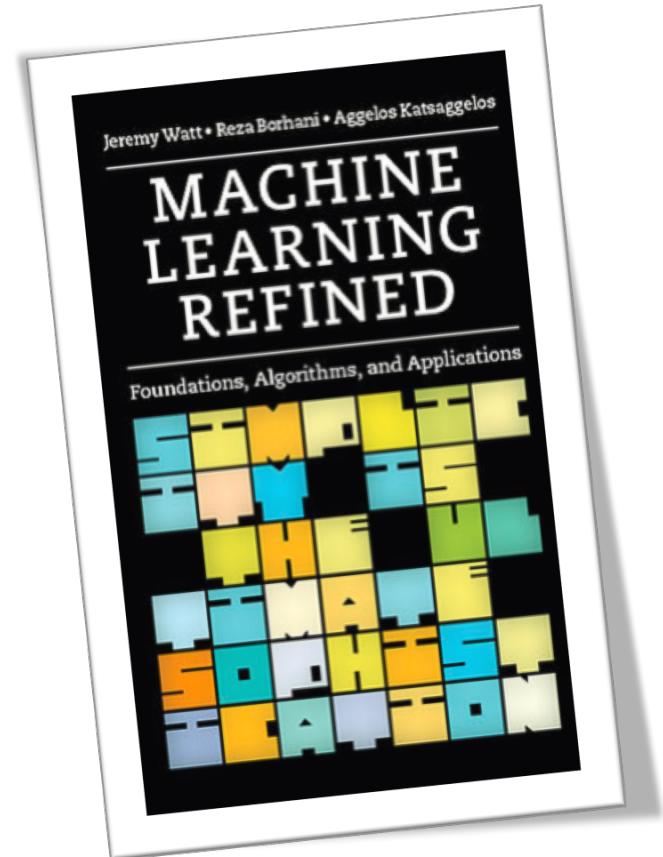
A new ML textbook!
Cambridge University Press
October 2016



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

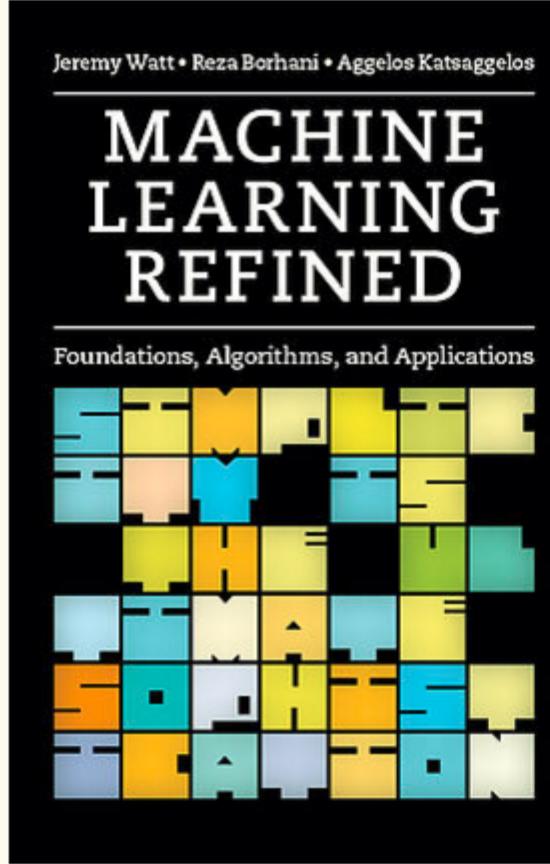
What makes our book unique?

- 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)



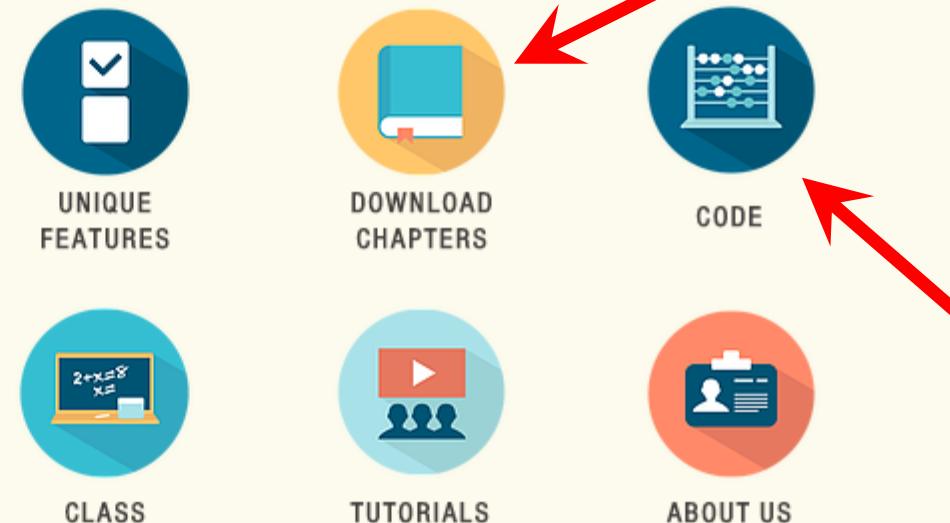
Where to download the slides?

www.MLrefined.com



Machine Learning Refined (to appear early 2016 with Cambridge University Press) is a new machine learning textbook containing fresh and intuitive yet rigorous descriptions of the most fundamental concepts necessary to conduct research, build products, tinker, and play. The text includes a plethora of practical examples and exercises, written in both Python and Matlab/Octave programming languages, to help readers gain complete mastery of the subject.

To learn more about what makes our textbook so great [click here](#), and then see for yourself by downloading free sample chapters via the link below!



Sample chapters
of the book

Slides

What is this talk about?

- Goal: give you an overview of computer vision - object detection – (in Python)
- So, we'll talk about
 - What are the basic building blocks of an object detection system?
 - How do they (basically) work?
 - Where do you find the tools?

Classification tasks

Regression

Predicting a *continuous-valued* variable

Classification

Predicting a *discrete-valued* variable
i.e., distinguishing between different
classes of data

Regression

Predicting a *continuous-valued* variable

Ex. Financial modeling

Brent crude oil prices, January 2014 - January 2016



Source: Bloomberg

BBC

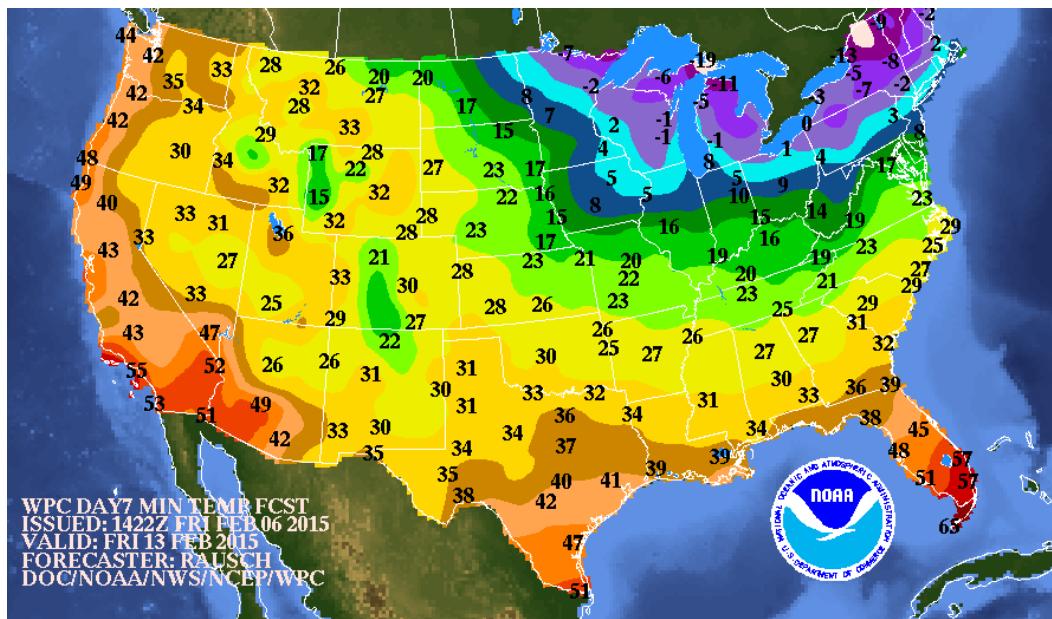
Classification

Predicting a *discrete-valued* variable
i.e., distinguishing between different
classes of data

Regression

Predicting a *continuous-valued* variable

Ex. Weather forecasting



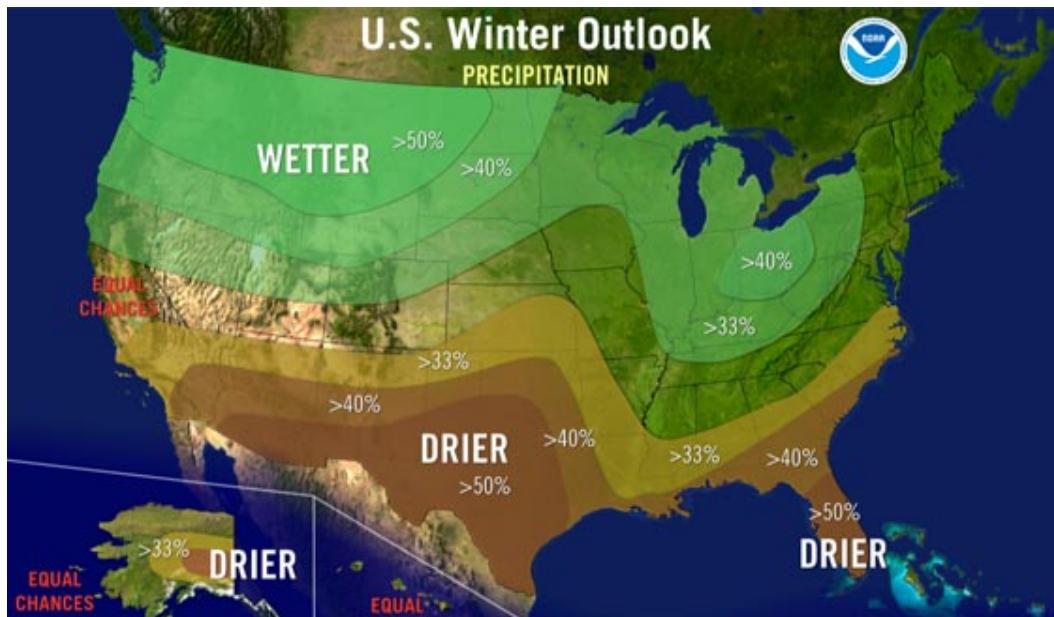
Classification

Predicting a *discrete-valued* variable
i.e., distinguishing between different classes of data

Regression

Predicting a *continuous-valued* variable

Ex. Weather forecasting



Classification

Predicting a *discrete-valued* variable
i.e., distinguishing between different
classes of data

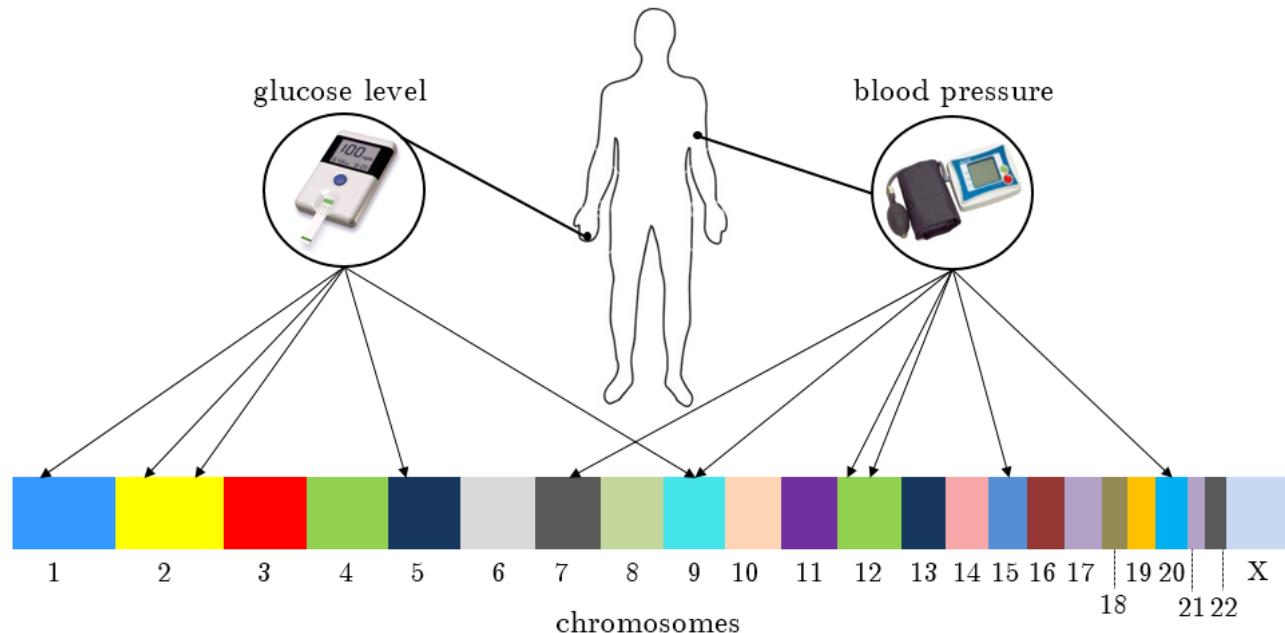
Regression

Predicting a *continuous-valued* variable

Classification

Predicting a *discrete-valued* variable
i.e., distinguishing between different
classes of data

Ex. Genetics



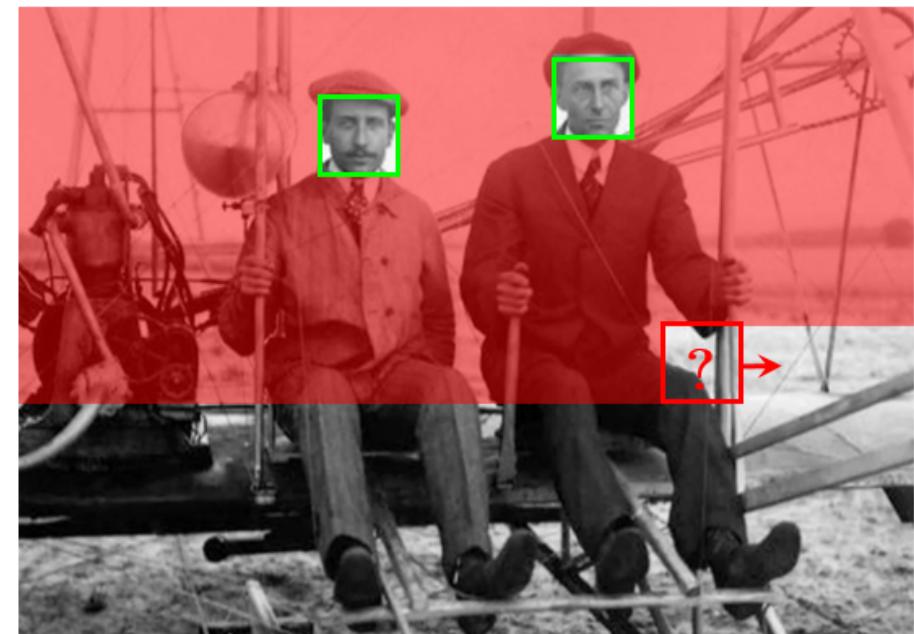
Regression

Predicting a *continuous-valued* variable

Classification

Predicting a *discrete-valued* variable
i.e., distinguishing between different
classes of data

Ex. Face detection



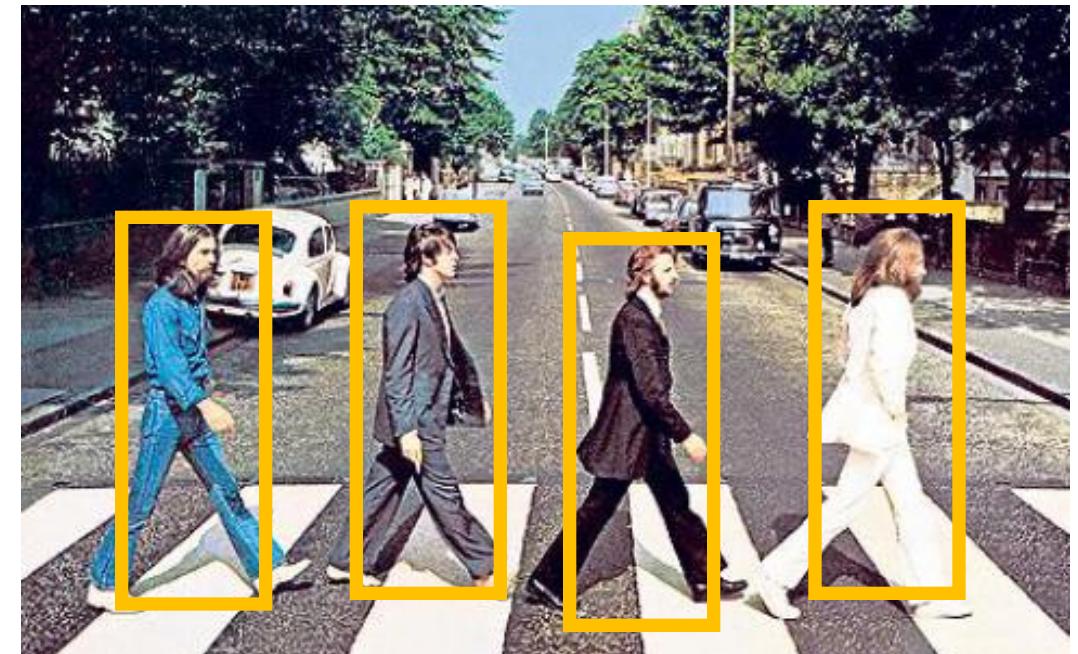
Regression

Predicting a *continuous-valued* variable

Classification

Predicting a *discrete-valued* variable
i.e., distinguishing between different
classes of data

Ex. Pedestrian detection



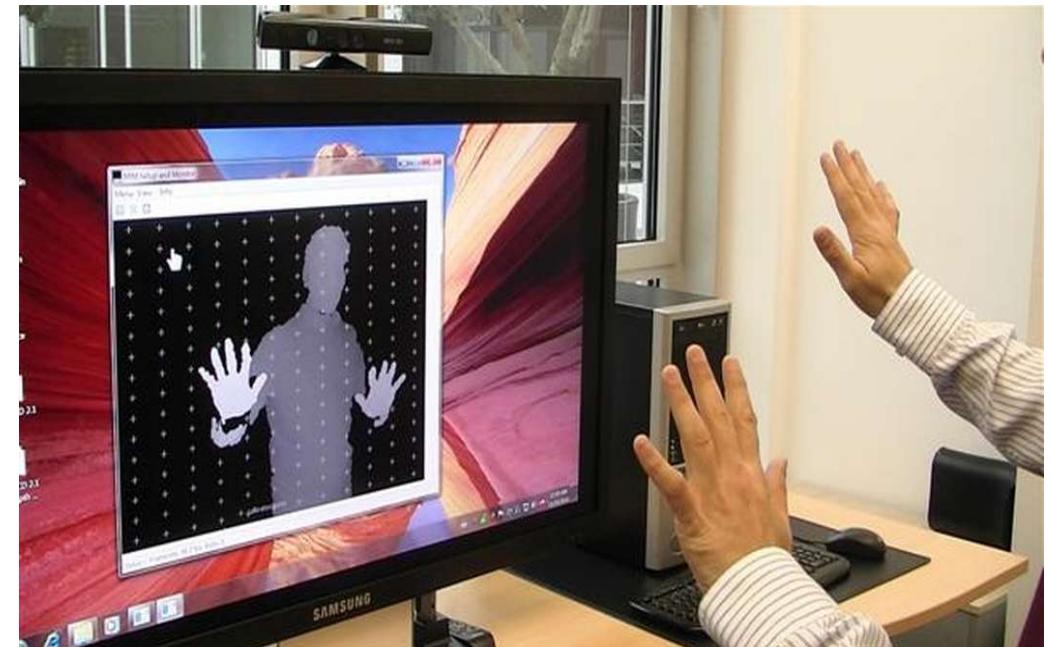
Regression

Predicting a *continuous-valued* variable

Classification

Predicting a *discrete-valued* variable
i.e., distinguishing between different
classes of data

Ex. Hand gesture recognition



Regression

Predicting a *continuous-valued* variable

Classification

Predicting a *discrete-valued* variable
i.e., distinguishing between different
classes of data

Ex. Speech recognition



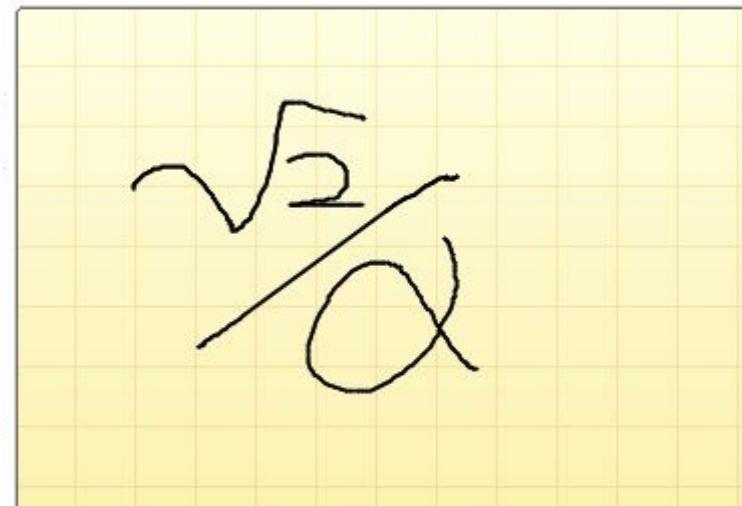
Regression

Predicting a *continuous-valued* variable

Classification

Predicting a *discrete-valued* variable
i.e., distinguishing between different
classes of data

**Ex. Optical character
recognition**



$$\sqrt{2}/\alpha$$

Regression

Predicting a *continuous-valued* variable

Classification

Predicting a *discrete-valued* variable
i.e., distinguishing between different
classes of data

Ex. Sentiment analysis

The screenshot shows a user interface for sentiment analysis. At the top right is a Twitter logo with a plus sign. Below it is a profile picture of a man and the handle @HVSVN. The main content area displays a tweet:
Don't fly [@BritishAirways](#). Their customer service is horrendous.

On the left, there's a section titled "Customer Revie" (partially cut off) showing a bar chart for "Apple Airport Expr". The chart has 45 reviews and the following distribution:

Star Rating	Count
5 star	(20)
4 star	(13)
3 star	(5)
2 star	(2)
1 star	(5)

Below the chart, there are two reviews side-by-side:

The most helpful favorable review:
20 of 21 people found the following review helpful:
★★★★★ Airport Express Set-up Instructions
The CD that comes with the Airport Express has been useless to me in setting up a Windows XP computer to work with an AE. The instructions below should get you up and running.
1. First download the latest version of both the Airport Update and Airport Express Firmware Updater from [...]
[Read the full review >](#)
Published 3 months ago by S. Monroe

The most helpful critical review:
6 of 7 people found the following review helpful:
★★★☆☆ Works fine after a painfully difficult set up.
It took me a full day to work out the bugs in setting my Express up to work with my Mac Mini and my wife's Mac Powerbook. First it worked on one, but the other could not find it. Then it didn't work at all. There is a lot more involved in setting up your own wireless network and making decisions as to what level of security you want (with no ready explanation of what the...)
[Read the full review >](#)
Published 3 months ago by David Haggith

At the bottom, there's a link: [See more 5 star, 2 star, 1 star reviews](#)

Regression

Predicting a *continuous-valued* variable

Classification

Predicting a *discrete-valued* variable
i.e., distinguishing between different
classes of data

Ex. Spam detection



Linear Classification

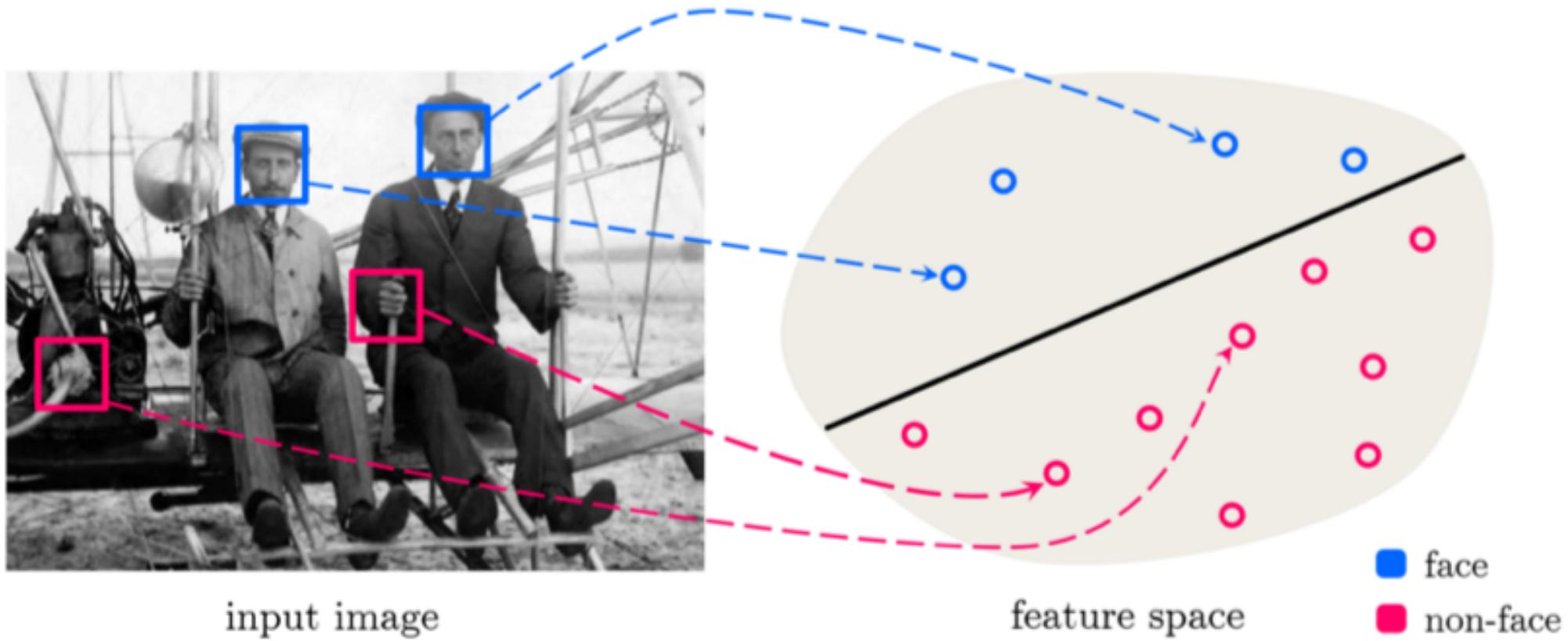
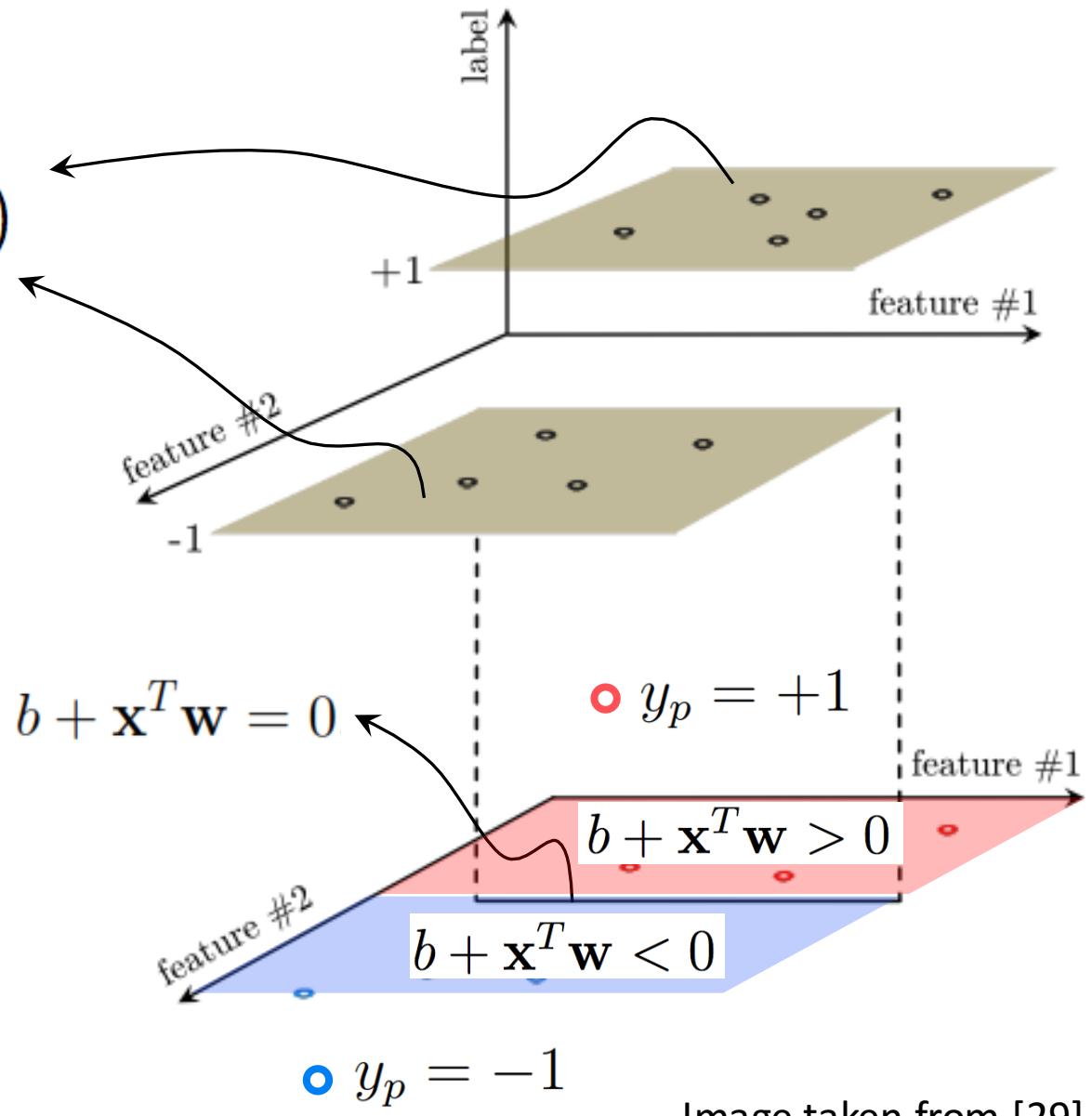


Image taken from [29]

Logistic regression

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



Logistic regression

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

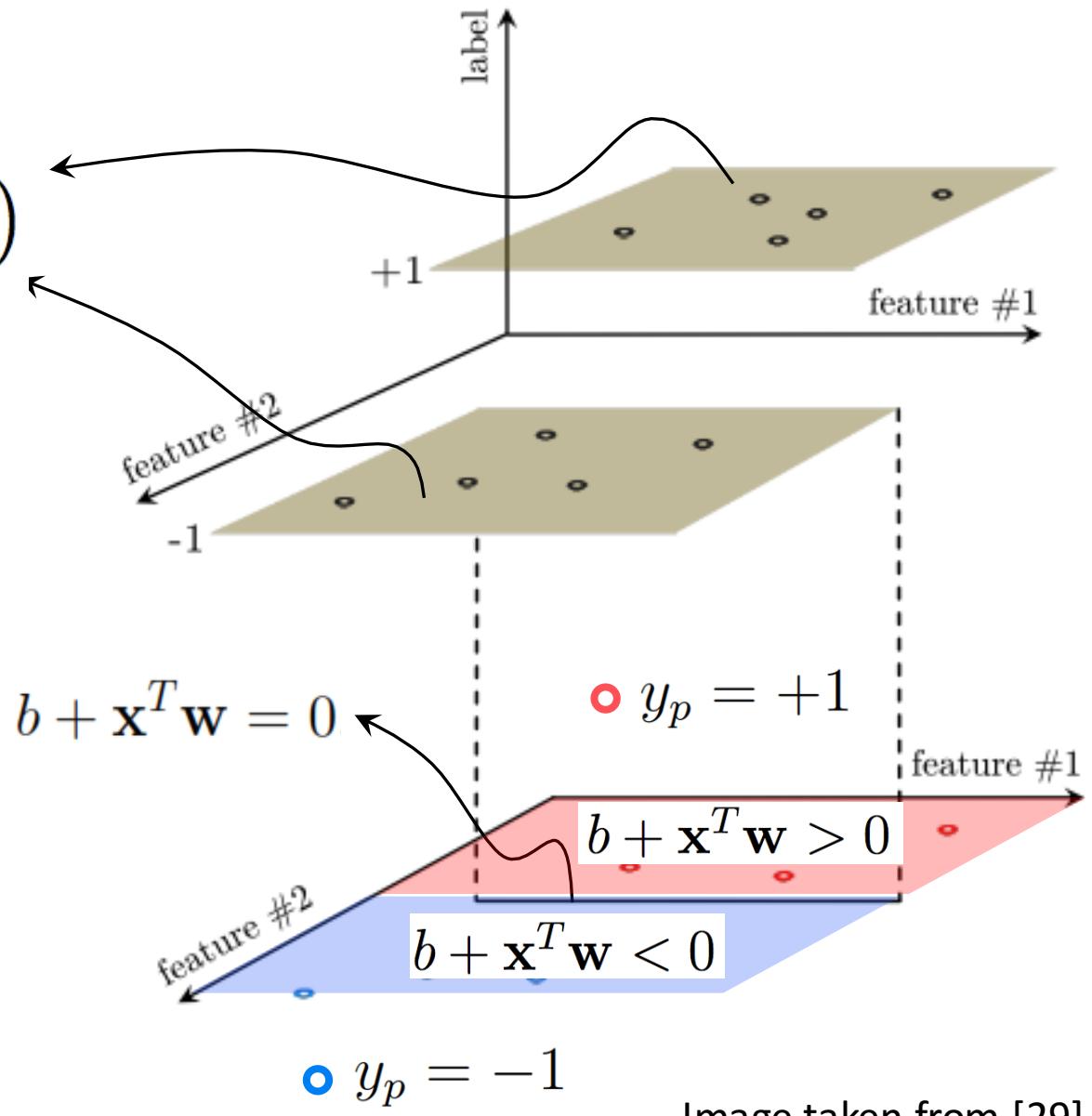


Image taken from [29]



of [29]) one can form a recovery problem whose solution gives parameters satisfying this desired relationship



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

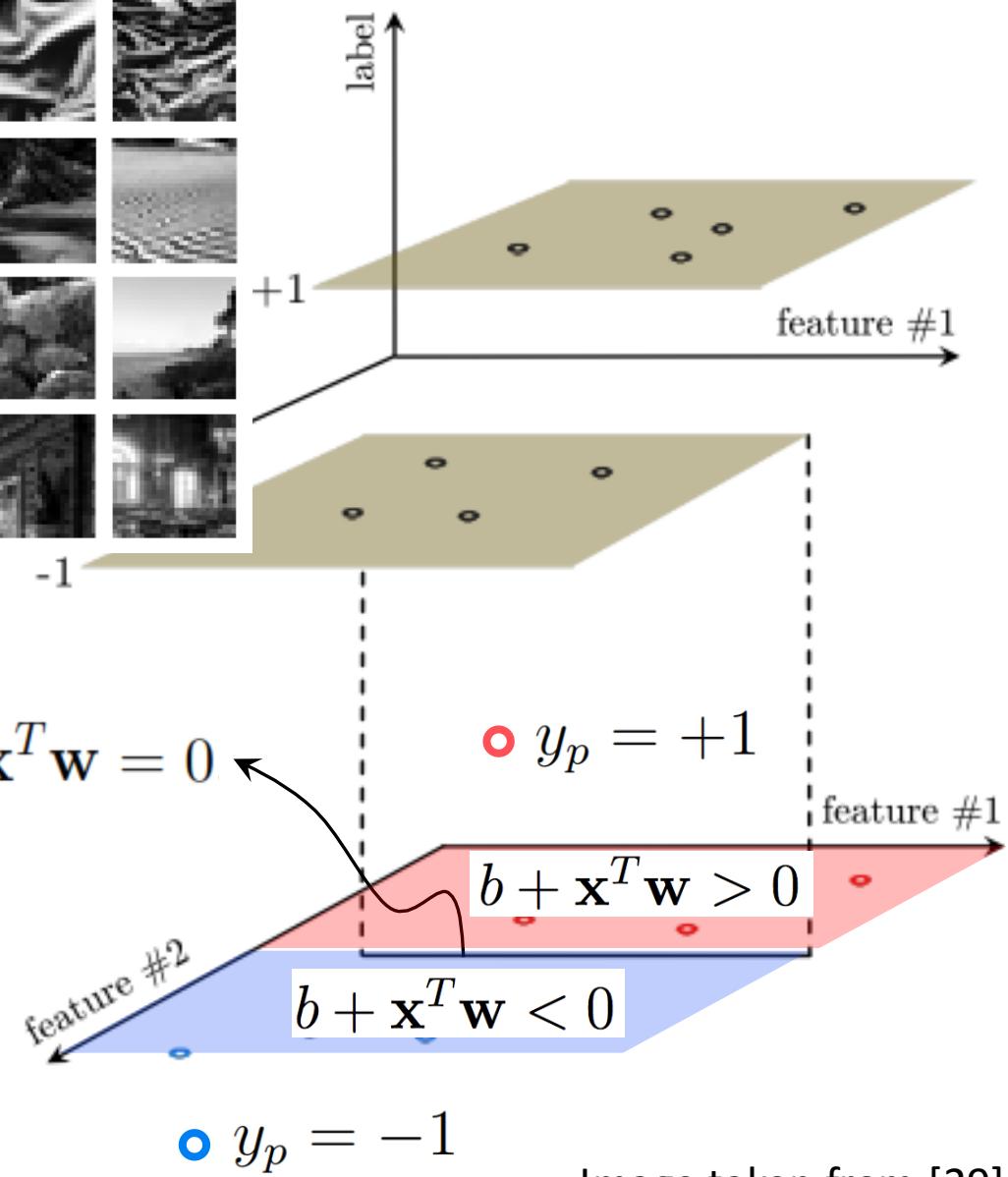


Image taken from [29]

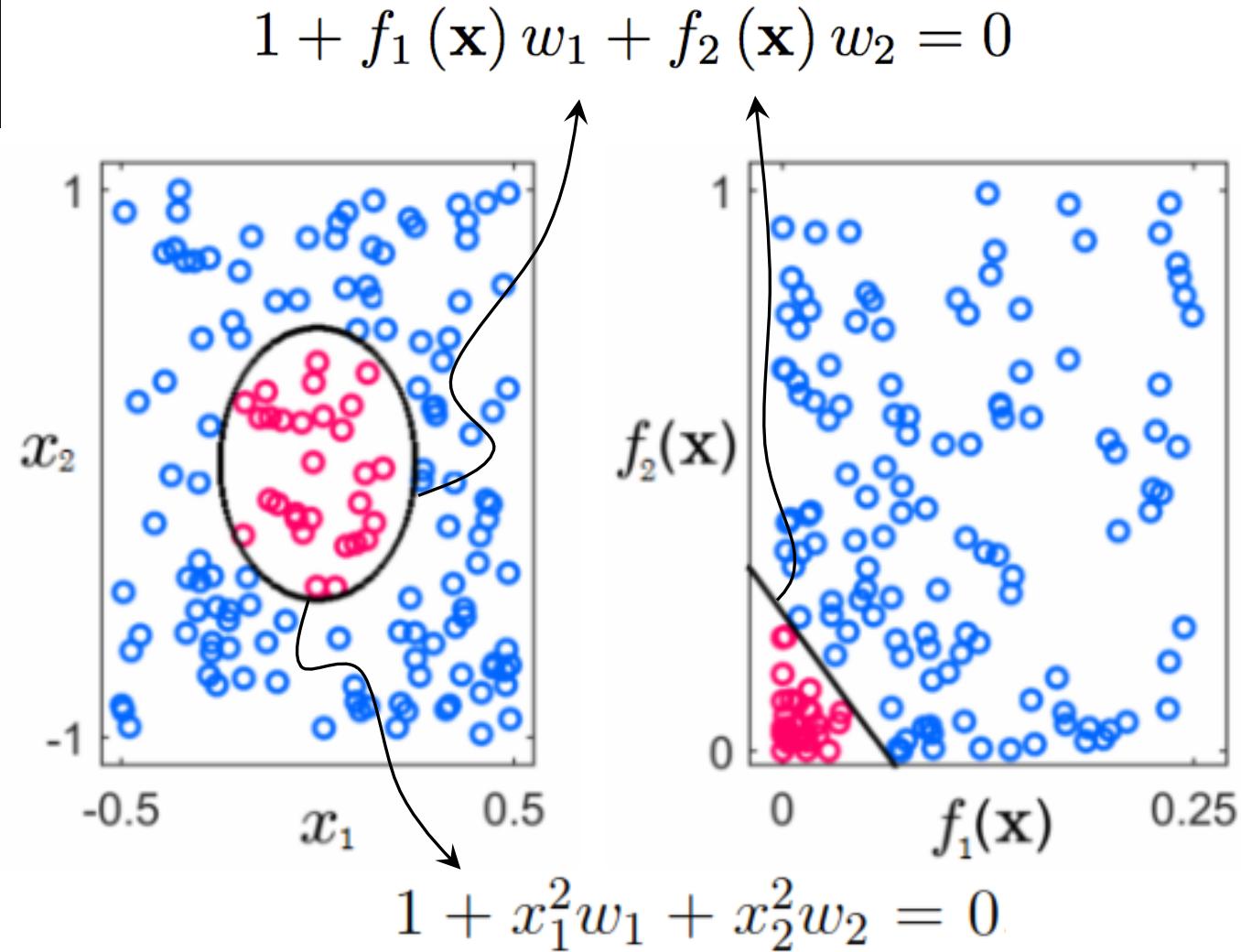
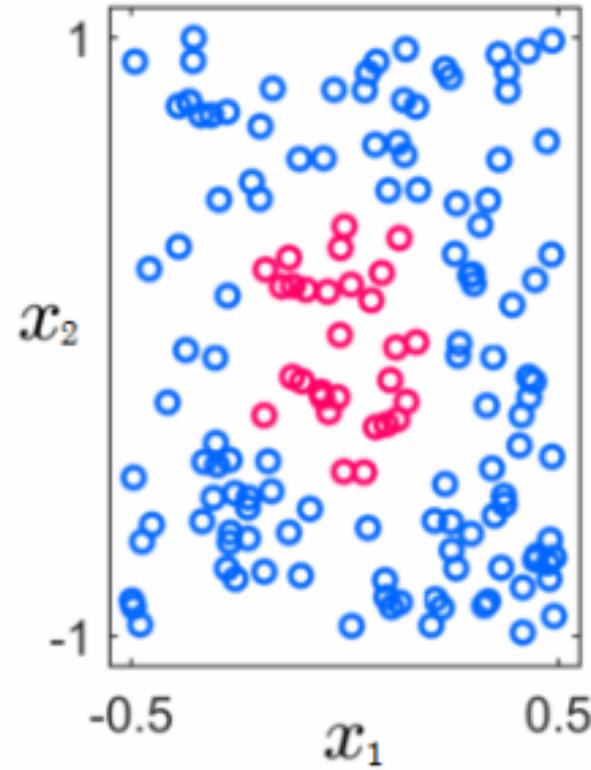
What are “features” ?

Features

What are features?

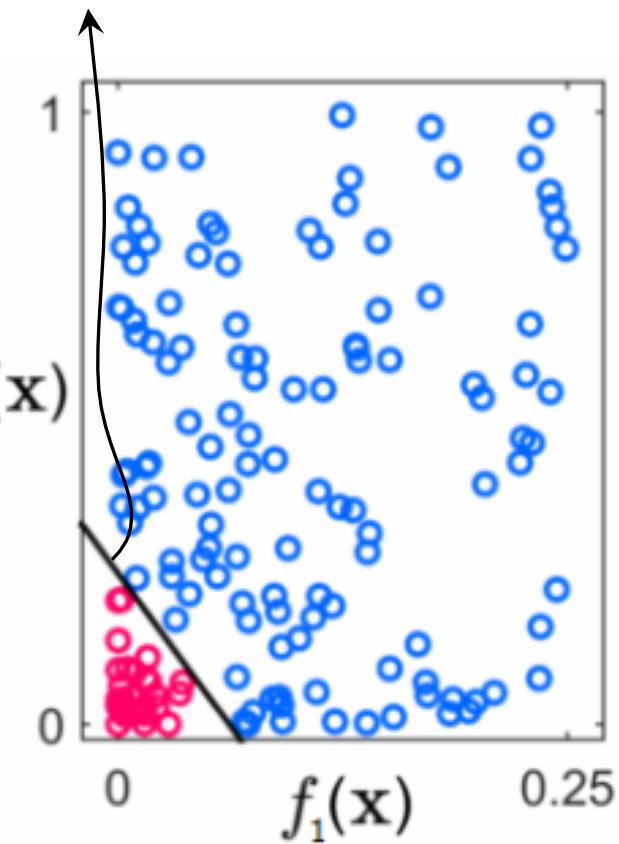
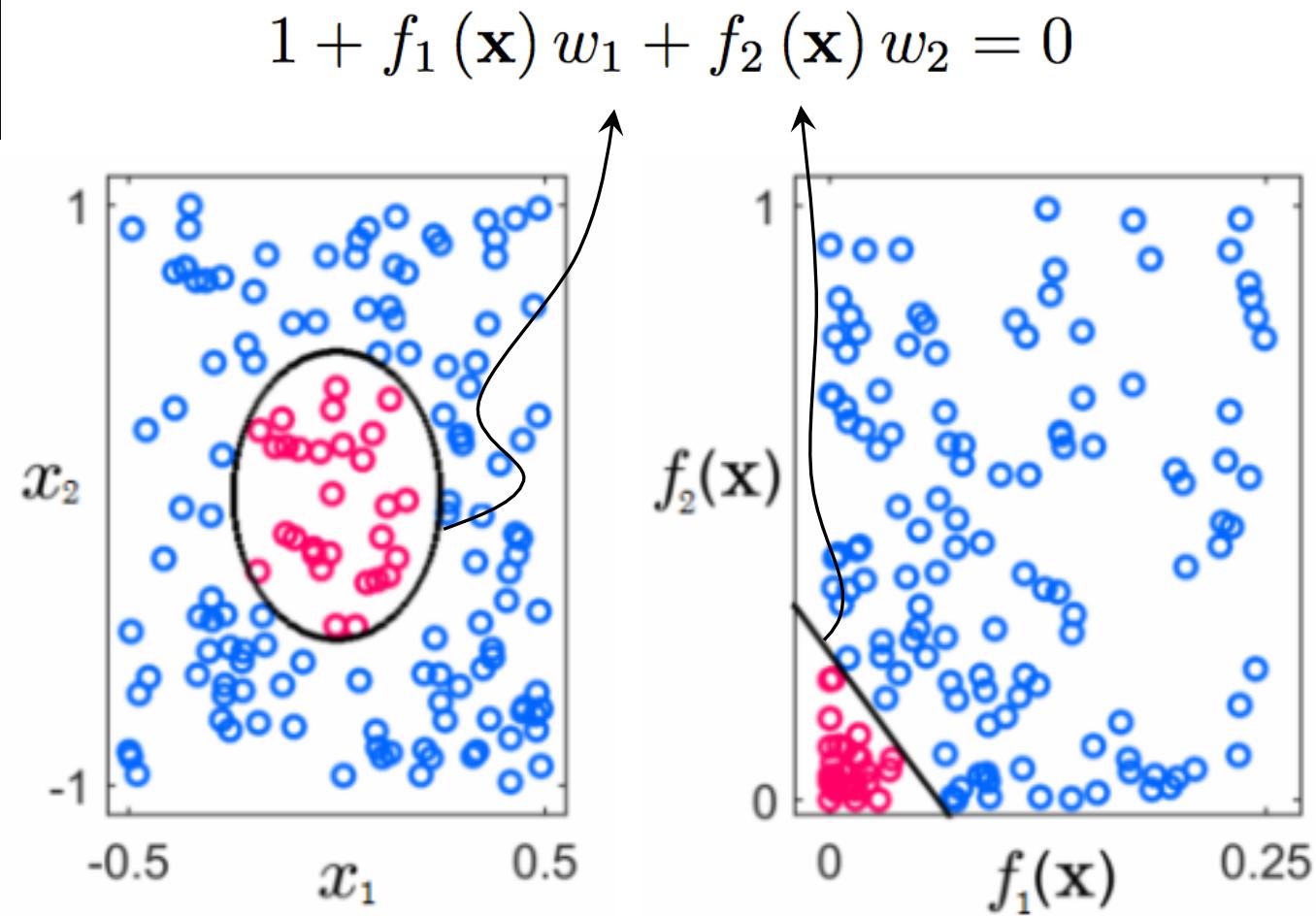
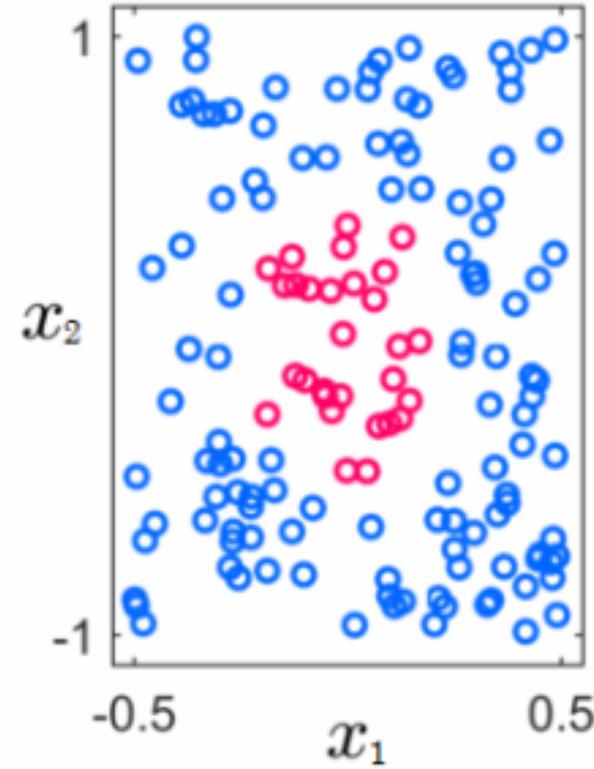
- **Geometrically:** features are transformations of the input that provoke a good *nonlinear* fit/ separation
- **Philosophically:** features are those defining characteristics of the phenomenon underlying a dataset that allow for optimal learning

Feature design: classification



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

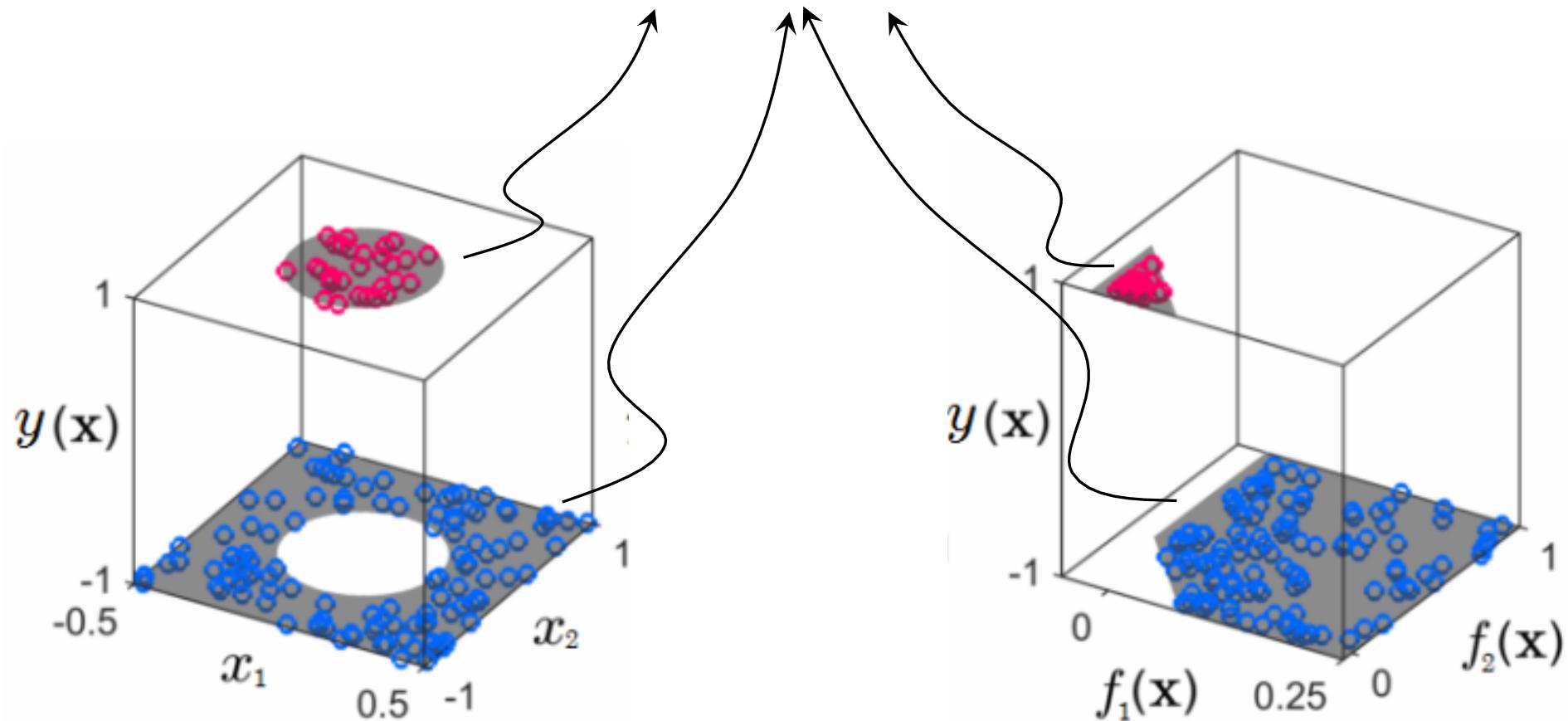
Feature design: classification



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.

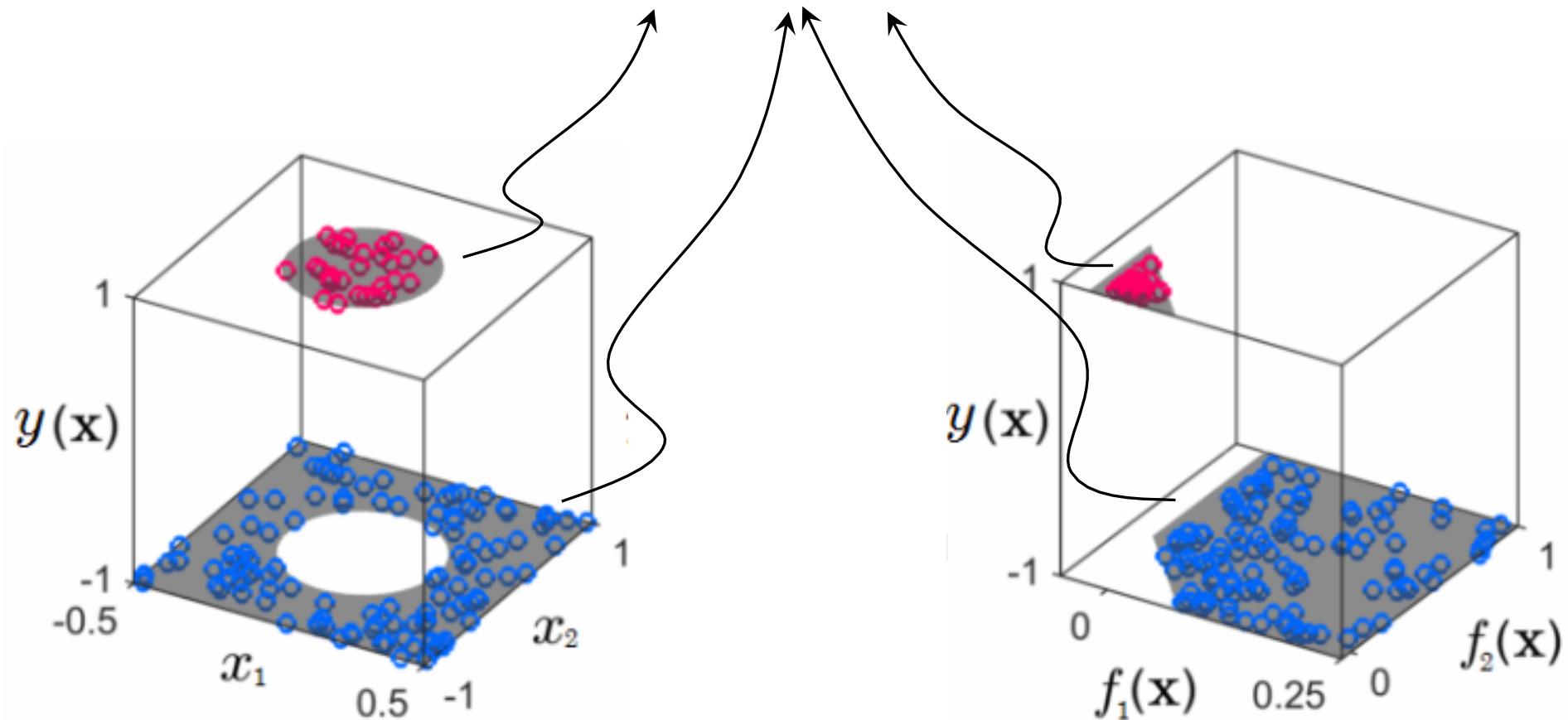
Full view of data

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



Full view of data

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



Features: summary (re-visited)

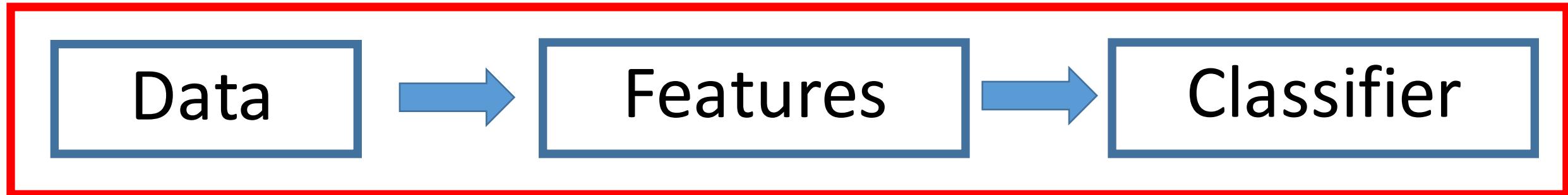
What are features?

- **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?

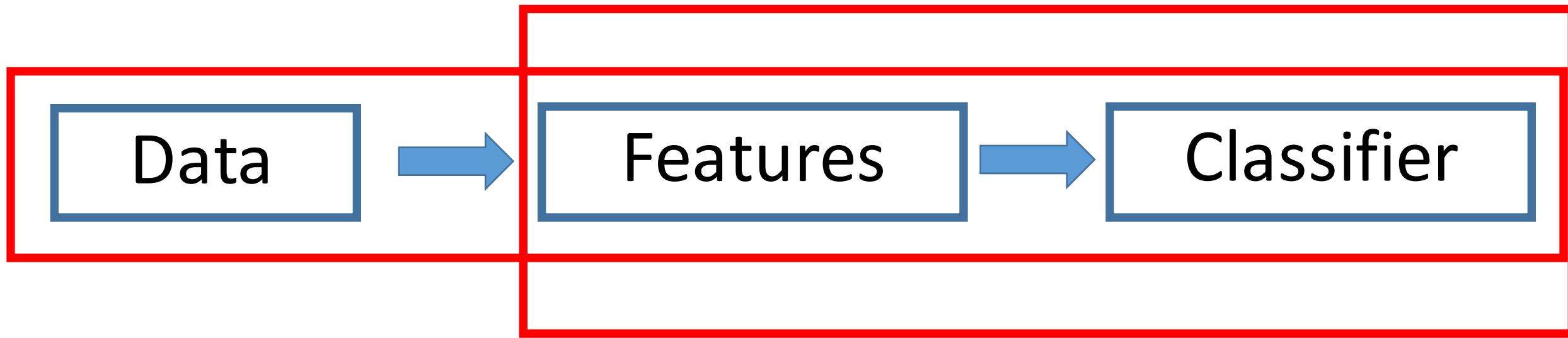
- **By hand (engineered):** translate understanding of a phenomenon into a set of geometric transformations of the input – often requires expertise
 - ❖ Requires strong knowledge to produce reasonable results
- **Learn from data (automatic):** use bases (e.g., neural networks) to determine directly from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

Detection Pipeline



- UCI ML repo
 - KD Nuggets
 - Kaggle
 -
- scikit-learn
 - scikit-image
 - opencv
 - tensorflow / theano / caffe

Detection Pipeline



These two can be performed simultaneously

- e.g., Viola-Jones face detector, deep neural nets, kernels,...

Classic (hand)
engineered features

Commonly-used engineered features

TEXT

- Sentiment analysis

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 $f_1(\mathbf{x}), \dots, f_M(\mathbf{x})$ of the input data, but they are not so easily expressed algebraically.

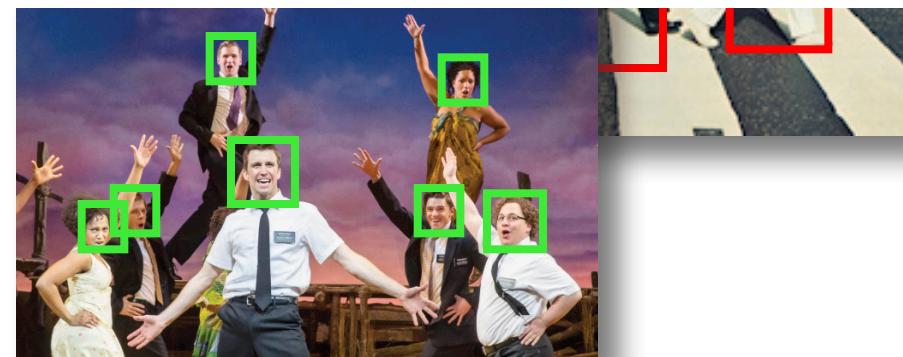
Instructions
The CD that comes with the Airport Express has been useless to me in setting up a Windows XP computer to work with an AE. The instructions below should get you up and running.

1. First download the latest version of both the Airport Update and Airport Express Firmware Updater from [...]

2. Run the latest version of the Airport Update (4.1 at the...
[Read the full review >](#)
Published 3 months ago by David Haggith

3. See more [5 star](#), [2 star](#), [1 star](#) reviews
Published 3 months ago by S. Monroe

4. See more [5 star](#), [4 star](#) reviews



IMAGE

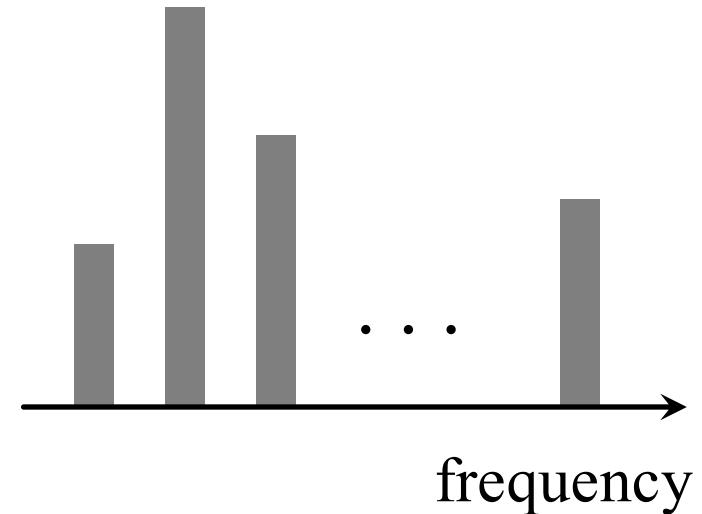
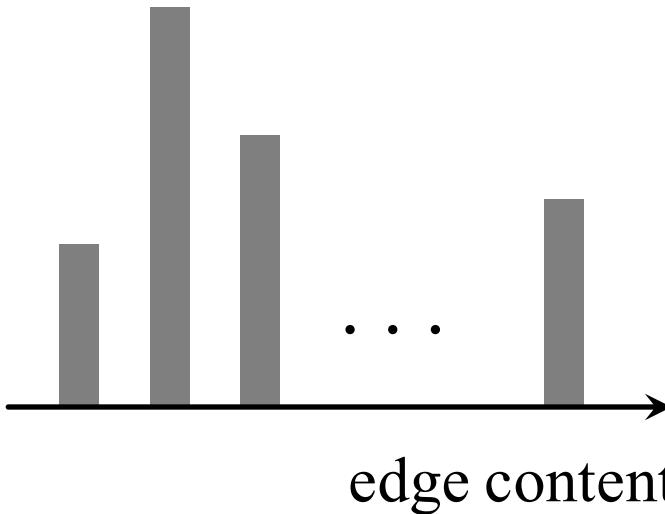
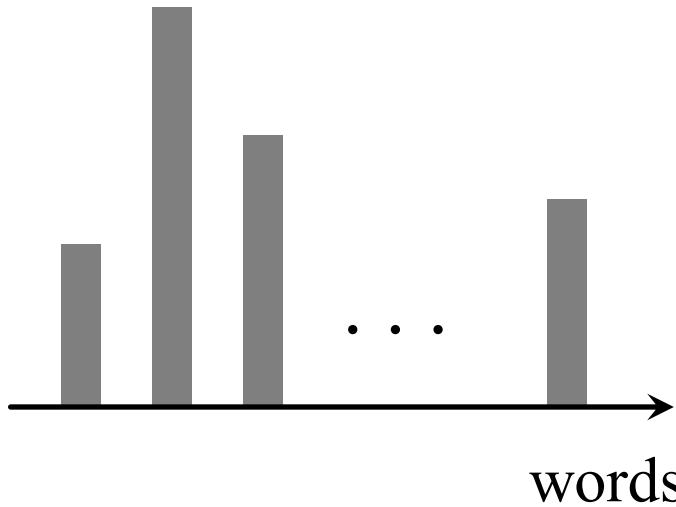
- Pedestrian detection

AUDIO

- Speech recognition



Histogram features



ALL CHARACTERS AND
EVENTS IN THIS SHOW--
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



Bag of Words (BoW) histogram

Parsing

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

dogs / are / the / best

cats / are / the / worst

Stop word removal

dogs / ~~are~~ / ~~the~~ / best

cats / ~~are~~ / ~~the~~ / worst

Stemming

~~dog~~ best

~~cat~~ worst

Merging

best cat dog worst

Normalized vector representation

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

Sentiment analysis with BoWs



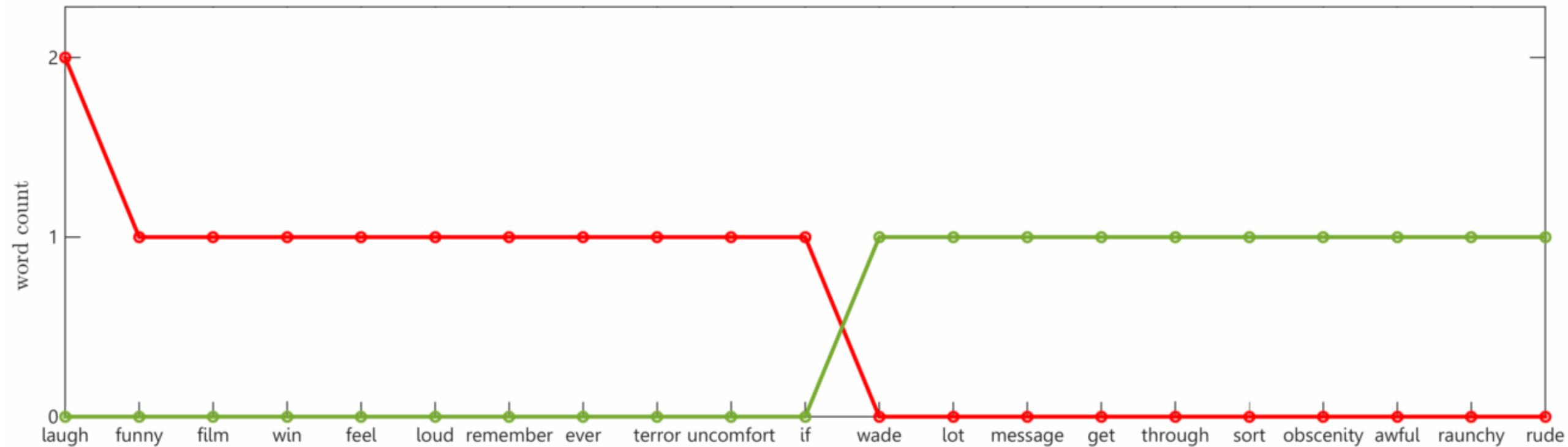
One of the funniest films you will ever feel this uncomfortable about laughing out loud at. Remember if you laugh, the terrorists win!

Kevin Ranson
MovieCrypt.com

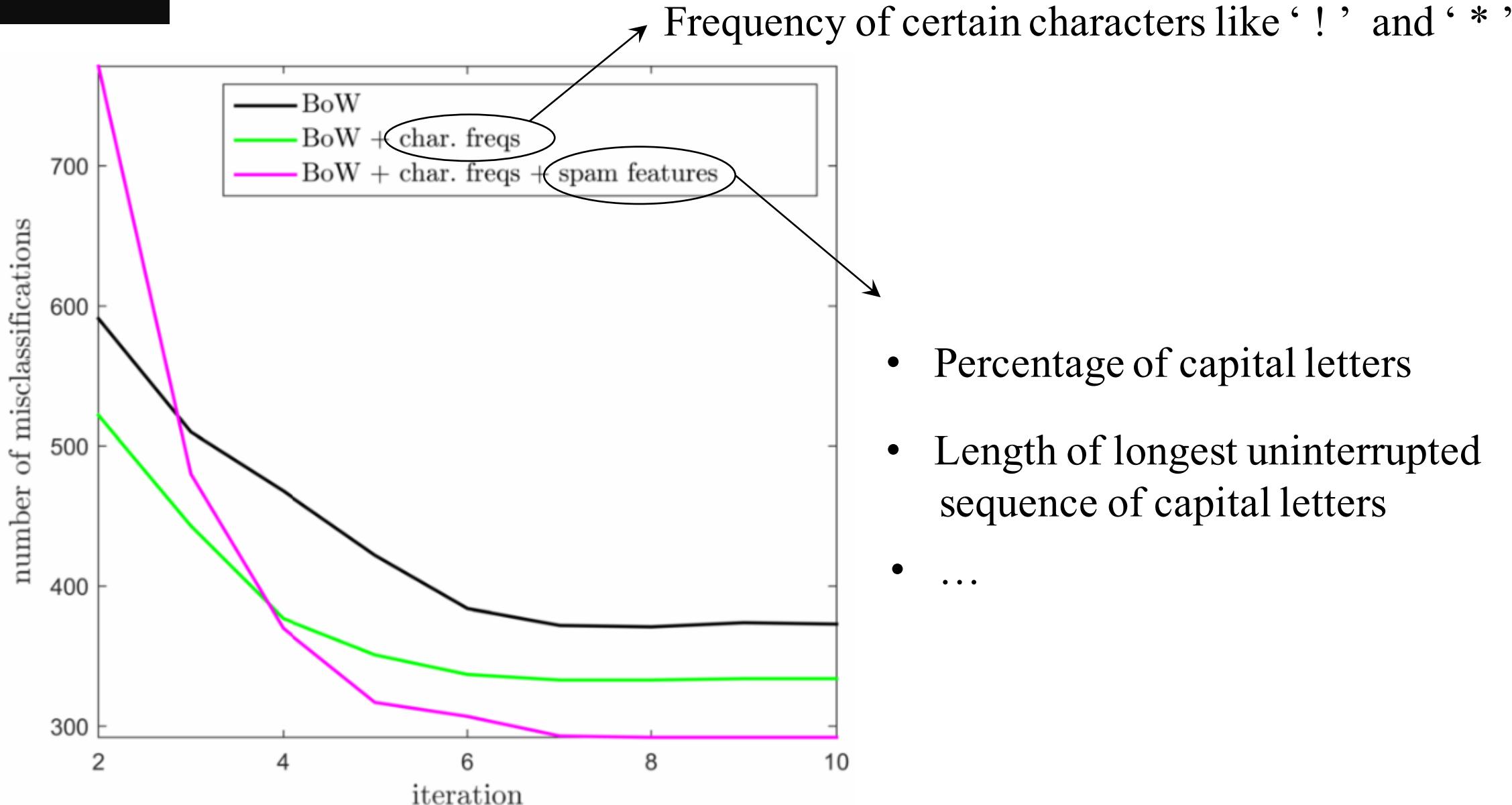


It is rude and raunchy, and it has a message, sort of. You have to wade through an awful lot of obscenity to get to it though.

Roger Moore
Orlando Sentinel



Spam detection



Images and edges

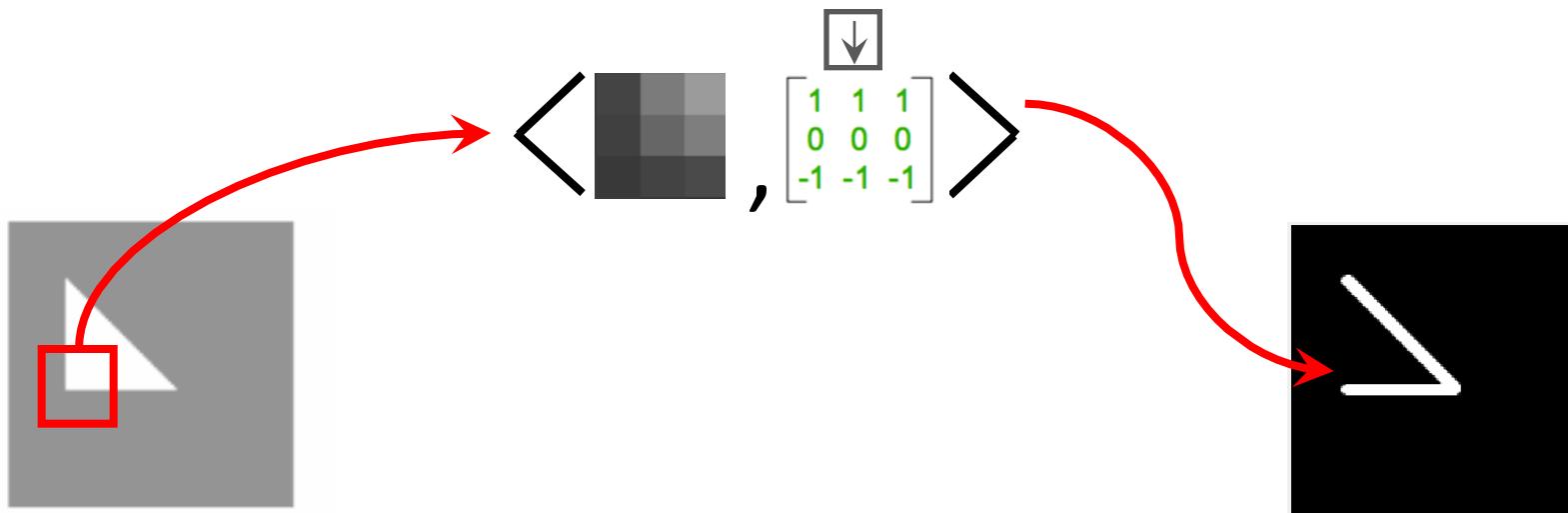


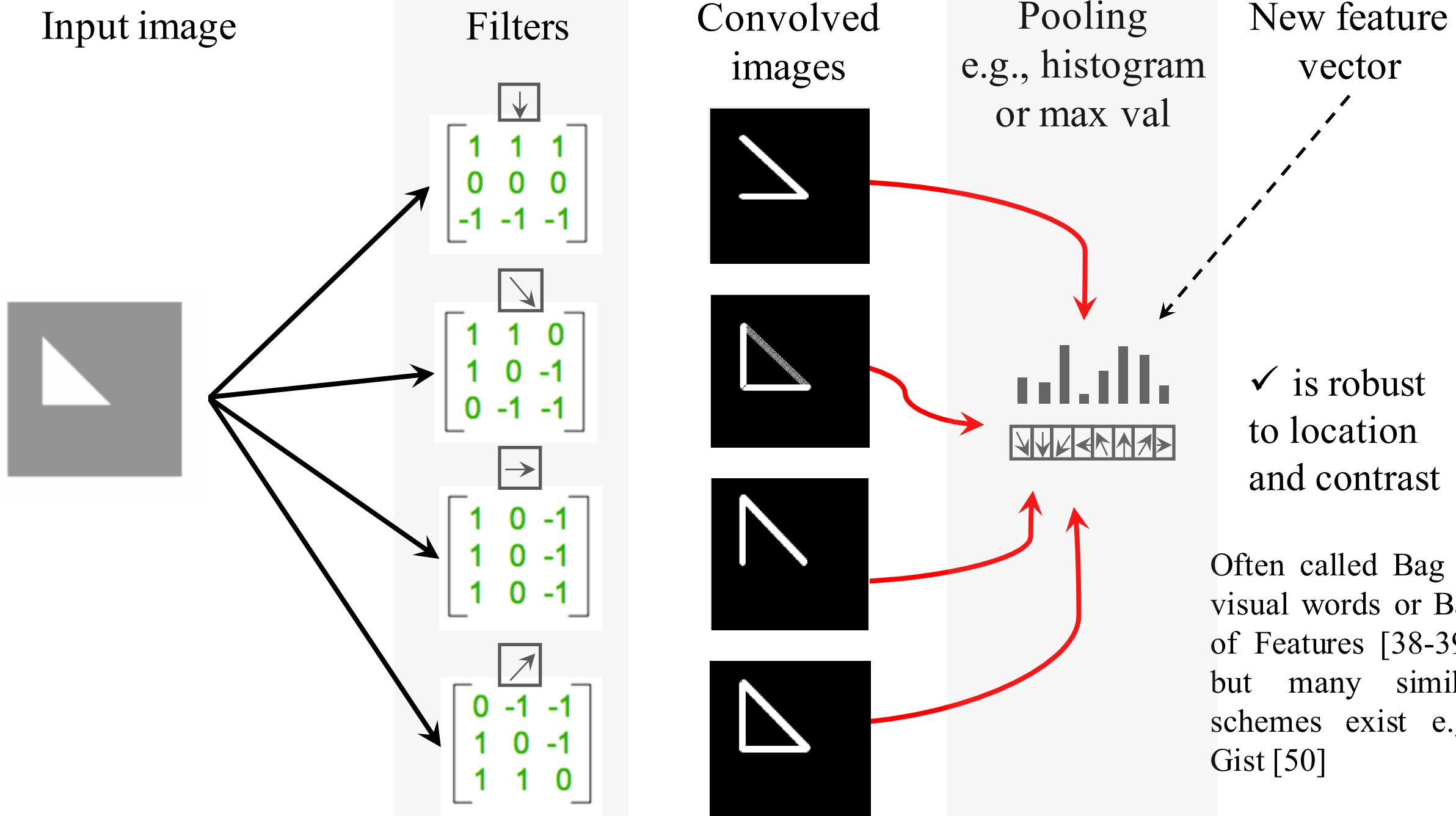
Image taken from [29]

Image histogram features: pixel values or edge orientations?

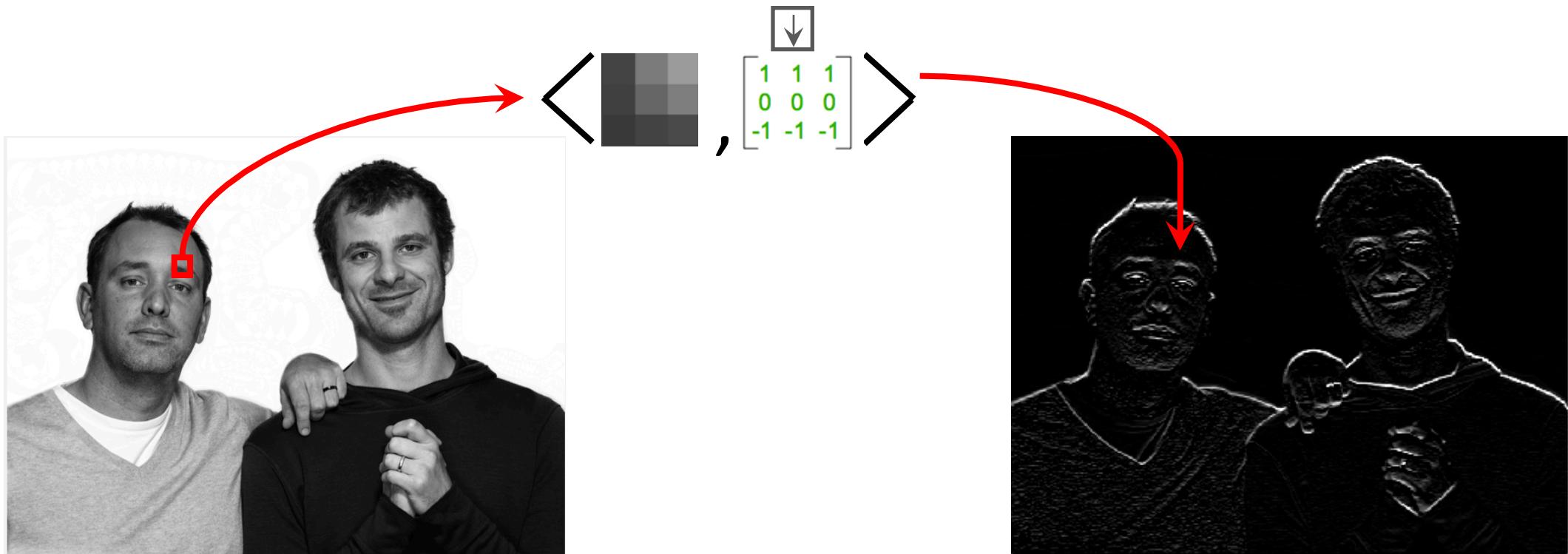


Discrete convolution





Discrete convolution



Input image



Filters

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

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

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

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

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

\downarrow

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

\swarrow

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

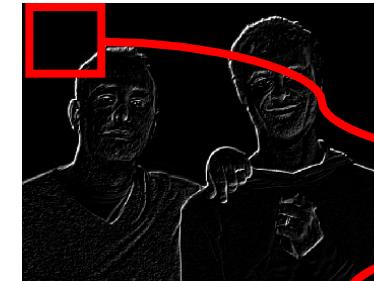
\rightarrow

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

\nearrow

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

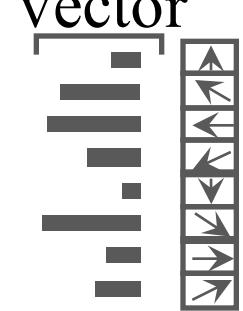
Convolved
images



Pooling
e.g., histogram
or max val



New feature
vector



Input image



Filters

\downarrow

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

\swarrow

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

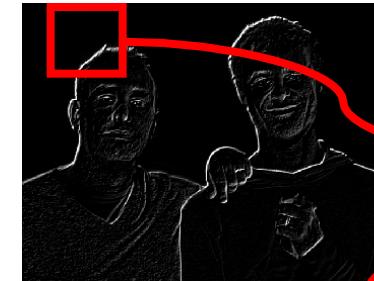
\rightarrow

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

\nearrow

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

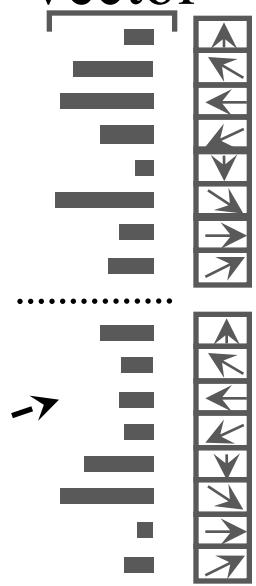
Convolved
images



Pooling
e.g., histogram
or max val



New feature
vector



Input image



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]

Filters

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

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

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

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

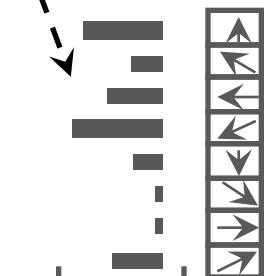
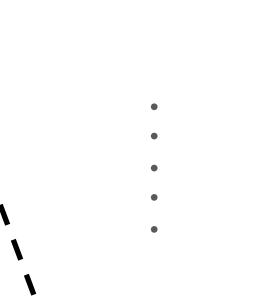
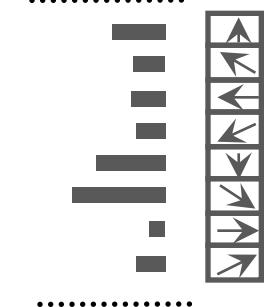
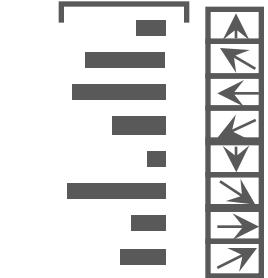
Convolved images



Pooling e.g., histogram or max val



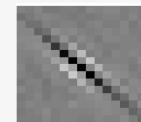
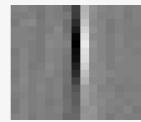
New feature vector



Input image

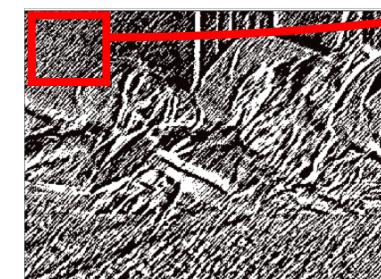
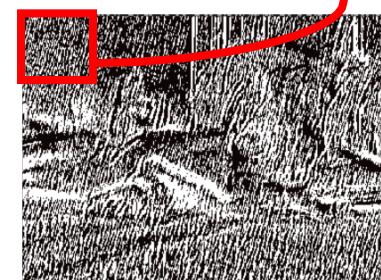
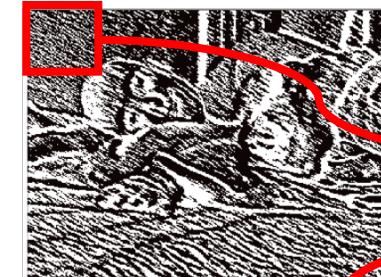
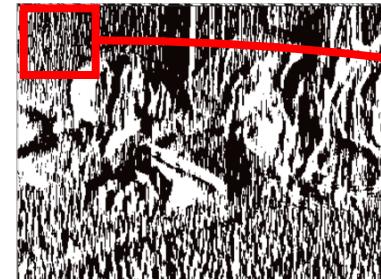


Filters



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]

Convolved images



Pooling

e.g., histogram or max val



New feature vector

$$\begin{bmatrix} \uparrow \\ \vdash \\ \leftarrow \\ \downarrow \\ \rightarrow \end{bmatrix}$$

$$\begin{bmatrix} \uparrow & \vdash & \leftarrow & \downarrow & \rightarrow \end{bmatrix}$$

$$\begin{bmatrix} \uparrow & \vdash & \leftarrow & \downarrow & \rightarrow \end{bmatrix}$$

$$\begin{bmatrix} \uparrow & \vdash & \leftarrow & \downarrow & \rightarrow \end{bmatrix}$$

$$\begin{bmatrix} \uparrow \\ \vdash \\ \leftarrow \\ \downarrow \\ \rightarrow \end{bmatrix}$$

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]

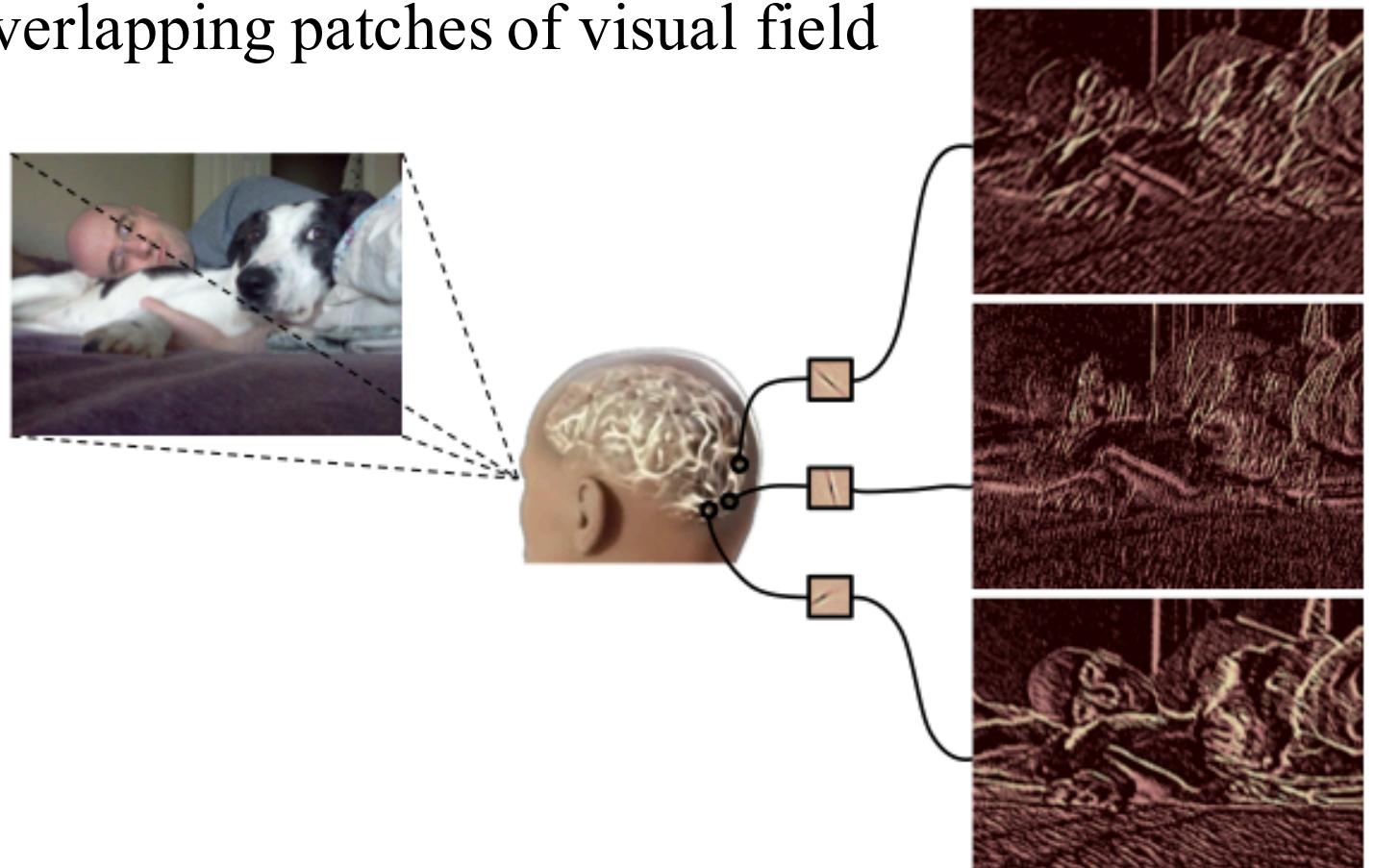


Image taken from [29]

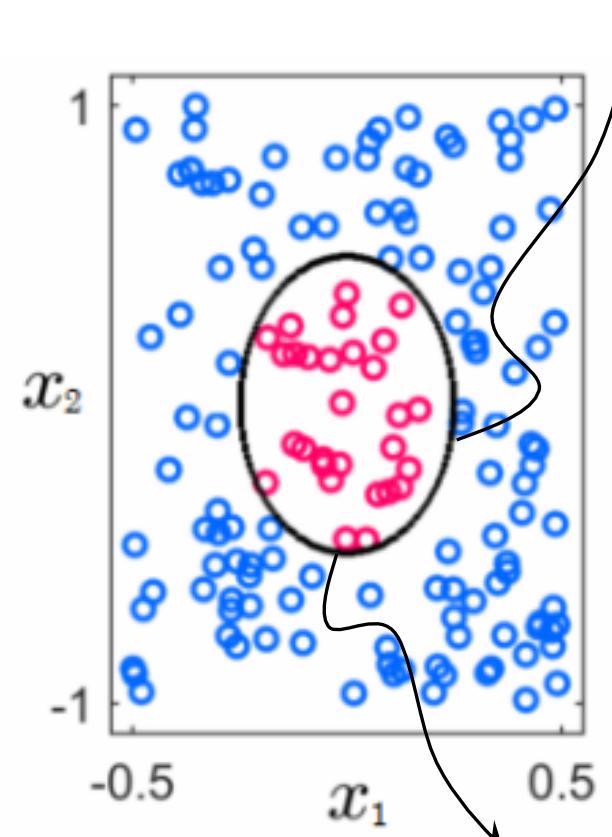
Commonly-used engineered features

Feature transformations for real data aim at ensuring that instances with different classes are "dissimilar". Transformations $f_1(\mathbf{x}), \dots, f_M(\mathbf{x})$ of the input \mathbf{x} can be done algebraically.

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



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



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

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!)