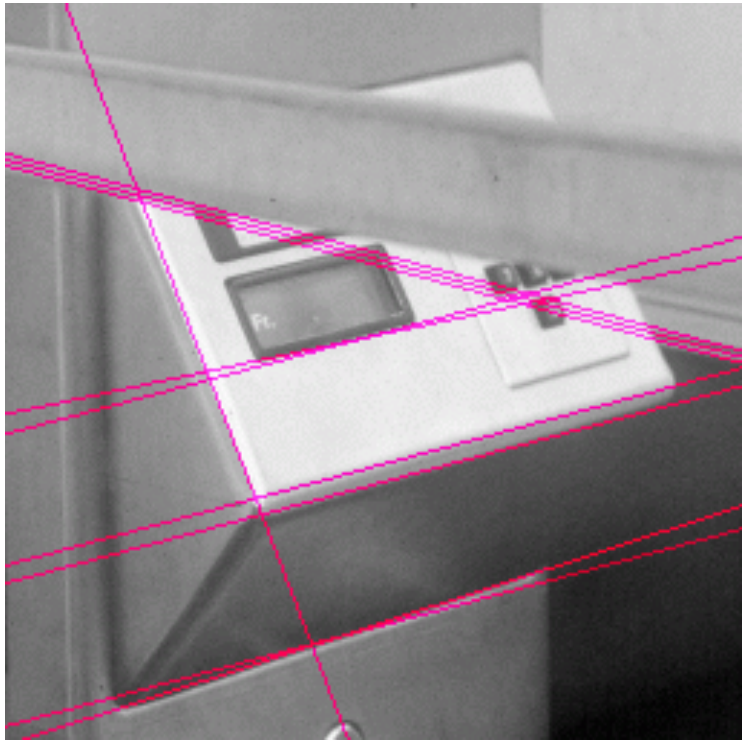
Classification



Dan Witzner Hansen

Today

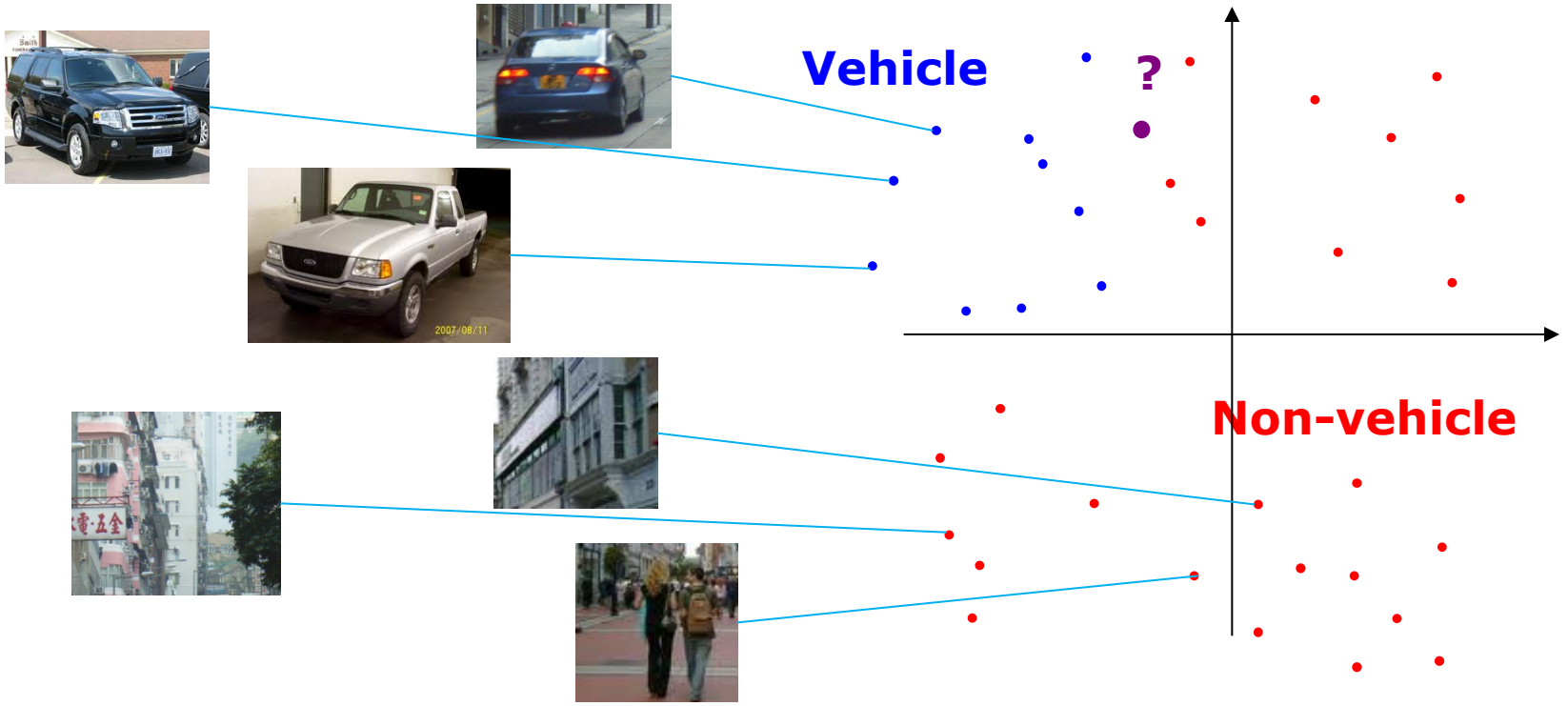
- Linear Classification (binary)
- HOG feature

- Key Concepts:
  - Classification as regression
  - Decision boundary
  - Loss function
  - Metrics to evaluate classification

SUMMARY OF VIDEO LECTURE

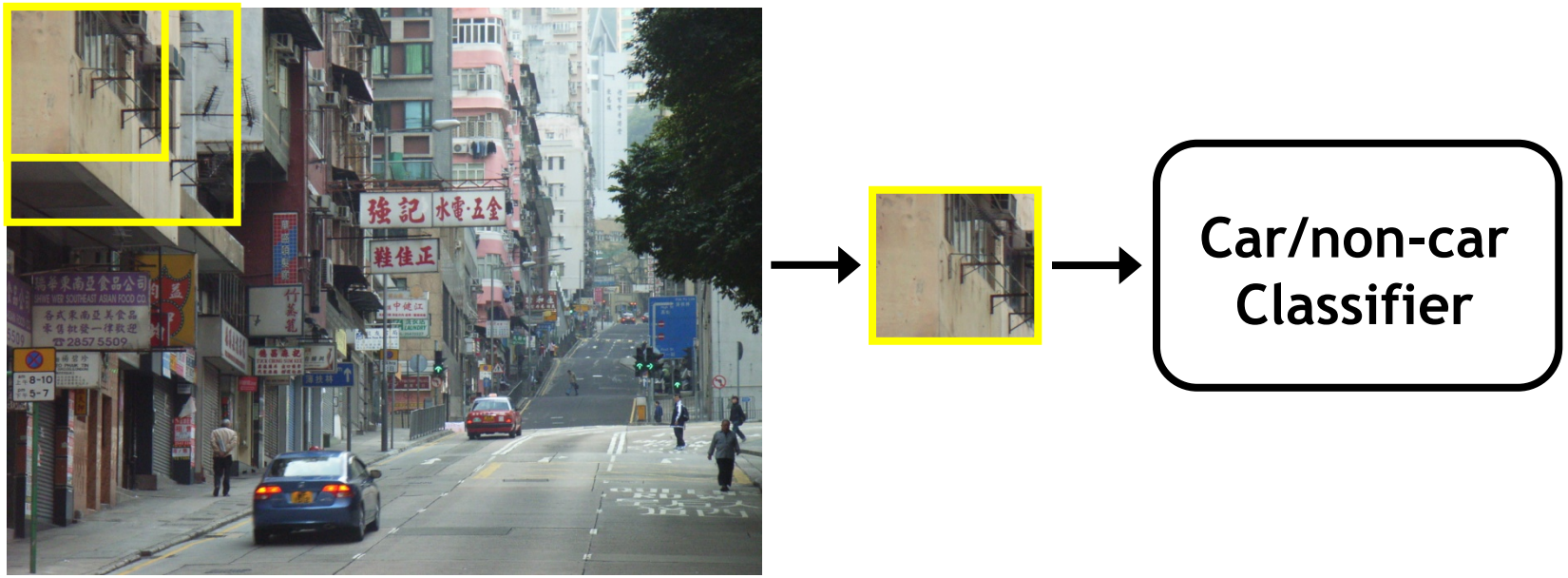
Image Classification
Supervised learning

- Provide training images to learn what are vehicles and what are not.



Sliding windows Object detection from object classification

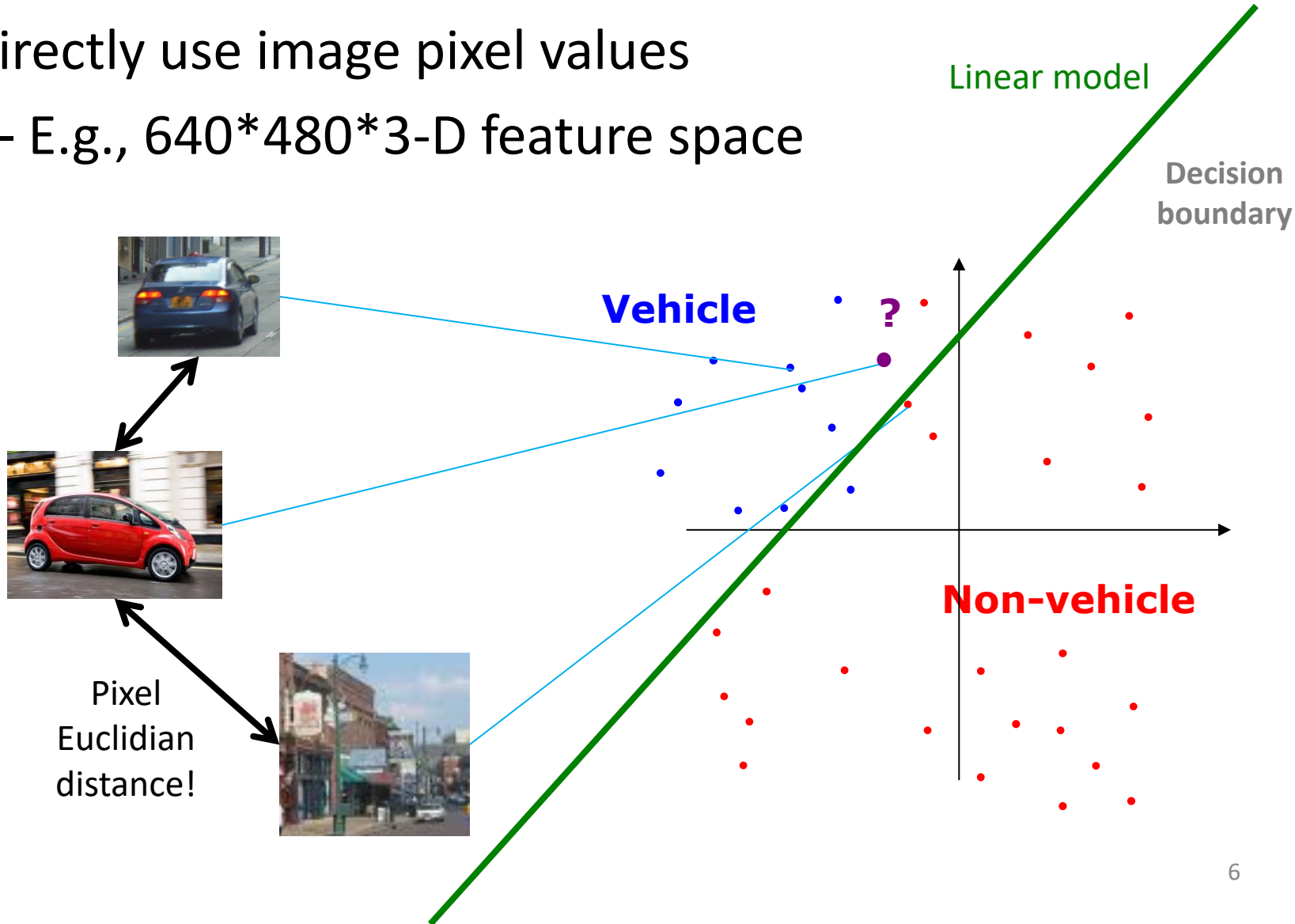
– Test all possible image segments (also different sizes)

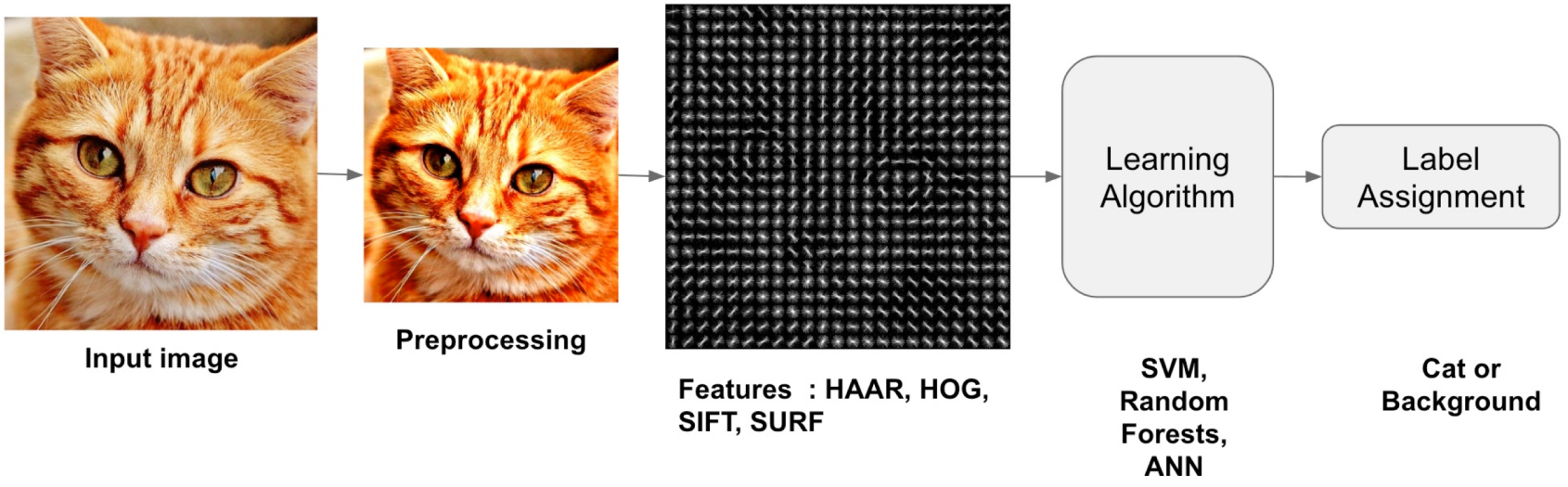


Slide credit: K. Grauman, B. Leibe

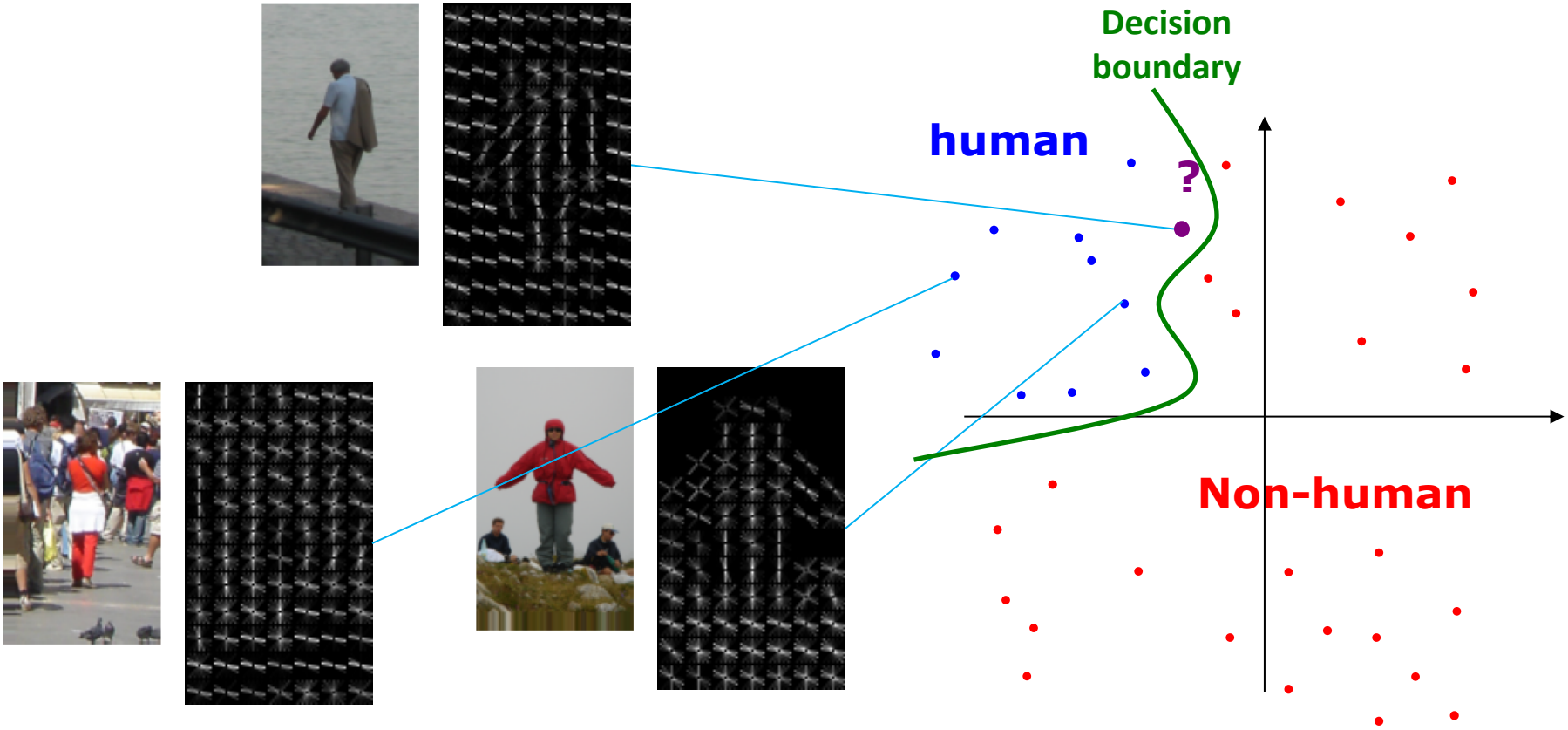
Simple solution – image pixels

- Directly use image pixel values
  - E.g., 640*480*3-D feature space



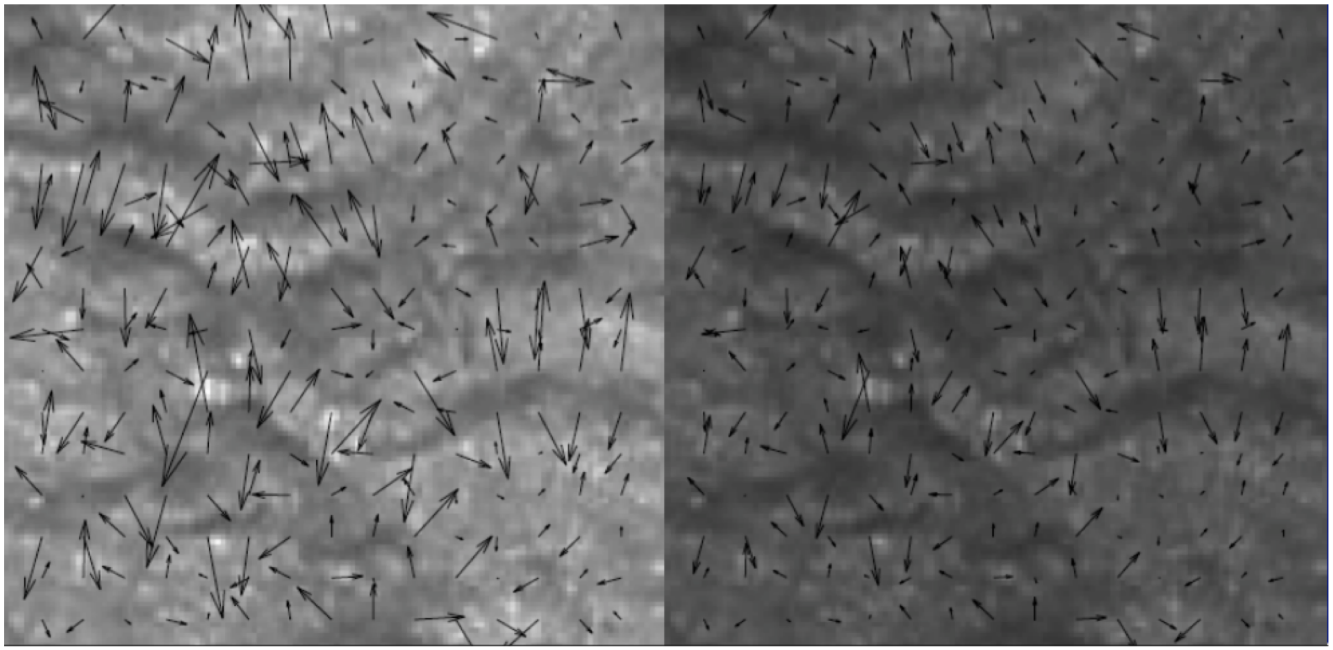

Learning Algorithm
Label Assignment
Input image
Preprocessing
Features : HAAR, HOG, SIFT, SURF
SVM, Random Forests, ANN
Cat or Background

Extracting features and Non-linear descision boundaries



Describe image patches by orientation

- Gradient Magnitude is affected by illumination changes – orientation is not



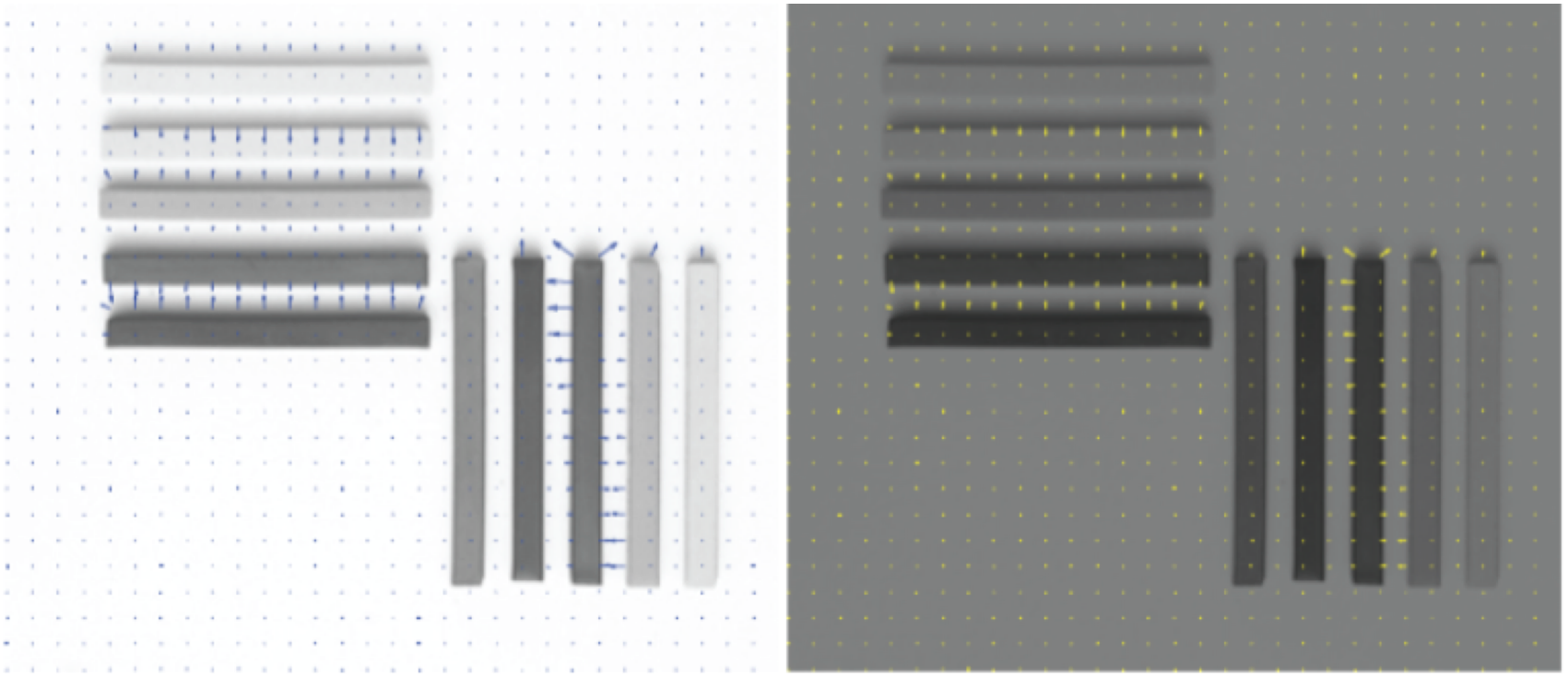
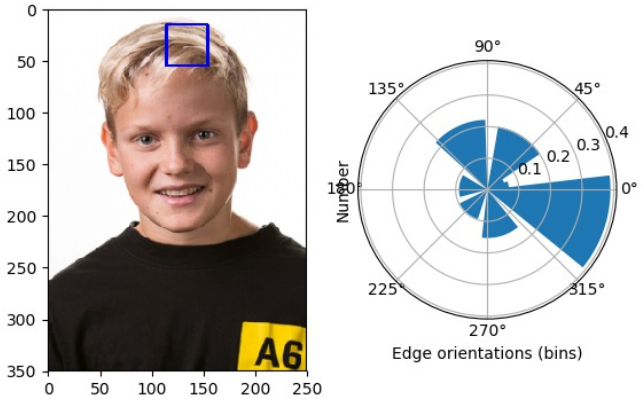
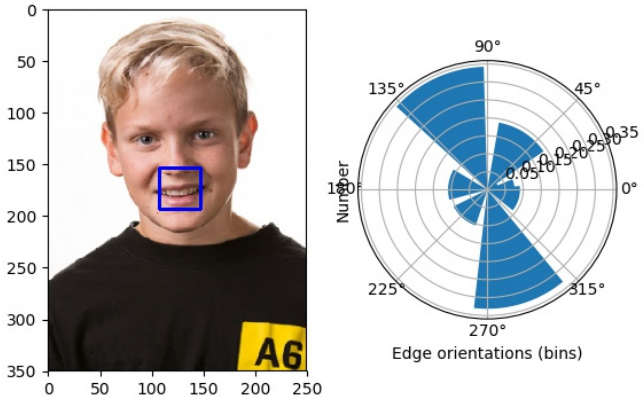
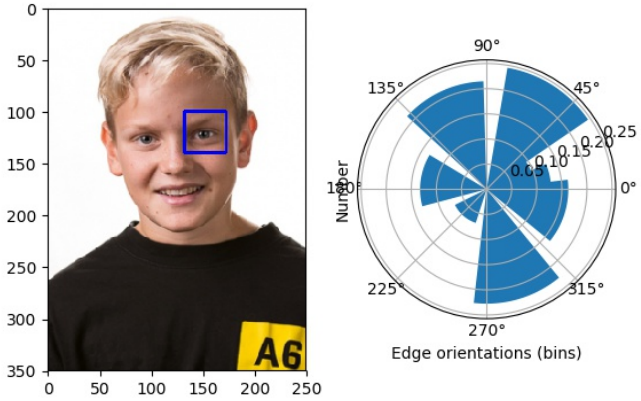


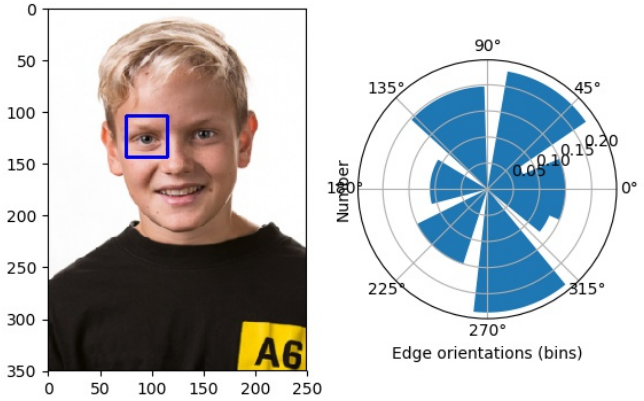
FIGURE 5.7: The magnitude of the image gradient changes when one increases or decreases the intensity. The orientation of the image gradient does not change; we have plotted every 10th orientation arrow, to make the figure easier to read. Note how the directions of the

Comparing orientation histograms

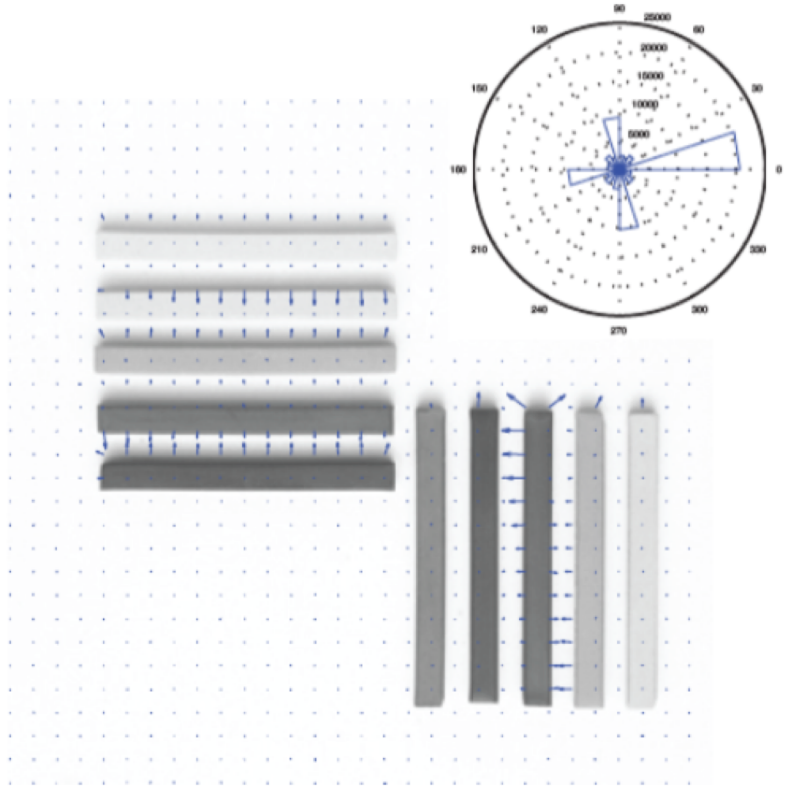


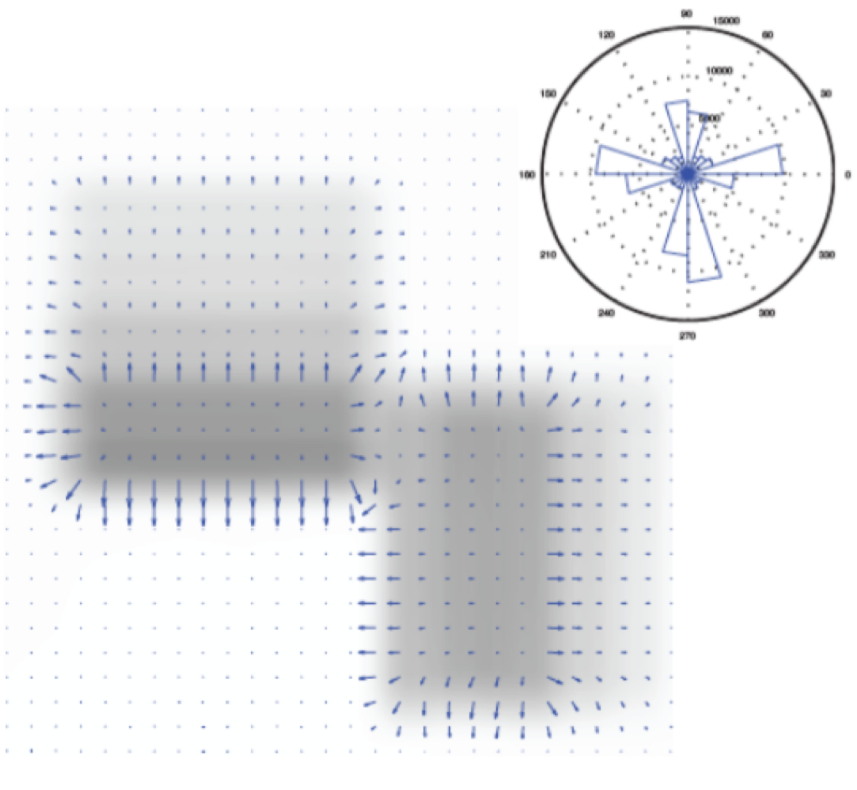






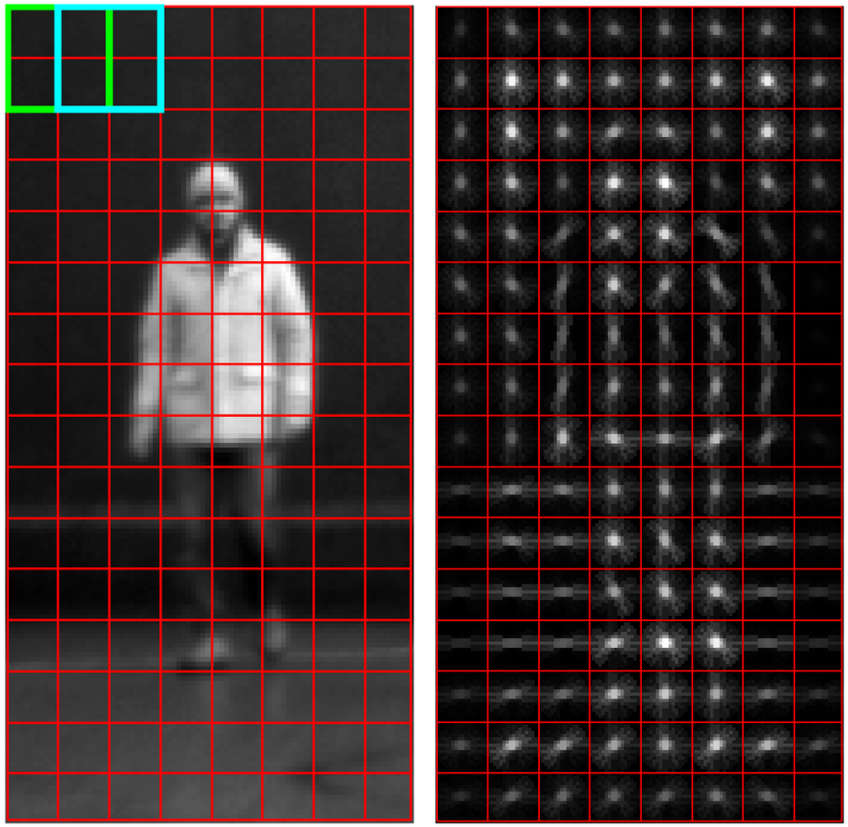
Orientation at different scales





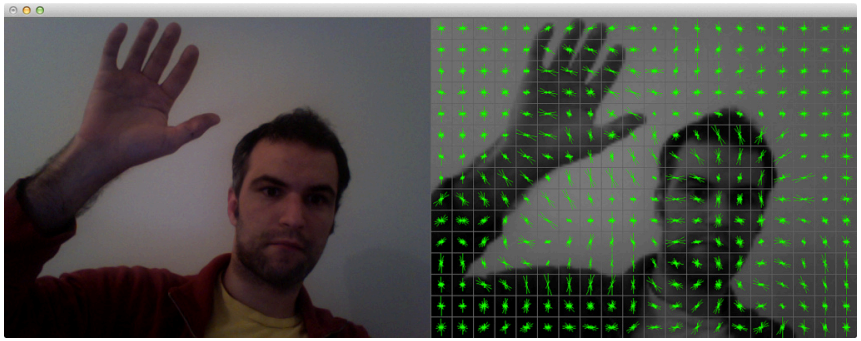
Idea of HOG features

- Divide image into small sub-images: “cells
- Cells can be rectangular (R-HOG) or circular (c-HOG)
- Accumulate a histogram of edge orientations within that cell
- The combined histogram entries are uses as the feature describing the object
- To provide better illumination invariance (light, shadows etc) normalize the cells across larger regions incorporating multiple cells: “blocks”

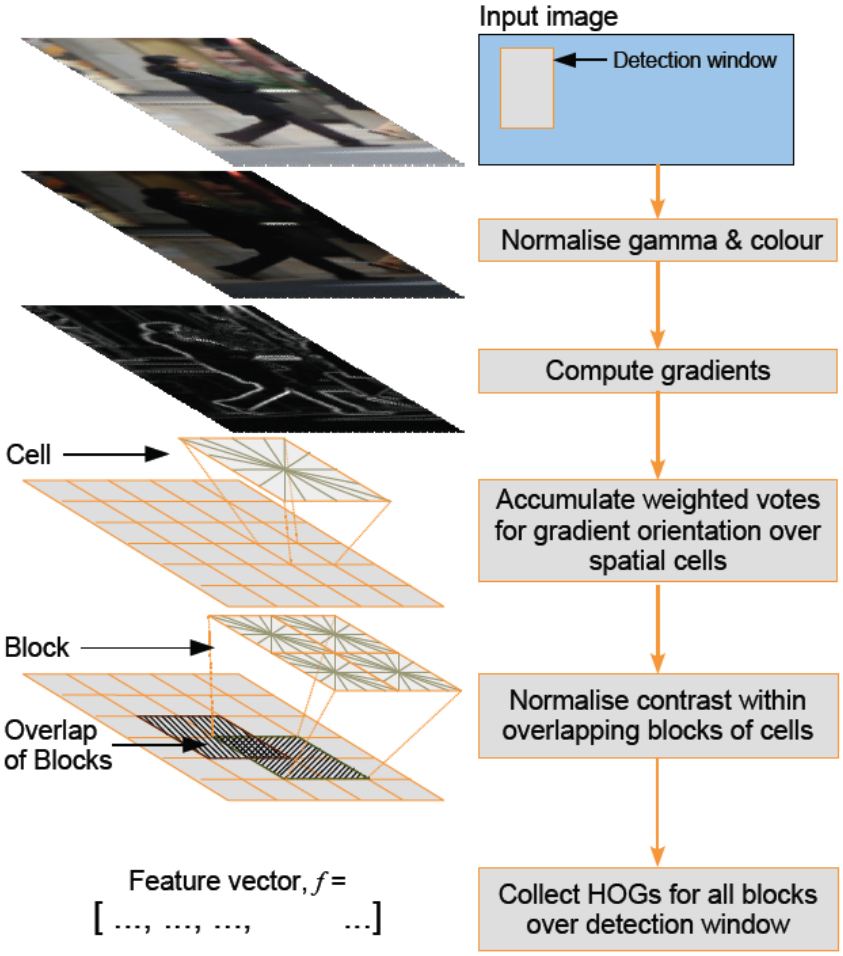


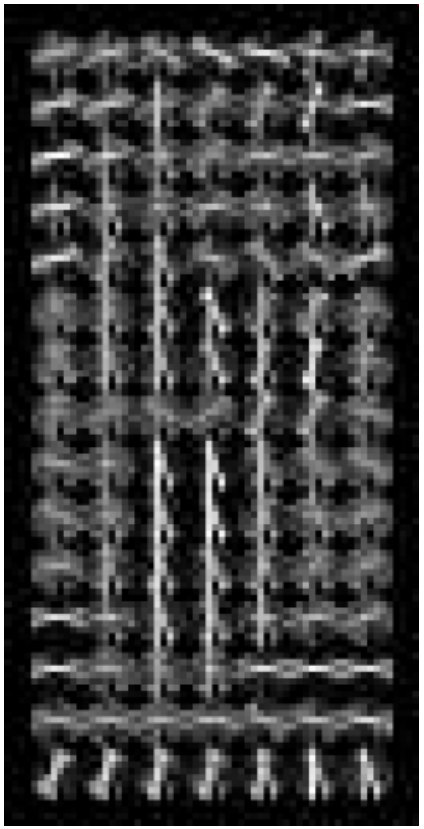
(a)

(b)



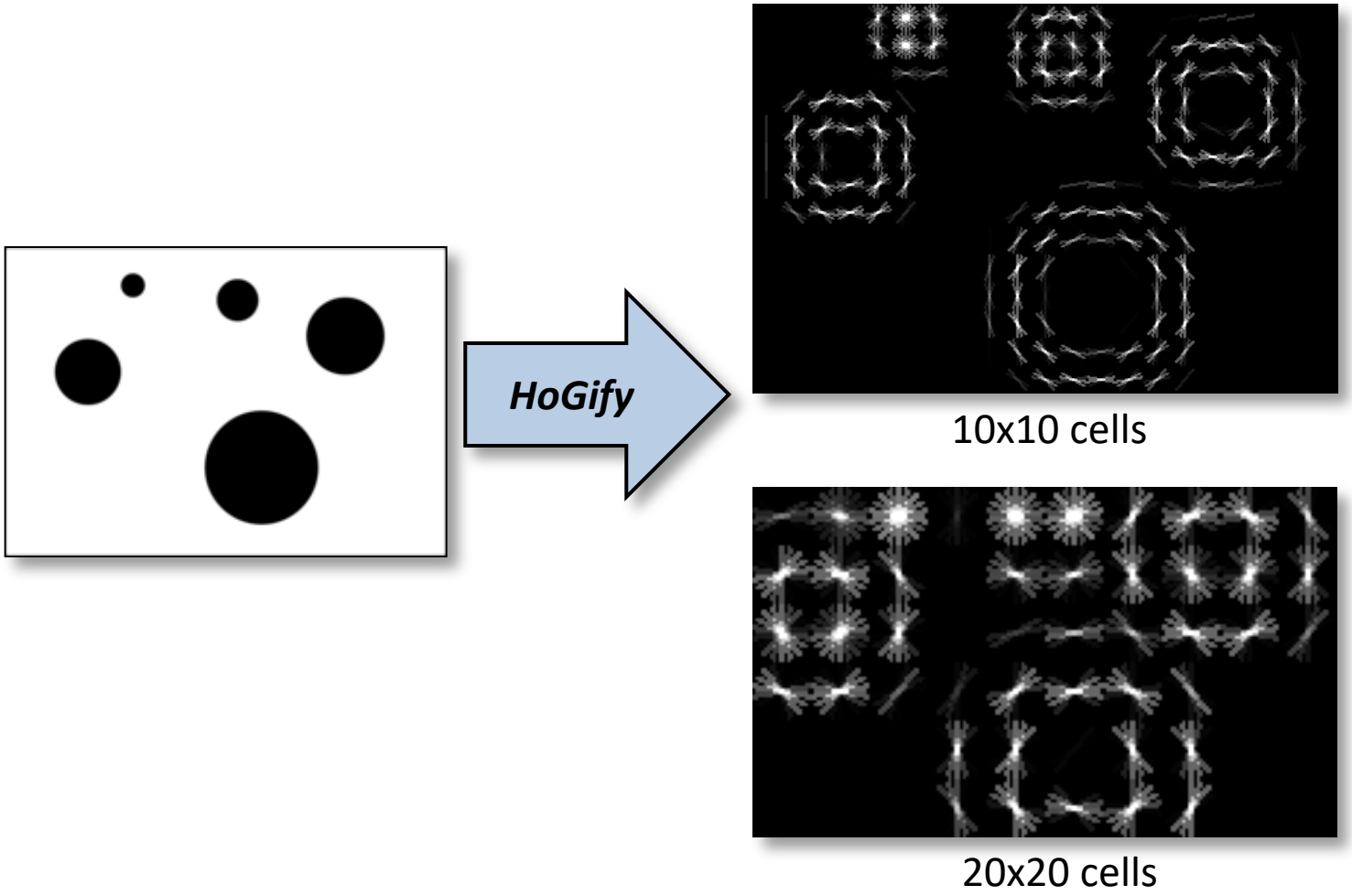
HOG





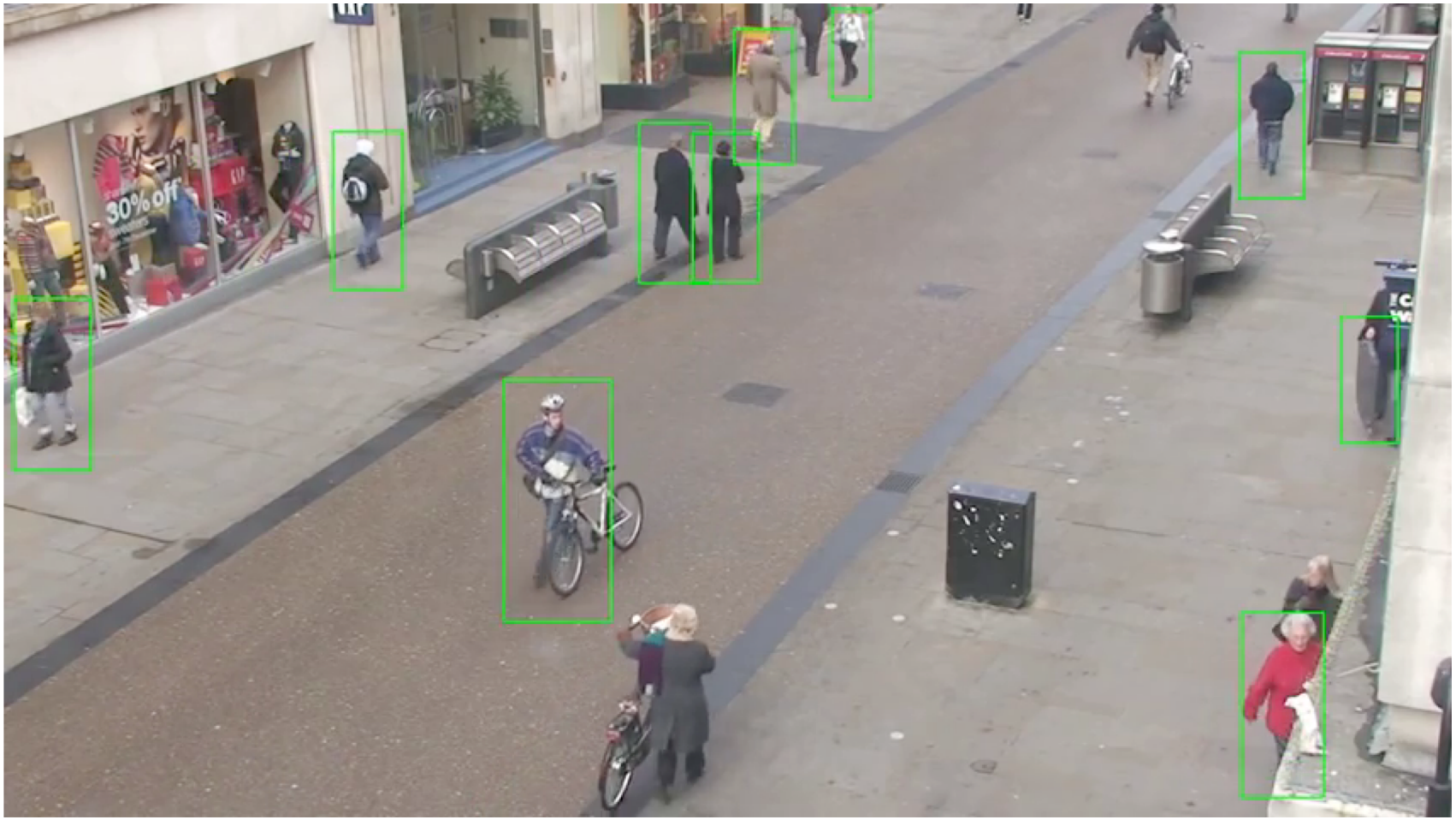
*Navneet Dalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, SanDiego, USA, June 2005. Vol. II, pp. 886-893.

Histogram of Oriented Gradients (HoG)



[Dalal and Triggs, CVPR 2005]

Results



Classification vs regression

- We are interested in mapping the input $\mathbf{x} \in \mathcal{X}$ to a *label* $t \in \mathcal{Y}$
- In regression typically $\mathcal{Y} = \Re$
- Now $\mathcal{Y}$ is categorical

Classification as regression

- Can we do this task using what we have learned in previous lectures?
- Simple hack: Ignore that the output is categorical!
- Suppose we have a binary problem, $t \in \{-1, 1\}$
- Assuming the standard model used for regression

$$y = f(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \mathbf{x}$$

- How can we obtain $\mathbf{w}$?
- Use least squares, $\mathbf{w} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{t}$. How is $\mathbf{X}$ computed? and $\mathbf{t}$?
- Which loss are we minimizing? Does it make sense?

$$\ell_{square}(\mathbf{w}, t) = \frac{1}{N} \sum_{n=1}^{N} (t^{(n)} - \mathbf{w}^T \mathbf{x}^{(n)})^2$$

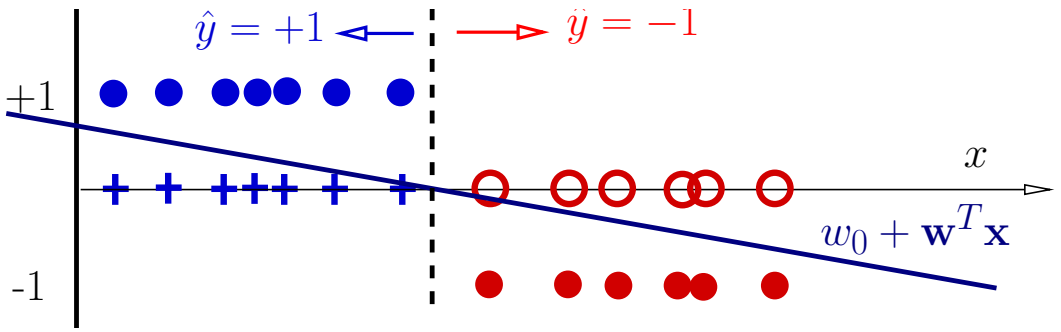
- How do I compute a label for a new example? Let's see an example

Classification as regression

- One dimensional example (input x is 1-dim)

- The colors indicate labels (a blue plus denotes that $t^{(i)}$ is from class -1, red circle that $t^{(i)}$ is class 1)

Decision Rule



- How can I mathematically write this rule?

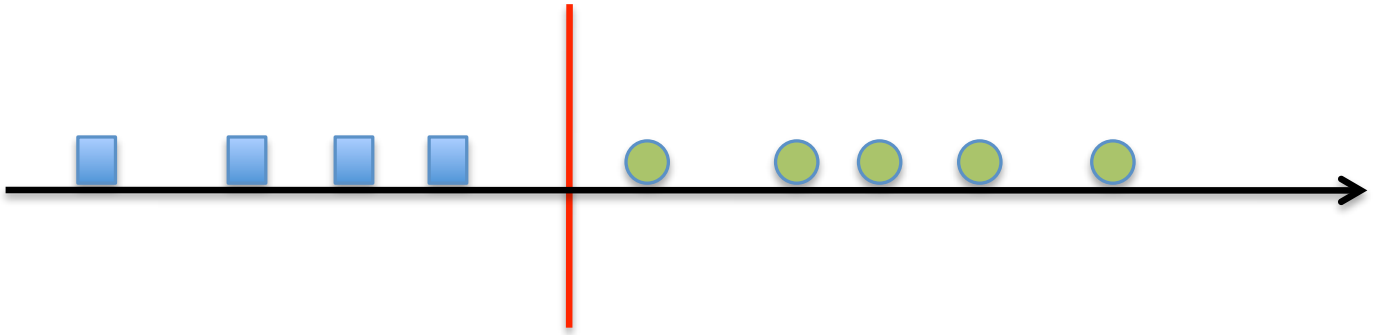
$$y = \text{sign}(w_0 + \mathbf{w}^T\mathbf{x})$$

- This specifies a linear classifier: it has a linear boundary (hyperplane)

$$w_0 + \mathbf{w}^T\mathbf{x} = 0$$

which separates the space into two "half-spaces"

Example in 1D

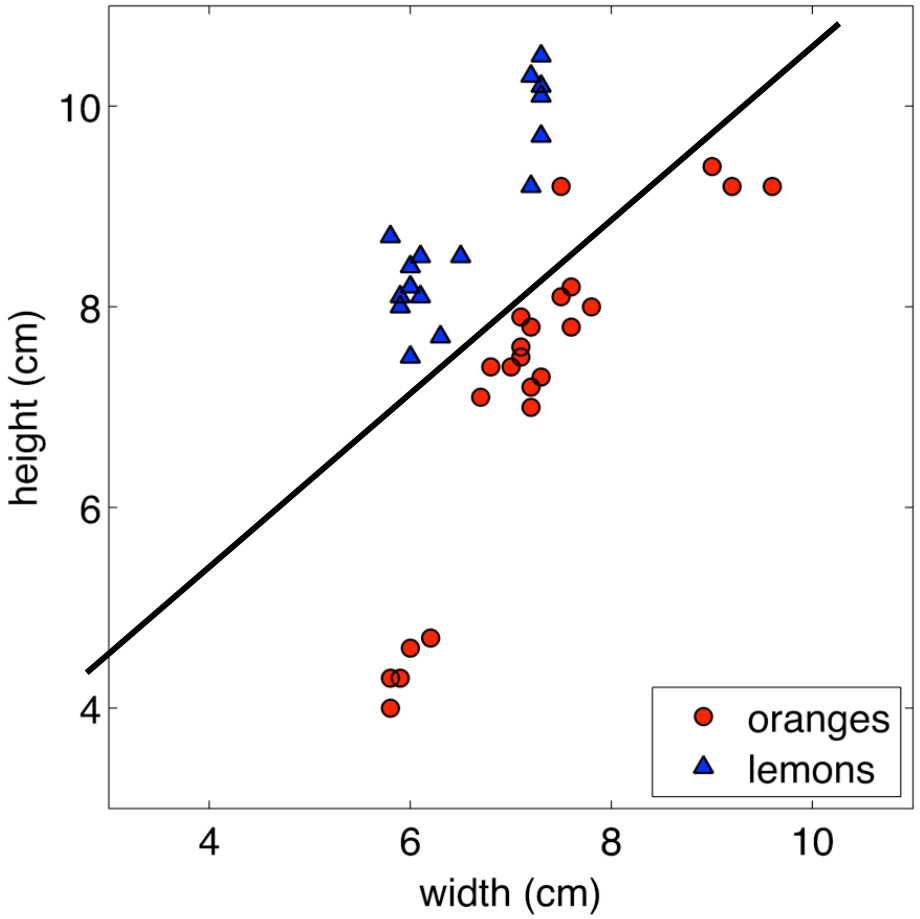


- The linear classifier has a linear boundary (hyperplane)

$$w_0 + \mathbf{w}^T\mathbf{x} = 0$$

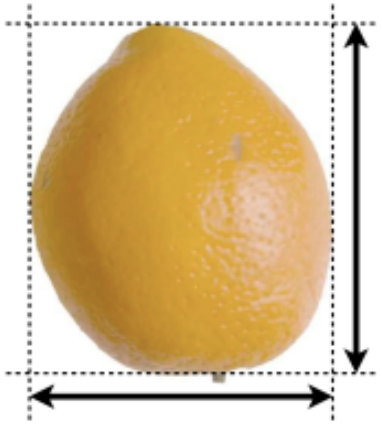
  which separates the space into two "half-spaces"
- In 1D this is simply a threshold

Classification Lemons and oranges

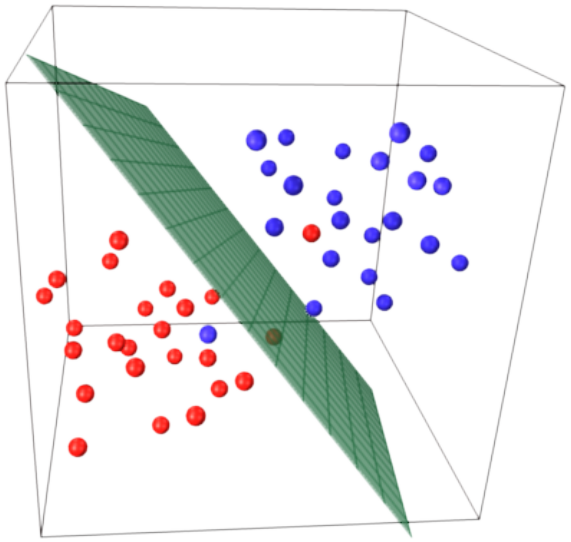


Can construct simple linear decision boundary:

$y = \text{sign}(w_0 + w_1 x_1 + w_2 x_2)$



Example in 3D



- The linear classifier has a linear boundary (hyperplane)

$$w_0 + \mathbf{w}^T \mathbf{x} = 0$$

  which separates the space into two "half-spaces"
- In 3D this is a plane
- What about higher-dimensional spaces?

Sometimes written in homogeneous form

$$\begin{bmatrix} \mathbf{w} & w_0 \end{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix} = 0$$

Geometry

$\mathbf{w}^T\mathbf{x} = 0$ a line passing though the origin and orthogonal to $\mathbf{w}$
$\mathbf{w}^T\mathbf{x} + w_0 = 0$ shifts it by w_0

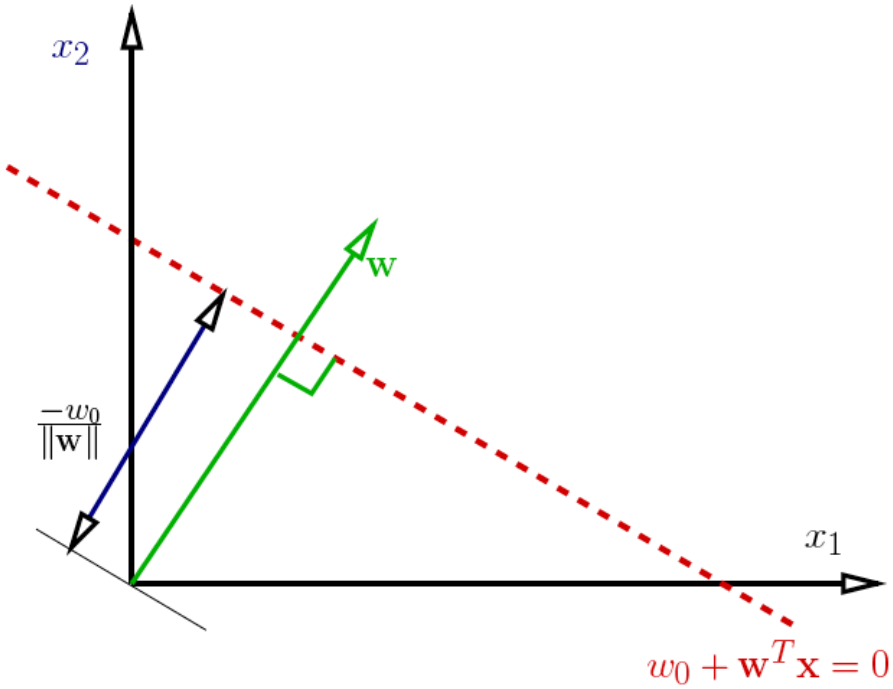


Figure from C. Shakhnarovich

Loss Functions

- Classifying using a linear decision boundary reduces the data dimension to 1

$$y(\mathbf{x}) = \text{sign}(w_0 + \mathbf{w}^T\mathbf{x})$$

- What is the cost of being wrong?
- Loss function: $L(y, t)$ is the loss incurred for predicting y when correct answer is t
- For medical diagnosis: For a diabetes screening test is it better to have false positives or false negatives?
- For movie ratings: The "truth" is that Alice thinks E.T. is worthy of a 4. How bad is it to predict a 5? How about a 2?

Other Loss Functions

- Zero/one loss for a classifier

$$L_{0-1}(y(\mathbf{x}), t) = \begin{cases} 0 & \text{if } y(\mathbf{x}) = t \\ 1 & \text{if } y(\mathbf{x}) \neq t \end{cases}$$

- Asymmetric Binary Loss

$$L_{ABL}(y(\mathbf{x}), t) = \begin{cases} \alpha & \text{if } y(\mathbf{x}) = 1 \wedge t = 0 \\ \beta & \text{if } y(\mathbf{x}) = 0 \wedge t = 1 \\ 0 & \text{if } y(\mathbf{x}) = t \end{cases}$$

- Squared (quadratic) loss

$$L_{squared}(y(\mathbf{x}), t) = (t - y(\mathbf{x}))^2$$

- Absolute Error

$$L_{absolute}(y(\mathbf{x}), t) = |t - y(\mathbf{x})|$$

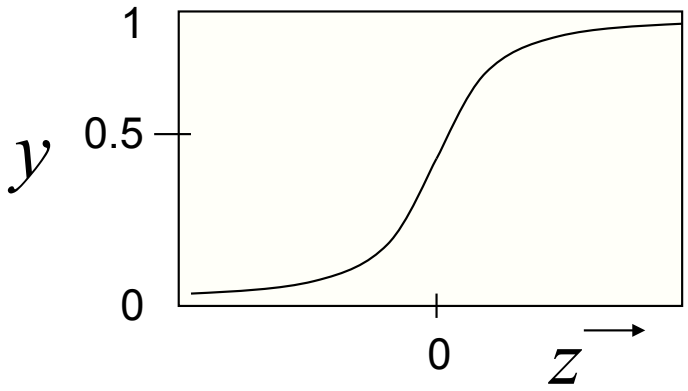
Logistic Regression

- An alternative: replace the $sign(\cdot)$ with the sigmoid or logistic function
- We assumed a particular functional form: sigmoid applied to a linear function of the data

$$y(\mathbf{x}) = \sigma\left(\mathbf{w}^T\mathbf{x} + w_0\right)$$

where the sigmoid is defined as

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$



- The output is a smooth function of the inputs and the weights. It can be seen as a smoothed and differentiable alternative to $sign(\cdot)$

Logistic regression

- We assumed a particular functional form: sigmoid applied to a linear function of the data

$$y(\mathbf{x}) = \sigma\left(\mathbf{w}^T\mathbf{x} + w_0\right)$$

  where the sigmoid is defined as

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

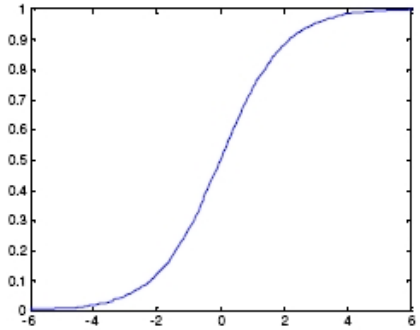
  - One parameter per data dimension (feature) and the bias
  - Features can be discrete or continuous
  - Output of the model: value $y \in [0, 1]$
  - Allows for gradient-based learning of the parameters

Shape of the logistic function

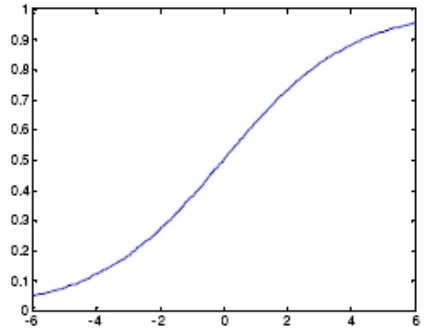
- Let's look at how modifying **w** changes the shape of the function
- 1D example:

$$y = \sigma(w_1 x + w_0)$$

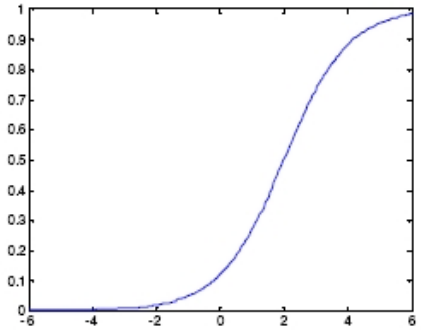
$w_0 = 0, w_1 = 1$



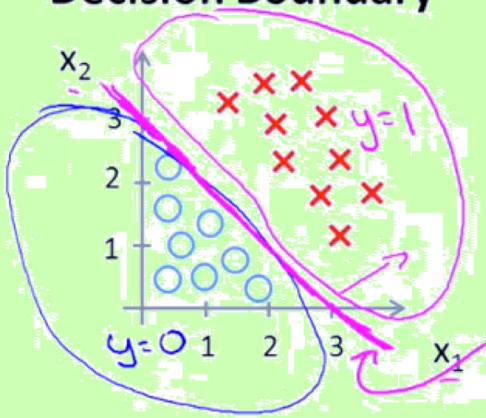
$w_0 = 0, w_1 = 0.5$



$w_0 = -2, w_1 = 1$



Decision Boundary



$$\theta = \begin{bmatrix} -3 \\ 1 \\ 1 \end{bmatrix}$$

$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

-3 $\quad 1 \quad 1$

Decision boundary

Predict "$y = 1$" if $-3 + x_1 + x_2 \geq 0$

$\theta^T x$

$x_1 + x_2 \geq 3$

x_1, x_2

$h_\theta(x) = 0.5$

$x_1 + x_2 = 3$

$x_1 + x_2 < 3$

$y = 0$

Higher order terms??

Non-linear decision boundaries

$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2)$$

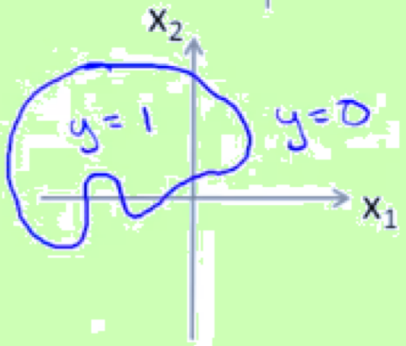
$\theta_0 = -1$, $\theta_1 = 0$, $\theta_2 = 0$, $\theta_3 = 1$, $\theta_4 = 1$

$$\theta = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

Predict "$y = 1$" if $-1 + x_1^2 + x_2^2 \geq 0$

$x_1^2 + x_2^2 = 1$

$x_1^2 + x_2^2 \geq 1$



$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_1^2 x_2 + \theta_5 x_1^2 x_2^2 + \theta_6 x_1^3 x_2 + \dots)$$

Change feature vector

Andrew N

Probabilistic interpretation

- If we have a value between 0 and 1, let's use it to model class probability

$$p(C = 0|\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x} + w_0) \quad \text{with} \quad \sigma(z) = \frac{1}{1 + \exp(-z)}$$

- Substituting we have

$$p(C = 0|\mathbf{x}) = \frac{1}{1 + \exp\left(-\mathbf{w}^T\mathbf{x} - w_0\right)}$$

- Suppose we have two classes, how can I compute $p(C = 1|\mathbf{x})$?
- Use the marginalization property of probability

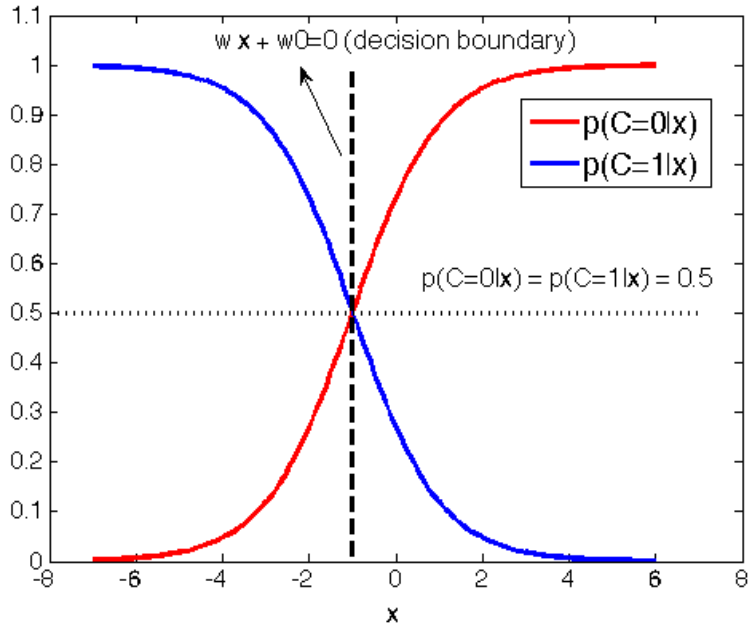
$$p(C = 1|\mathbf{x}) + p(C = 0|\mathbf{x}) = 1$$

- Thus

$$p(C = 1|\mathbf{x}) = 1 - \frac{1}{1 + \exp\left(-\mathbf{w}^T\mathbf{x} - w_0\right)} = \frac{\exp(-\mathbf{w}^T\mathbf{x} - w_0)}{1 + \exp\left(-\mathbf{w}^T\mathbf{x} - w_0\right)}$$

Decision Boundary for Logistic Regression

- What is the decision boundary for logistic regression?
- $p(C = 1|\mathbf{x}, \mathbf{w}) = p(C = 0|\mathbf{x}, \mathbf{w}) = 0.5$
- $p(C = 0|\mathbf{x}, \mathbf{w}) = \sigma\left(\mathbf{w}^T\mathbf{x} + w_0\right) = 0.5$, where $\sigma(z) = \frac{1}{1+\exp(-z)}$
- Decision boundary: $\mathbf{w}^T\mathbf{x} + w_0 = 0$
- Logistic regression has a **linear decision boundary**

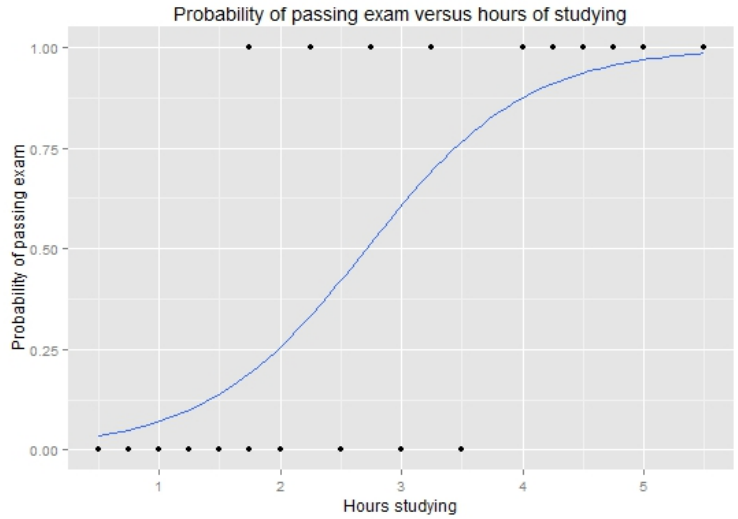


Example

- **Problem**: Given the number of hours a student spent learning, will (s)he pass the exam?
- Training data (top row: $\mathbf{x}^{(i)}$, bottom row: $t^{(i)}$)

Hours	0.50	0.75	1.00	1.25	1.50	1.75	1.75	2.00	2.25	2.50	2.75	3.00	3.25	3.50	4.00	4.25	4.50	4.75	5.00	5.50
Pass	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1

- Learn **w** for our model, i.e. logistic regression (coming up)
- Make predictions:



Hours of study	Probability of passing exam
1	0.07
2	0.26
3	0.61
4	0.87
5	0.97

Fitting parameters to logistic model

Training set: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \cdots, (x^{(m)}, y^{(m)})\}$

m examples $\quad x \in \begin{bmatrix} x_0 \\ x_1 \\ \cdots \\ x_n \end{bmatrix} \quad x_0 = 1, y \in \{0, 1\}$

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

How to choose parameters θ ?

Cost function

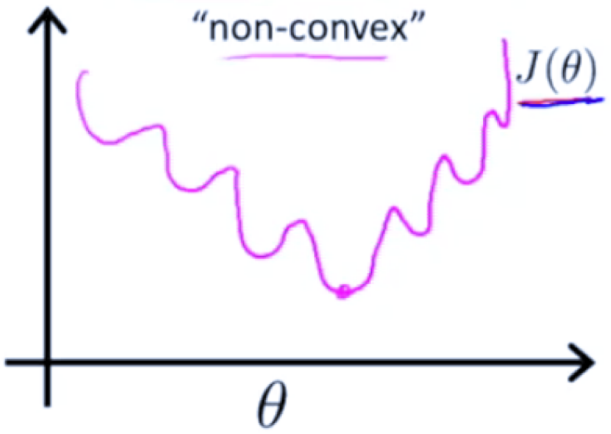
→ ~~Linear~~ regression: $J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{2} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$

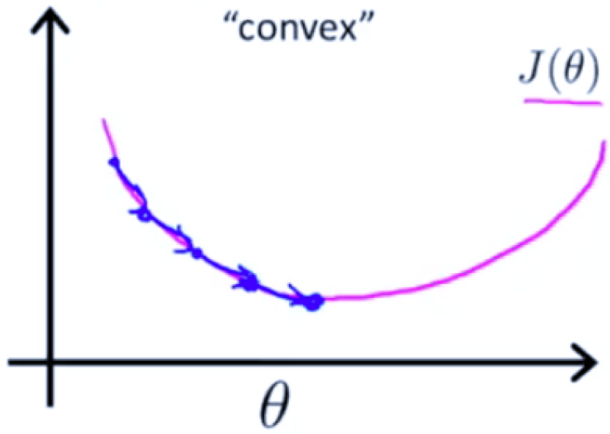
Logistic

$\rightarrow \text{cost}(h_\theta(x^{(i)}), y)$

$\rightarrow \text{Cost}(h_\theta(x^{(i)}), y^{(i)}) = \frac{1}{2} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$ ←

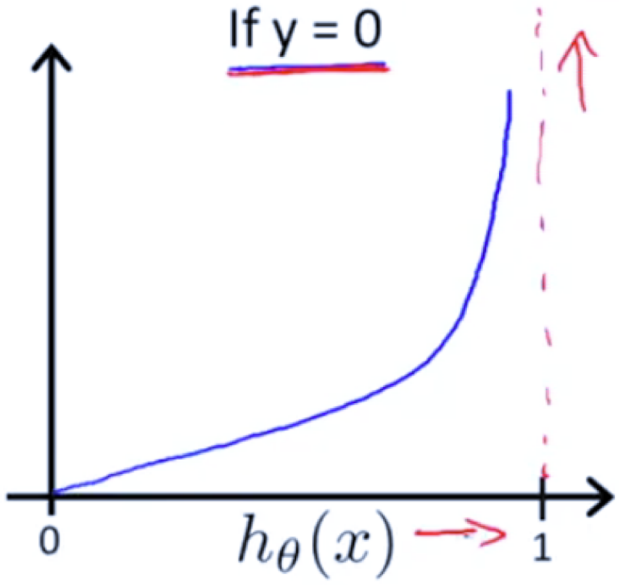
$\frac{1}{1 + e^{-\theta^T x}}$





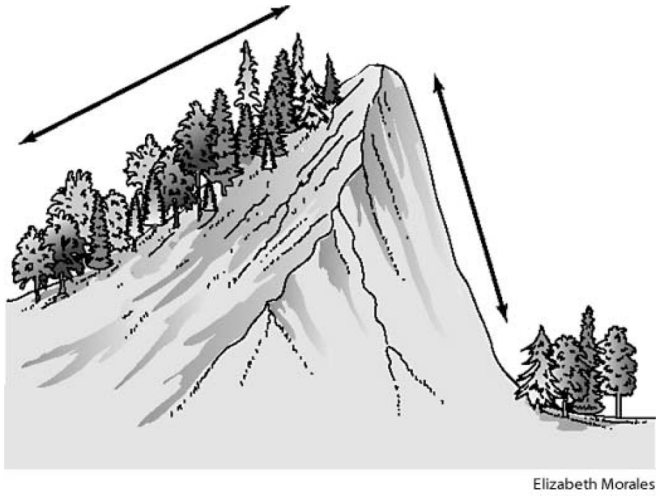
Logistic regression cost function

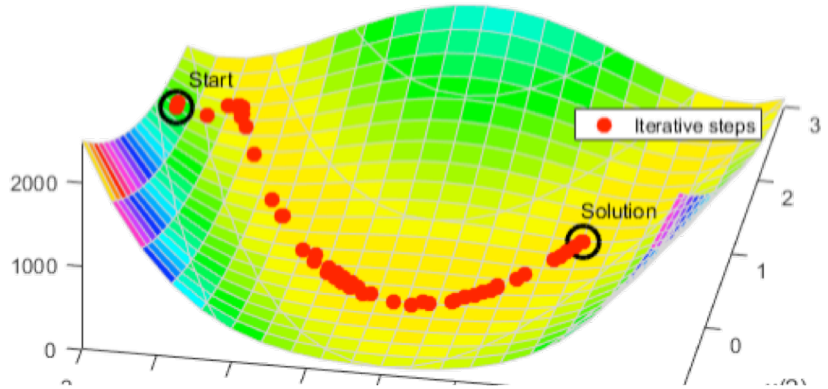
$$\text{Cost}(h_\theta(x^{(i)}, y^{(i)}) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$



$-\log(1-z)$

A few notes on non-linear parameter learning



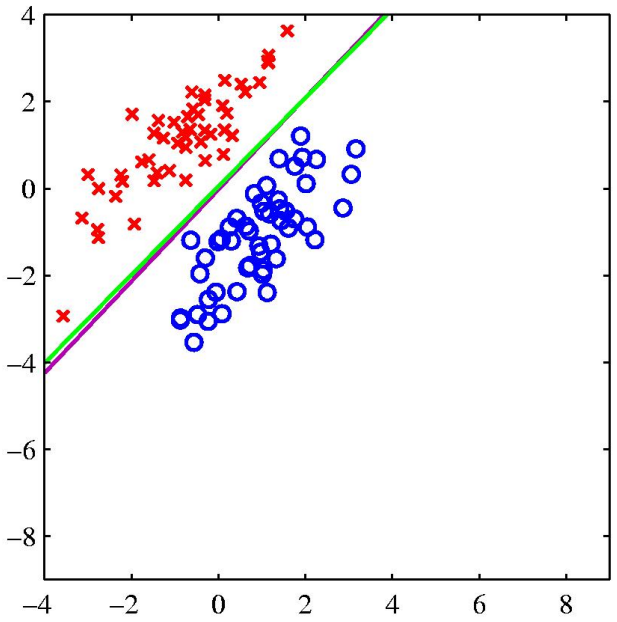


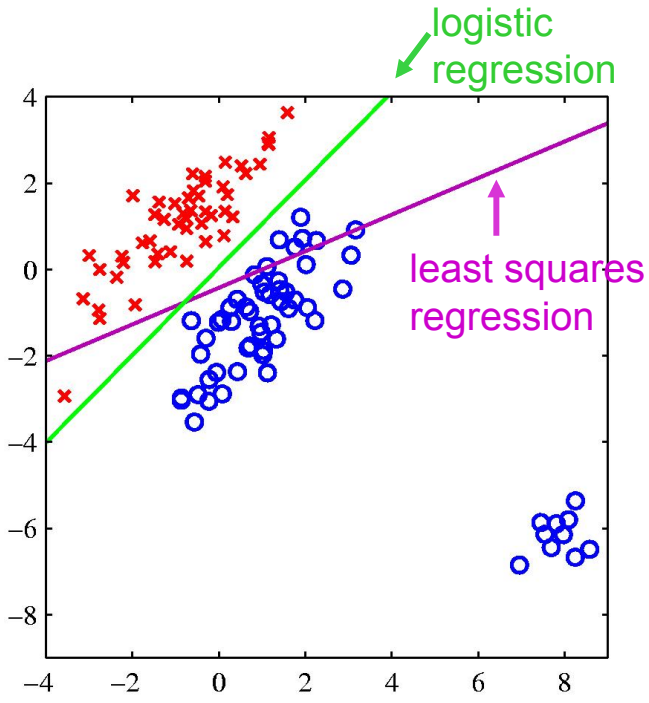
Repeat until convergence {

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

}

Logistic Regression vs Least Squares regression





If the right answer is 1 and the model says 1.5, it loses, so it changes the boundary to avoid being "too correct" (tilts away from outliers)

33

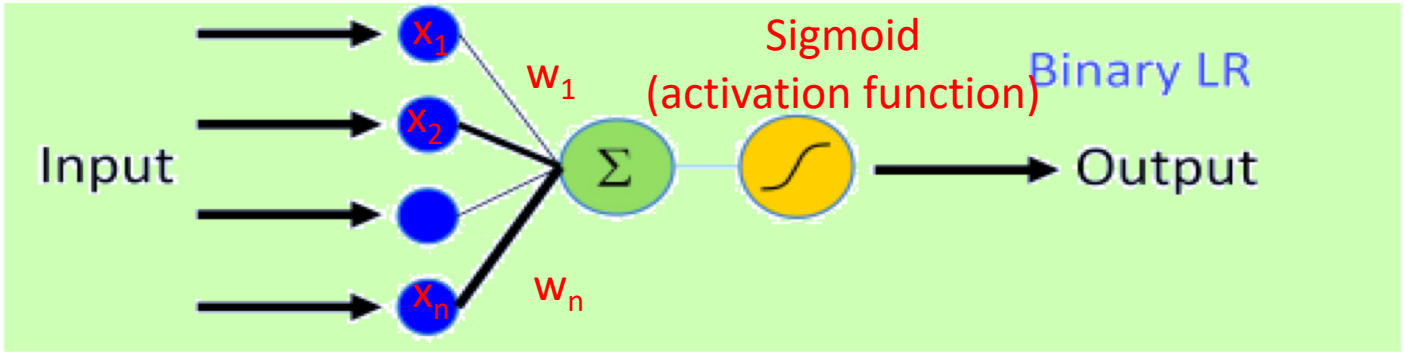
One way to think about it…

- Training labels dictate that two examples are the same or different (in some sense).
- Features and distance measures define visual similarity
- Classifiers try to learn weights or parameters for features and distance measures so that visual similarity predicts label similarity

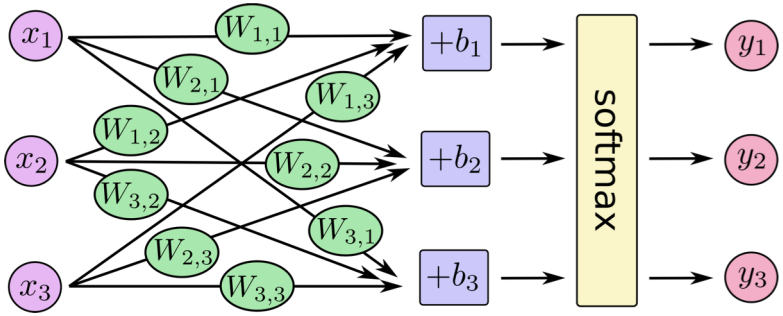
Slide: Derek Hoiem

Visualizing logistic regression

Binary classification

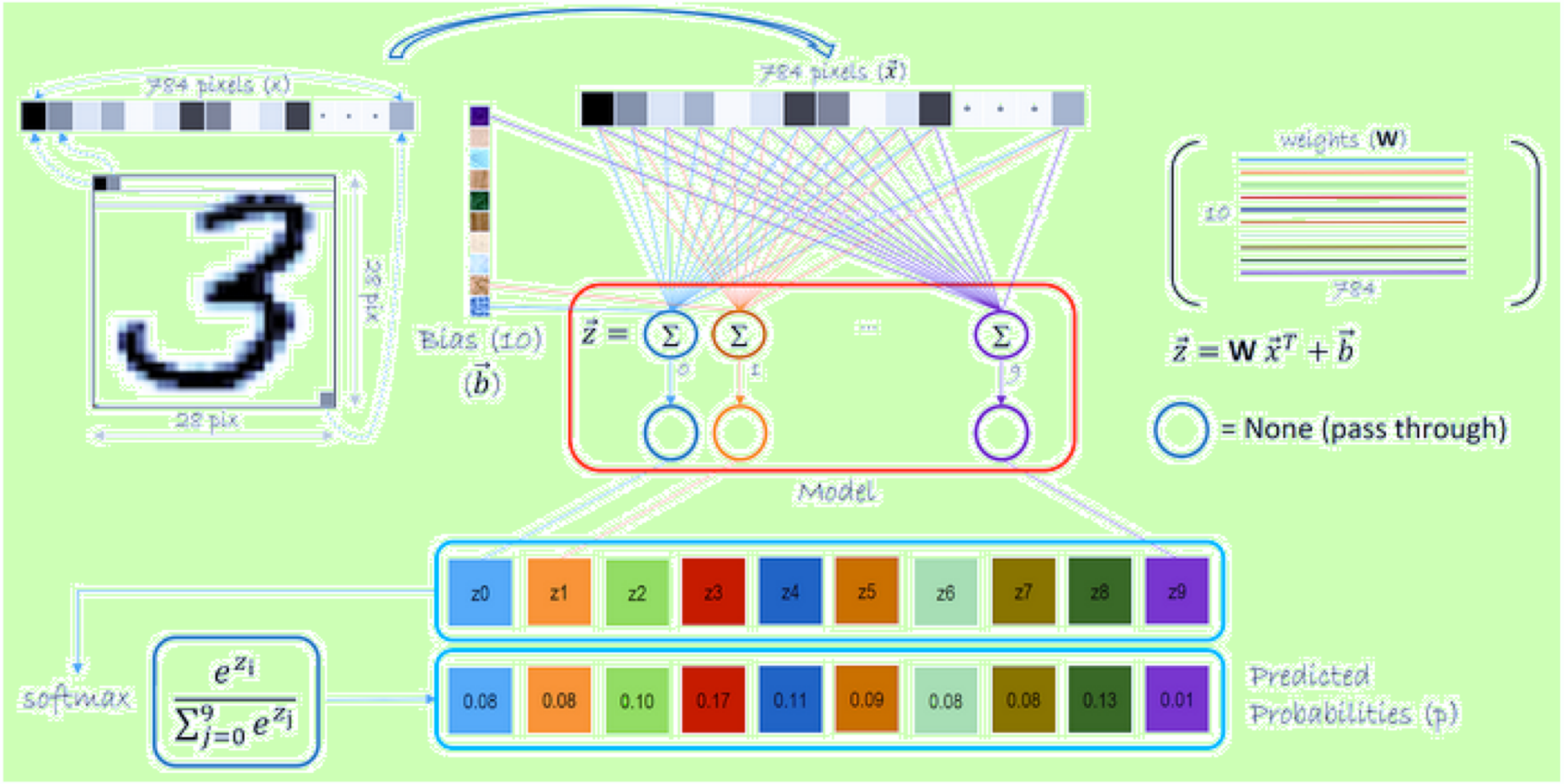


Multi class classification

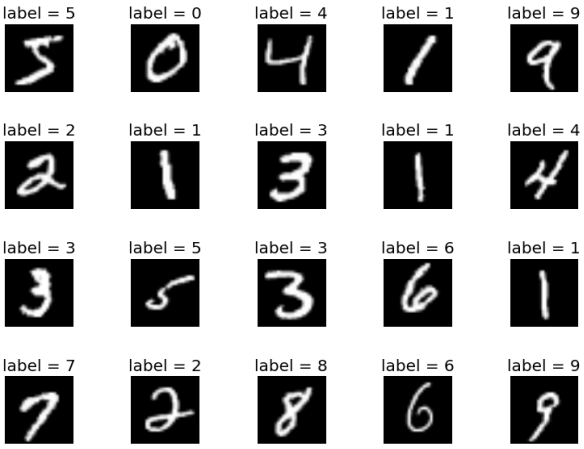


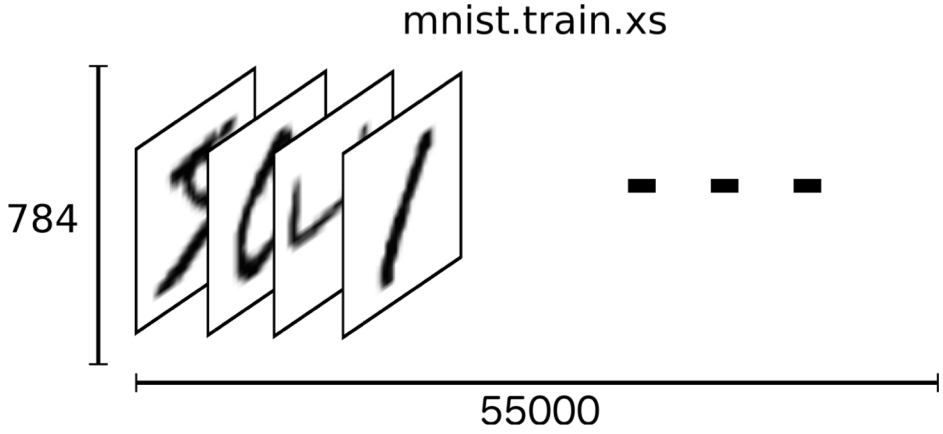
$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \text{softmax}\left(\begin{matrix} W_{1,1}x_1 + W_{1,2}x_2 + W_{1,3}x_3 + b_1 \\ W_{2,1}x_1 + W_{2,2}x_2 + W_{2,3}x_3 + b_2 \\ W_{3,1}x_1 + W_{3,2}x_2 + W_{3,3}x_3 + b_3 \end{matrix} \right)$$

$$P(y = j \mid \mathbf{x}) = \frac{e^{\mathbf{x}^\top \mathbf{w}_j}}{\sum_{k=1}^{K} e^{\mathbf{x}^\top \mathbf{w}_k}}$$


784 pixels (x)
28 pix
28 pix
784 pixels ($\vec{x}$)
Bias (10)
($\vec{b}$)
$\vec{z}$ =
Σ
Σ
Σ
0
1
9
Model
weights (W)
10
784
$\vec{z} = \mathbf{W}\,\vec{x}^{T} + \vec{b}$
= None (pass through)
z0
z1
z2
z3
z4
z5
z6
z7
z8
z9
0.08
0.08
0.10
0.17
0.11
0.09
0.08
0.08
0.13
0.01
Predicted Probabilities (p)
softmax
$\frac{e^{z_i}}{\sum_{j=0}^{9} e^{z_j}}$

MNIST


label = 5
label = 0
label = 4
label = 1
label = 9
label = 2
label = 1
label = 3
label = 1
label = 4
label = 3
label = 5
label = 3
label = 6
label = 1
label = 7
label = 2
label = 8
label = 6
label = 9


mnist.train.xs
784
55000

Logistic regression using keras

```
def build_logistic_model(input_dim, output_dim):
    model = Sequential()
    model.add(Dense(output_dim, input_dim=input_dim, activation='softmax'))

    return model
```

Can Classes Always be Separated

Causes of non perfect separation:

- Model is too simple
- Noise in the inputs (i.e., data attributes)
- Simple features that do not account for all variations
- Errors in data targets (miss labelings)

Should we make the model complex enough to have perfect separation in the training data?

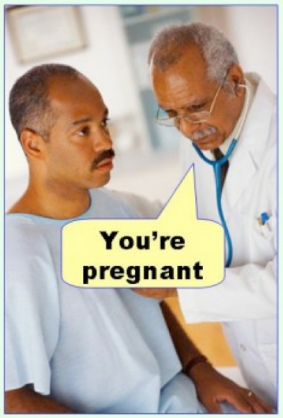
Metrics

How to evaluate how good my classifier is? How is it doing on dog vs no-dog?

TP (True Positive) FP (False Positive) FN (False Negative)

Measured / Perceived	Reality: True	Reality: False
True	Correct ☺	Type I False Positive
False	Type II False Negative	Correct ☺

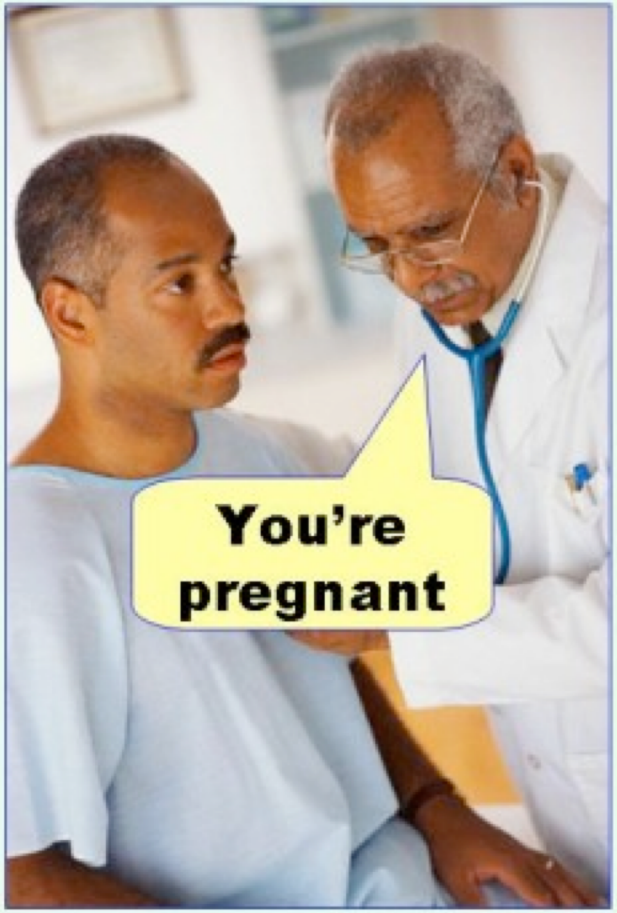
Type I error
(false positive)



Type II error
(false negative)

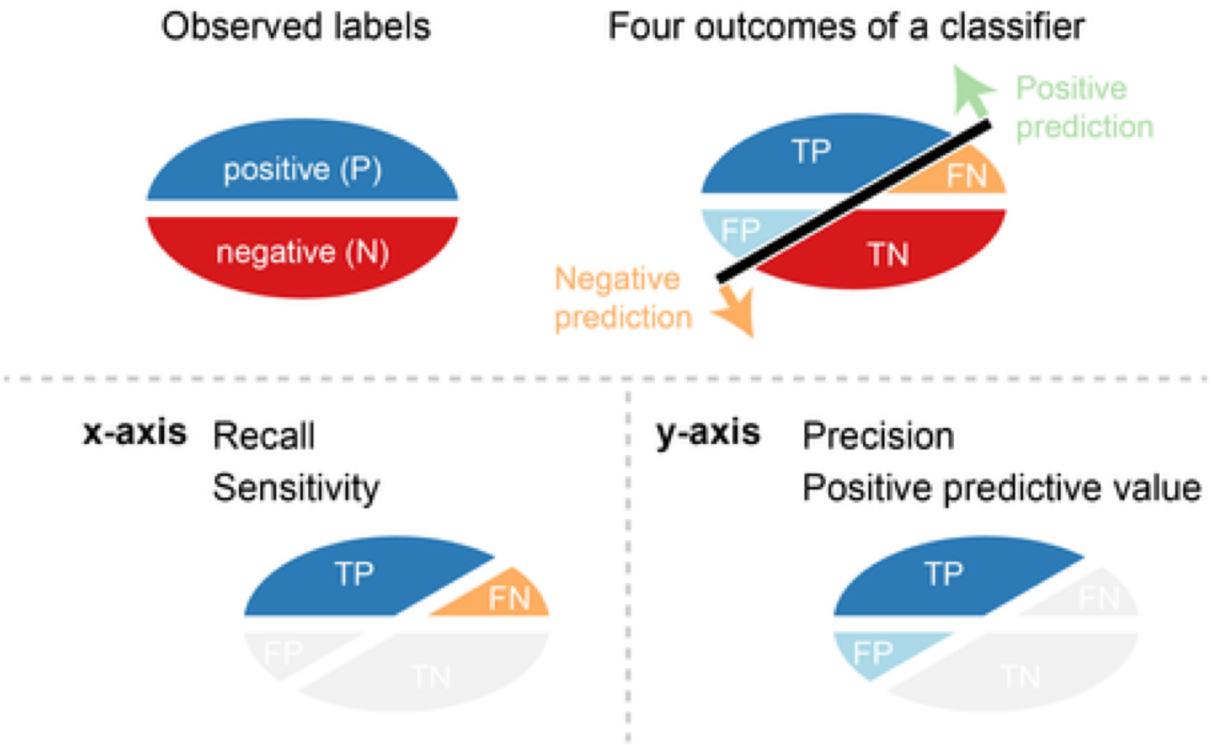
Type I error

(false positive)



Type II error

(false negative)


Observed labels
positive (P)
negative (N)
Four outcomes of a classifier
Positive prediction
TP
FN
FP
TN
Negative prediction
x-axis
Recall
Sensitivity
TP
FN
FP
TN
y-axis
Precision
Positive predictive value
TP
FN
FP
TN

Metrics

How to evaluate how good my classifier is?

- Recall: is the fraction of relevant instances that are retrieved

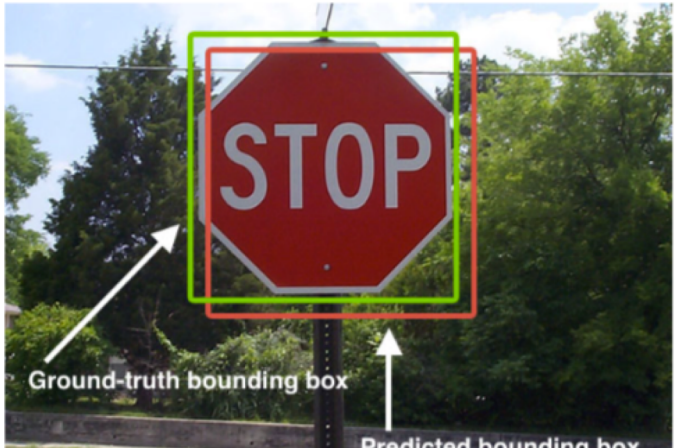
$$R = \frac{TP}{TP + FN} = \frac{TP}{\text{all groundtruth instances}}$$

- Precision: is the fraction of retrieved instances that are relevant

$$P = \frac{TP}{TP + FP} = \frac{TP}{\text{all predicted}}$$

- F1 score: harmonic mean of prec

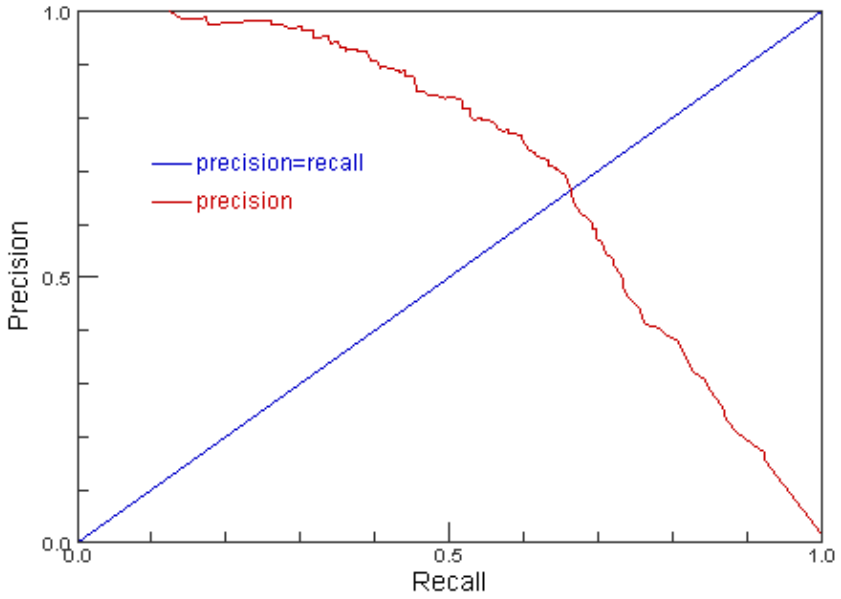
$$\text{IoU} = \frac{\text{area}(B_p \cap B_{gt})}{\text{area}(B_p \cup B_{gt})}$$



More on metrics

How to evaluate how good my classifier is?

- Precision: is the fraction of retrieved instances that are relevant
- Recall: is the fraction of relevant instances that are retrieved
- Precision Recall Curve



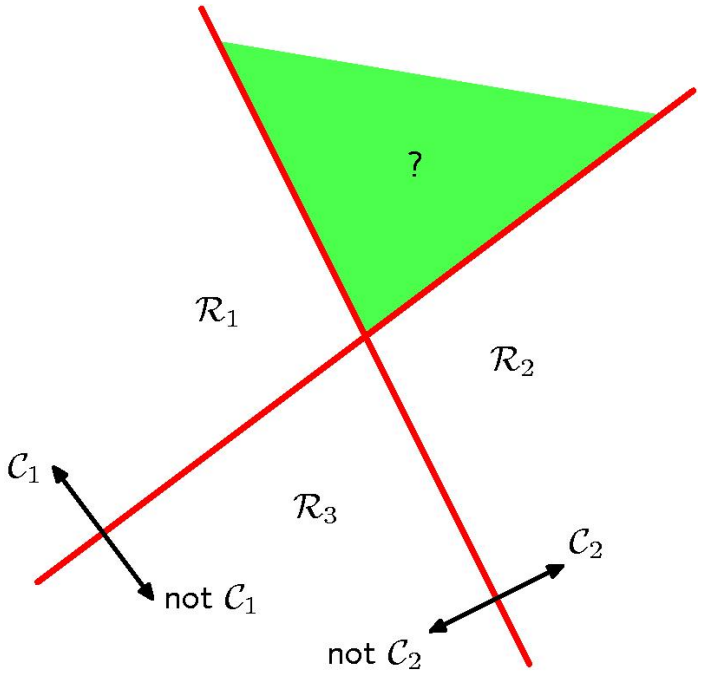
- Average Precision (AP): mean under the curve

Multi-class classification

https://www.kickstarter.com/projects/adrianrosebrock/deep-learning-for-computer-vision-with-python-eboo/posts/1796832

Idea of discriminant functions K>2

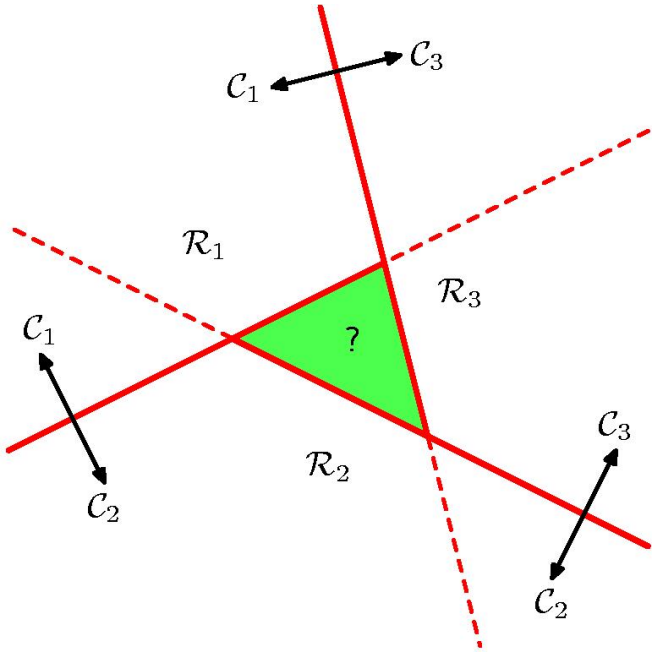
- First idea: Use $K-1$ classifiers, each solving a two class problem of separating point in a class C_k from points not in the class.
- Known as **1 vs all** or **1 vs the rest** classifier



- PROBLEM: More than one good answer for green region!

Idea of discriminant functions K>2

- Another simple idea: Introduce $K(K-1)/2$ two-way classifiers, one for each possible pair of classes
- Each point is classified according to majority vote amongst the disc. func.
- Known as the **1 vs 1 classifier**



- PROBLEM: Two-way preferences need not be transitive

Next week

- Non-linear models
  - Neural networks
  - SVM
  - Kernels

  - WATCH THE VIDEOS BEFORE CLASS

FOR NN

- https://github.com/Hvass-Labs/TensorFlow-Tutorials/blob/master/02_Convolutional_Neural_Network.ipynb