# Introduction to Computer Vision

## *4. Classification of Visual Features*

UCLA – CS 188 – Fall 2019

Fabien Scalzo, Ph.D.

| | | | | |
|---|---|---|---|---|
| **Week 1** | | | 26-Sep | Introduction |
| **Week 2** | 1-Oct | Basic Image Processing | 3-Oct | Feature Extraction and Classification |
| **Week 3** | 8-Oct | **Classification of Visual Features** | 10-Oct | 2D Image transformations, RANSAC |
| **Week 4** | 15-Oct | SVD, 2D camera model, projective plane | 17-Oct | Euclidean geometry, rigid body motion |
| **Week 5** | 22-Oct | Epipolar Geometry | 24-Oct | 3D Cameras and processing |
| **Week 6** | 29-Oct | Midterm | 31-Oct | Statistical decision theory/Pattern Recognition |
| **Week 7** | 5-Nov | Deep Learning | 7-Nov | Deep Learning for Image Classification |
| **Week 8** | 12-Nov | Object Detection | 14-Nov | Generative models |
| **Week 9** | 19-Nov | Medical Imaging | 21-Nov | Autonomous Navigation |
| **Week 10** | 26-Nov | Guest Lecture (Nikhil Naik) | 28-Nov | |
| **Week 11** | 3-Dec | Recursive 3D reconstruction and pose estimation | 5-Dec | Recap |
| **Week 12** | 10-Dec | | 12-Dec | Final |

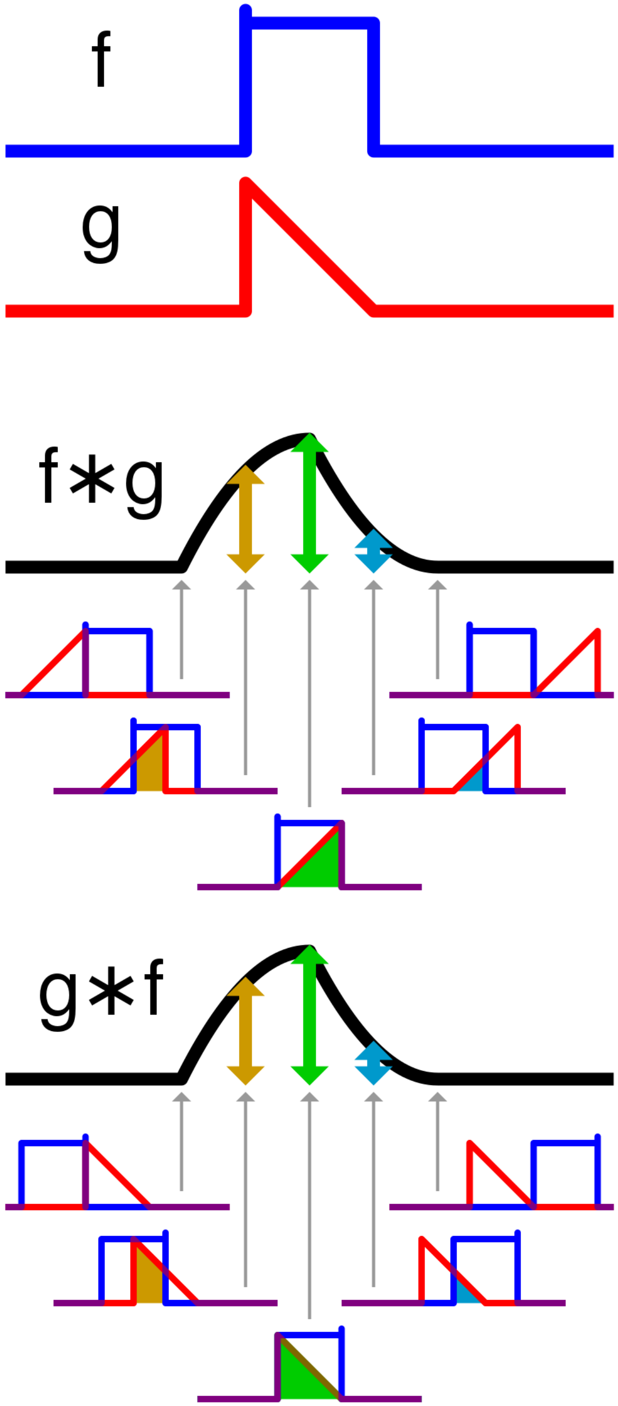
- If $f$ and $g$ are defined over integers $\mathbb{Z}$ (e.g. a 1D raster image), their *discrete convolution* is

$$[f * g](n) = \sum_{i=-\infty}^{\infty} f(i)\, g(n-i) = \sum_{i=-\infty}^{\infty} g(i)\, f(n-i)$$

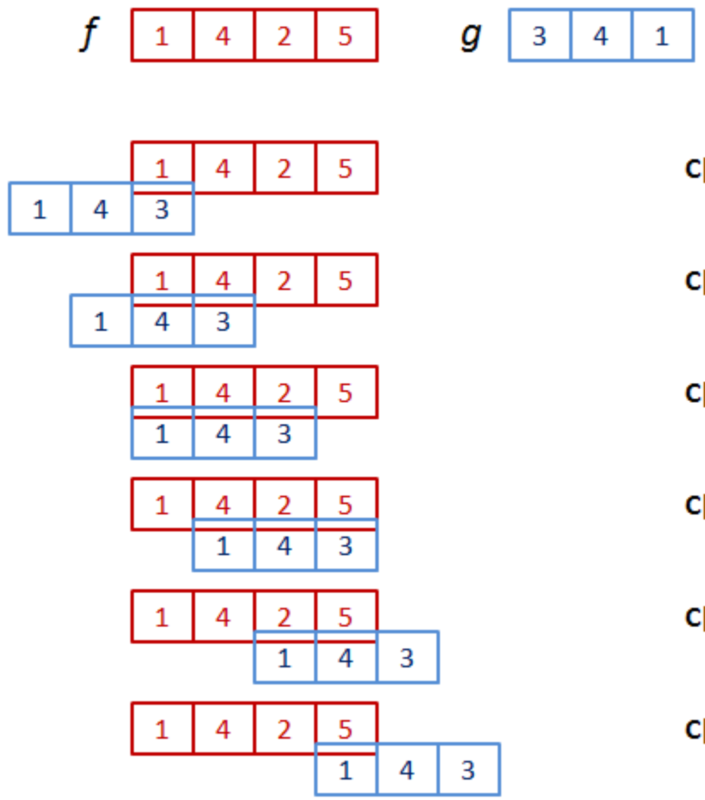
- Intuition:
  - Center the *kernel/filter function* $g$ at the $n^{\text{th}}$ pixel
  - Weight every pixel in the image by the value of $g$ there
  - Add up the weighted values to get the new color at the $n^{\text{th}}$ pixel

19

# Convolution



# Convolution



$$c = f * g$$

**C[0]** = 1*3 = **3**

**C[1]** = 1*4 + 4*3 = **16**

**C[2]** = 1*1 + 4*4 + 2*3 = **23**

**C[3]** = 4*1 + 2*4 + 5*3 = **27**

**C[4]** = 2*1 + 5*4 = **22**

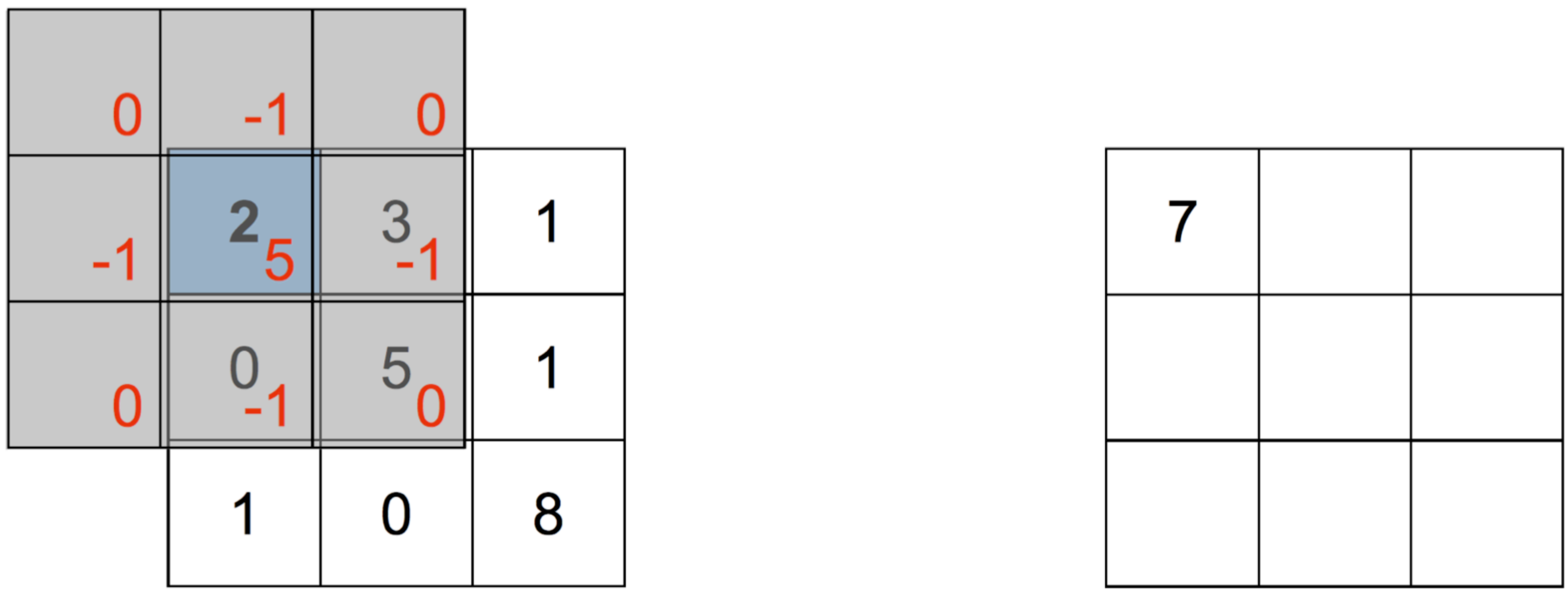
**C[5]** = 5*1 = **5**

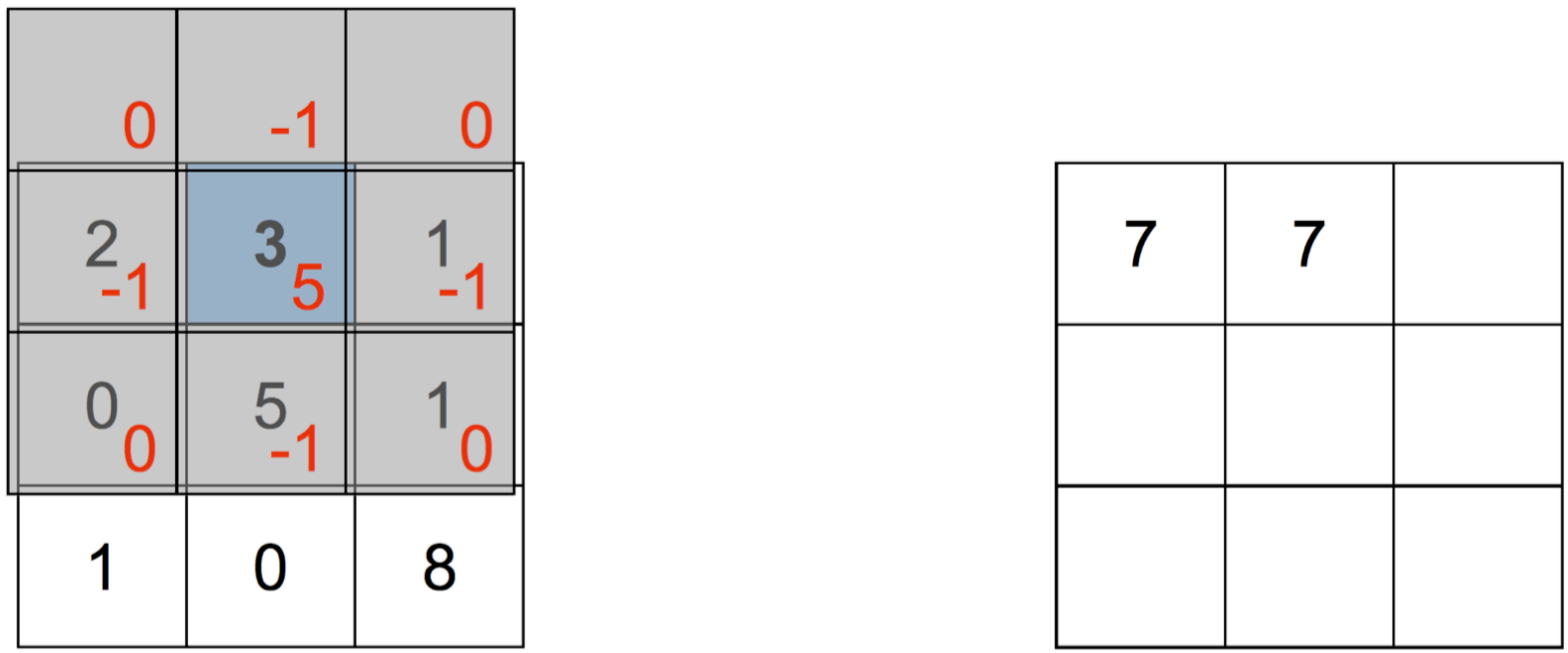
http://toto-share.com

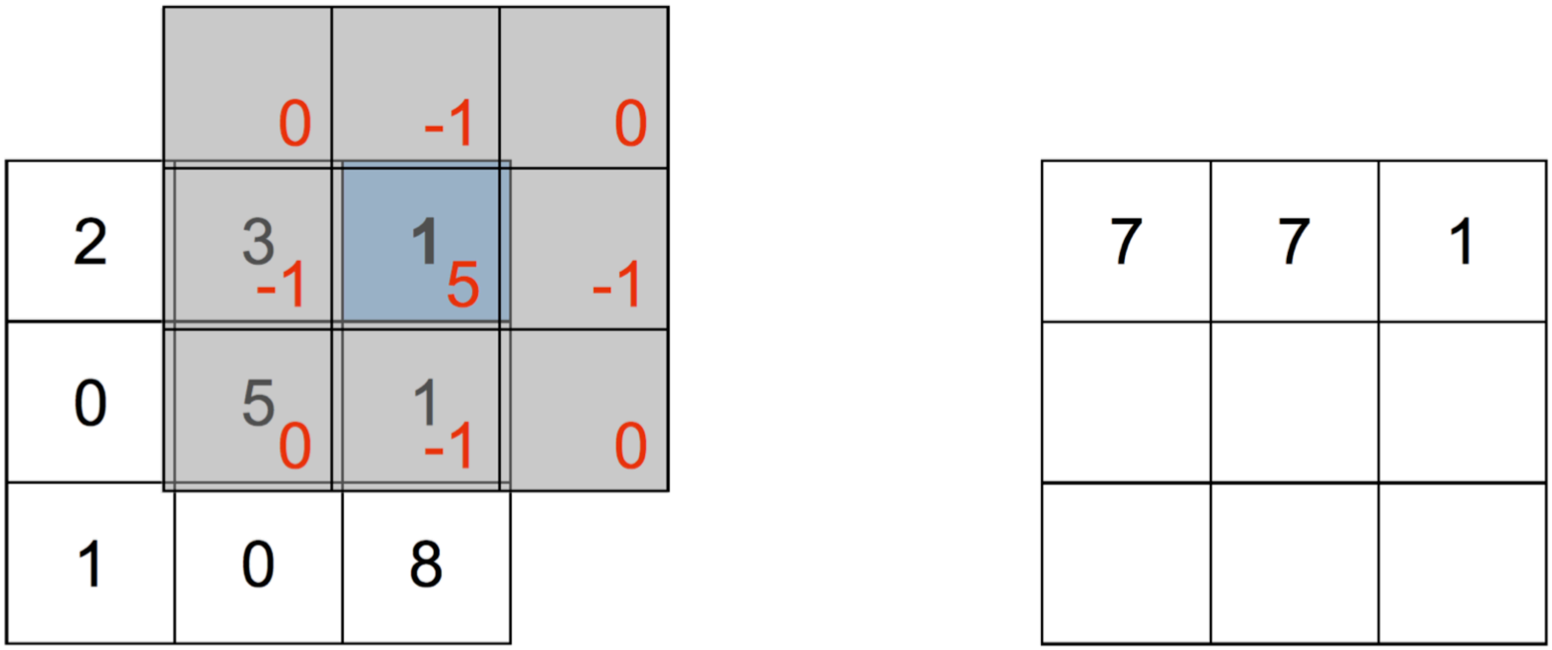
# 2D

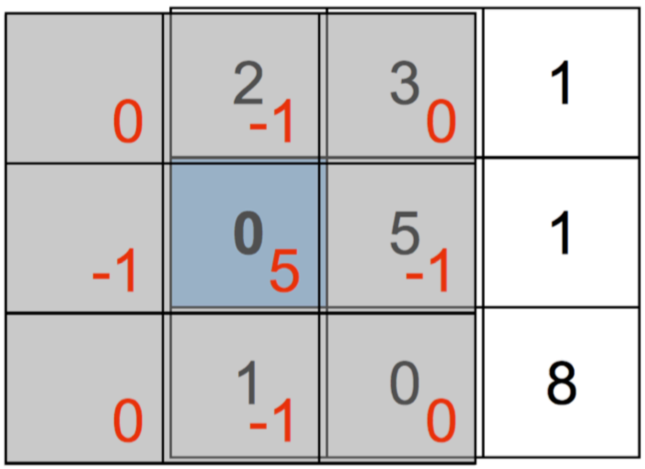
$$[f*g](m,n) = \sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty} f(i,j)\,g(m-i,n-j)$$

$$\begin{bmatrix} 2 & 3 & 1 \\ 0 & 5 & 1 \\ 1 & 0 & 8 \end{bmatrix} * \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} = ?$$

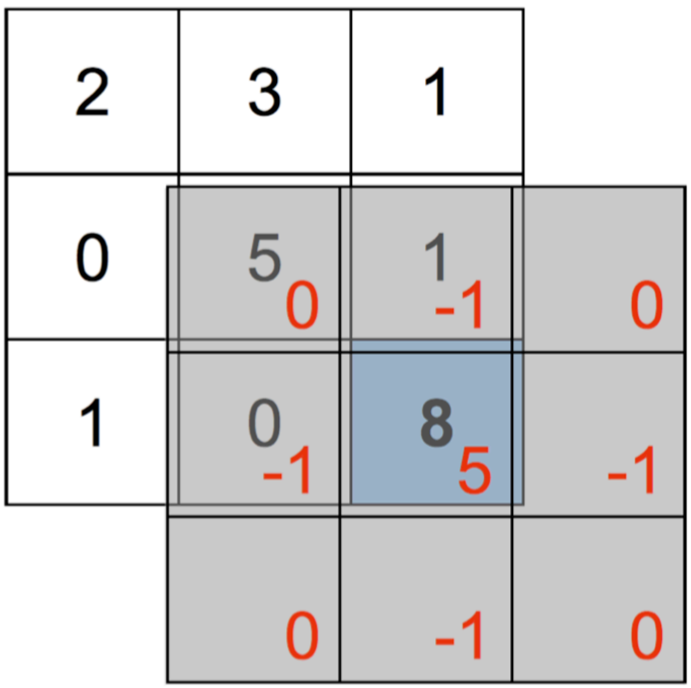

0 -1 0
-1 2 5 3 -1 1
0 0 -1 5 0 1
1 0 8
7


0 -1 0
2 -1 3 5 1 -1
0 0 5 -1 1 0
1 0 8
7 7


0 -1 0
2 3 -1 1 5 -1
0 5 0 1 -1 0
1 0 8
7 7 1




| | | |
|---|---|---|
| 7 | 7 | 1 |
| -8 | | |
| | | |



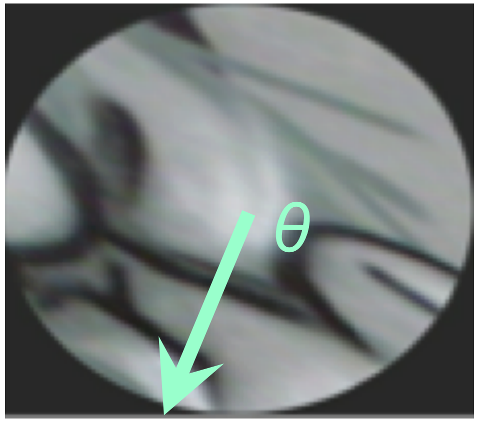

| | | |
|---|---|---|
| 7 | 7 | 1 |
| -8 | 21 | -9 |
| 5 | -14 | 39 |

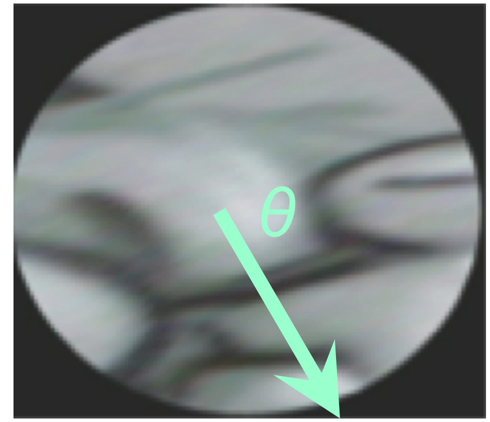
| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

| n \ m | -1 | 0 | 1 |
|---|---|---|---|
| -1 | -1 | -2 | -1 |
| 0 | 0 | 0 | 0 |
| 1 | 1 | 2 | 1 |

| | | |
|---|---|---|
| -13 | -20 | -17 |
| -18 | -24 | -18 |
| 13 | 20 | 17 |

# Gradient

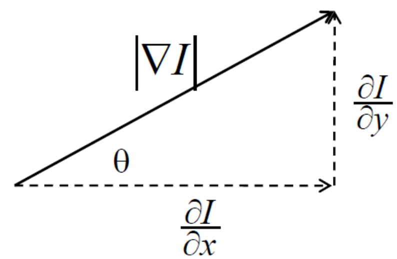


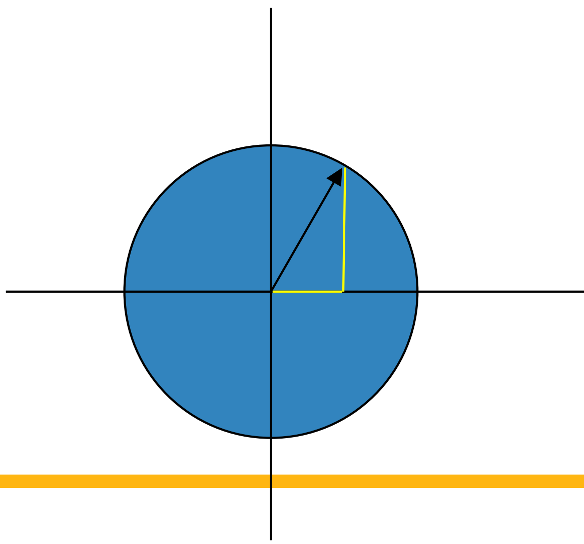




$$\nabla I = \begin{bmatrix} \dfrac{\partial I}{\partial x} \\ \\ \dfrac{\partial I}{\partial y} \end{bmatrix}$$

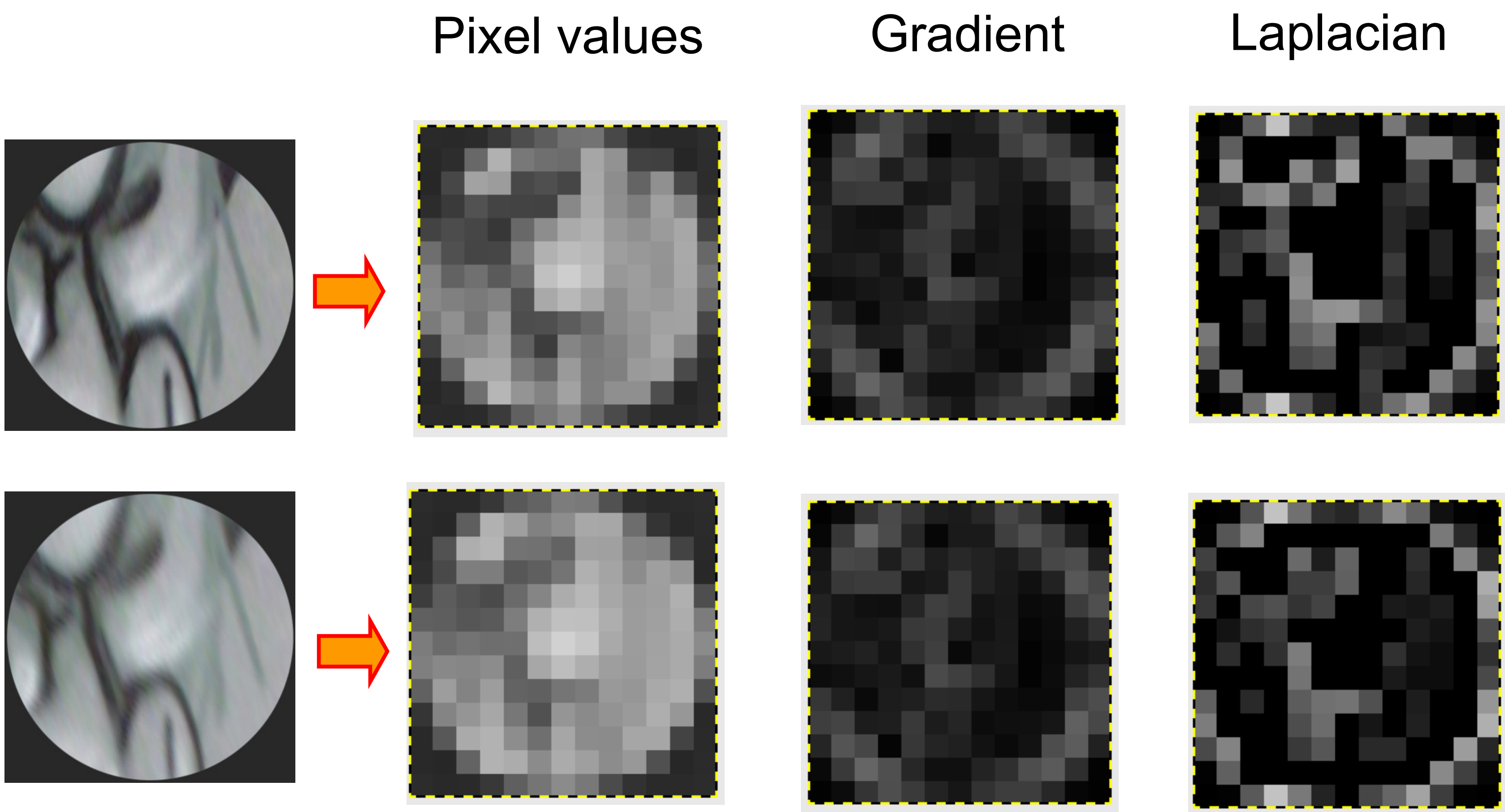
$$|\nabla I| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2} \approx \left|\frac{\partial I}{\partial x}\right| + \left|\frac{\partial I}{\partial y}\right|$$

$$\theta = \tan^{-1}\frac{\left(\dfrac{\partial I}{\partial y}\right)}{\left(\dfrac{\partial I}{\partial x}\right)}$$



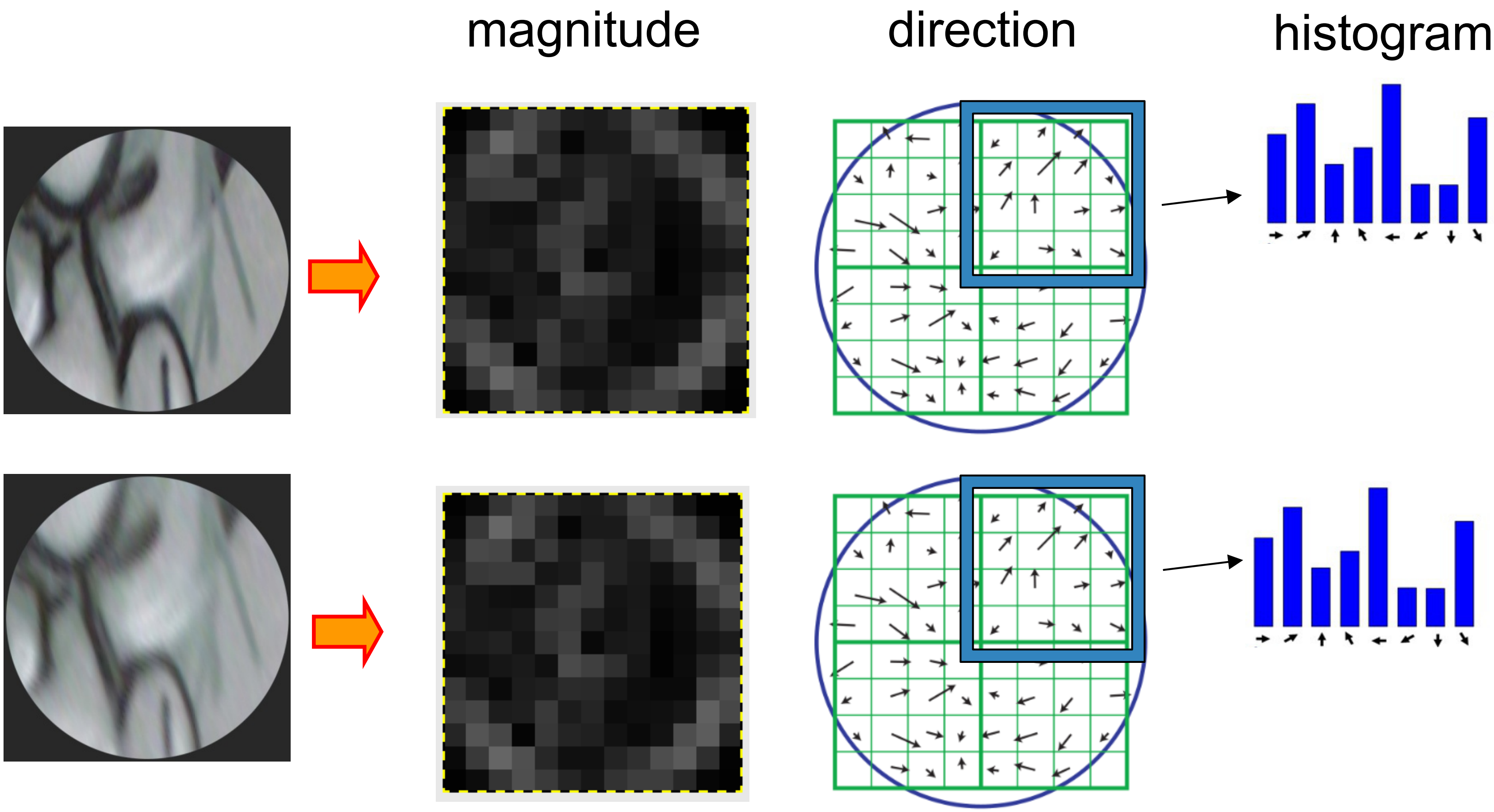
# Local Descriptor



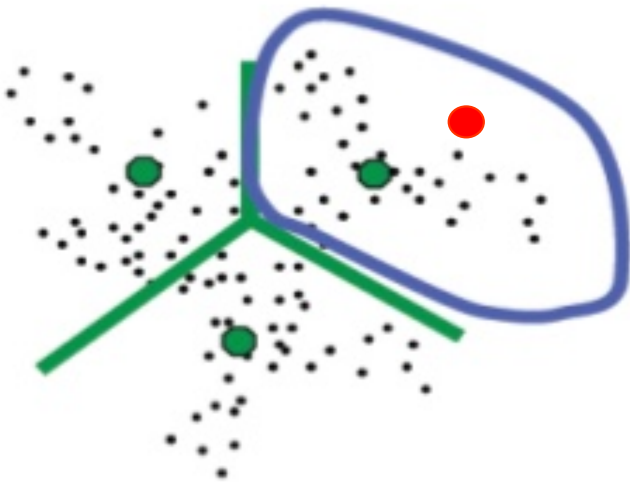
# Local Descriptor

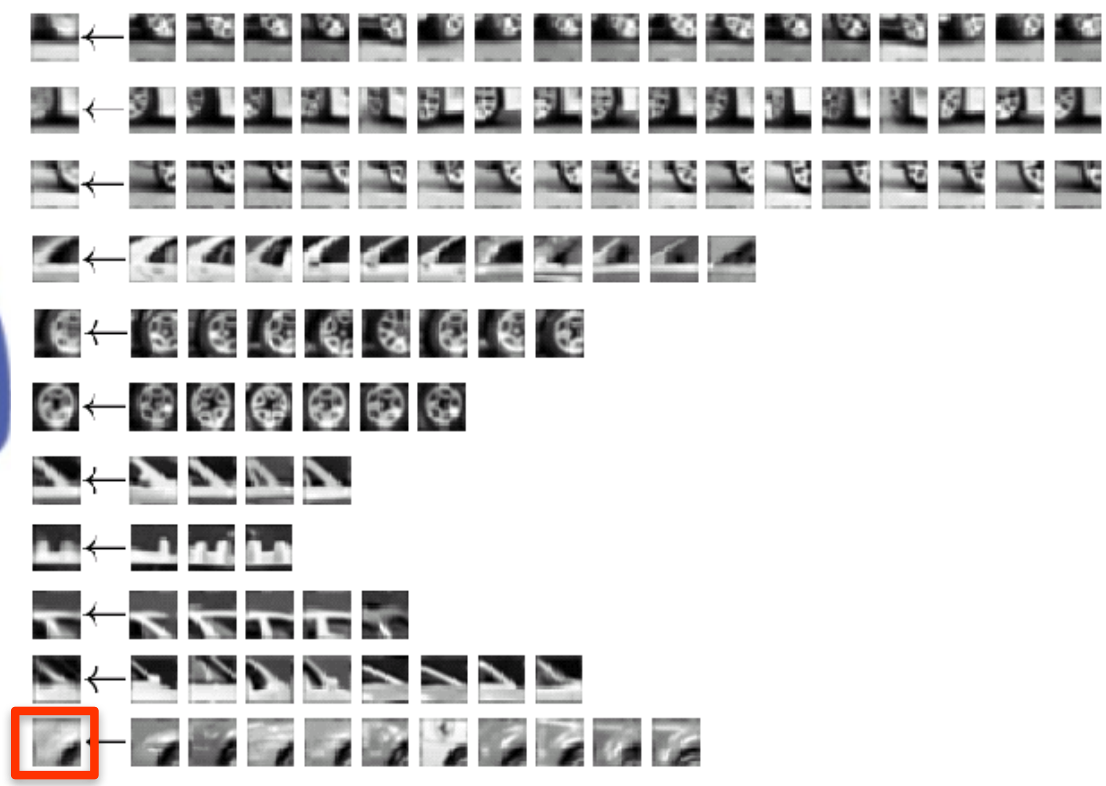
**Distinctive Image Features from Scale-Invariant Keypoints**

David G. Lowe

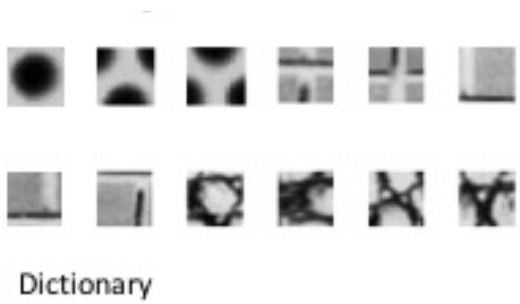


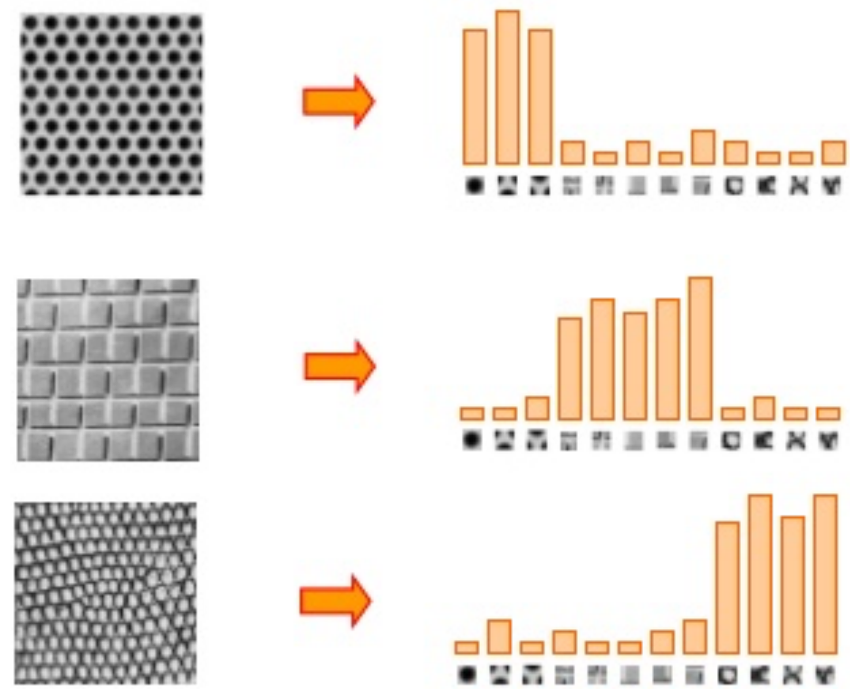
# Visual words per object category



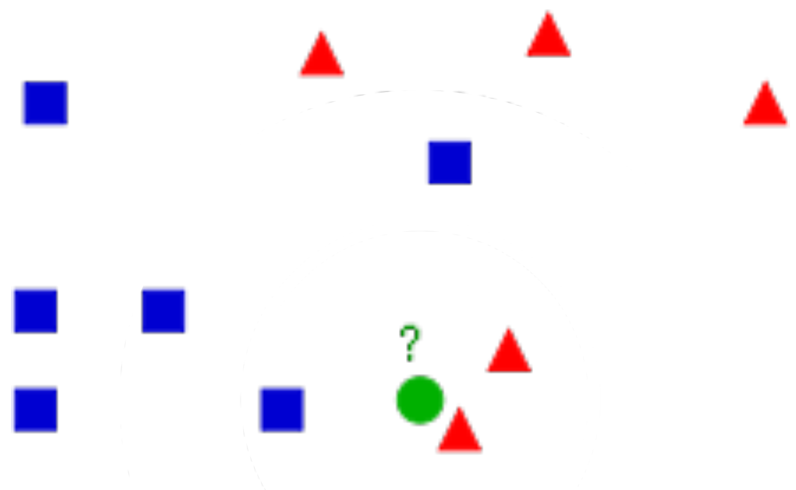


# Bag of words



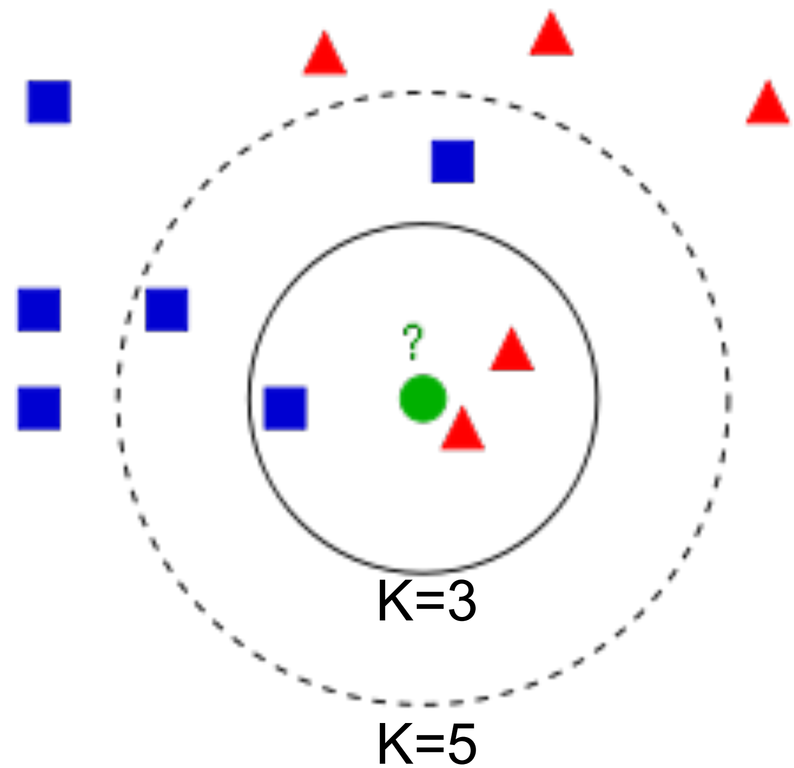
# K-nearest Neighbor classifier



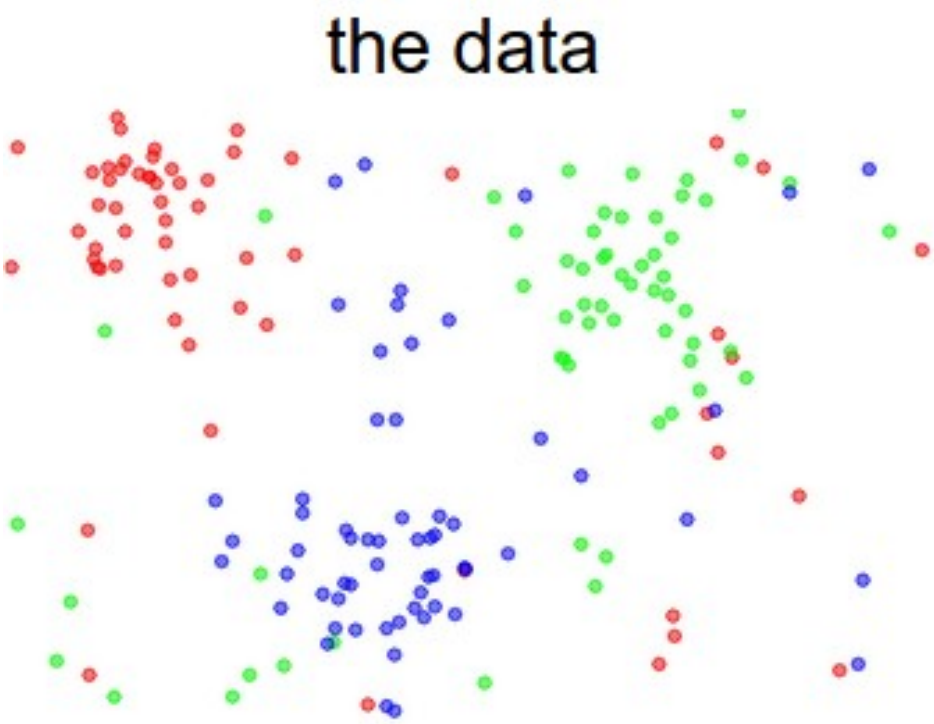

Red triangles and blue squares are the points in the training dataset.

To which class should ● (test point) be assigned to?

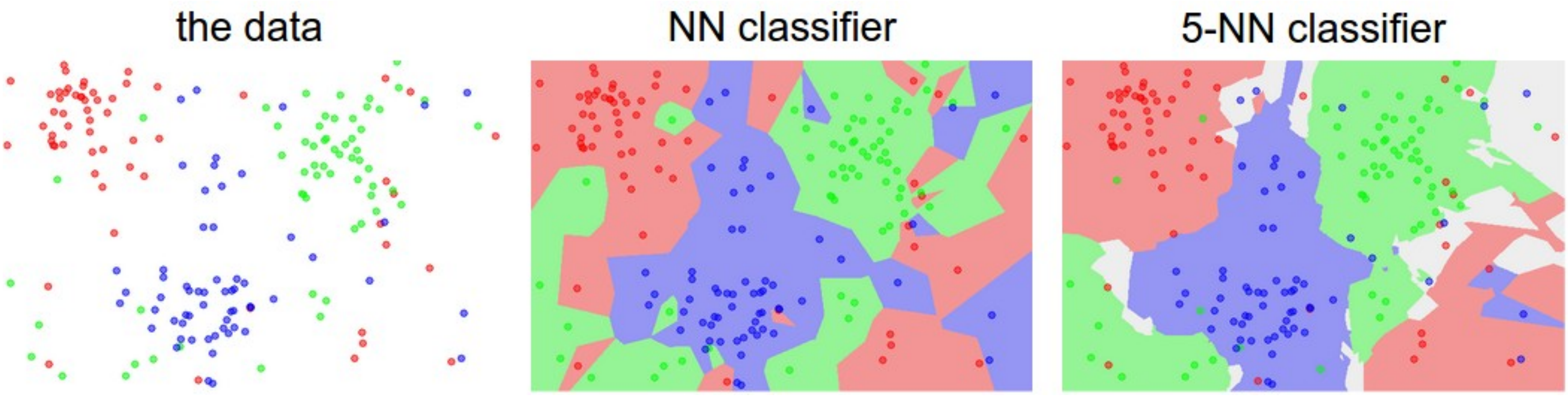
# Classification



# K-nearest Neighbor classifier

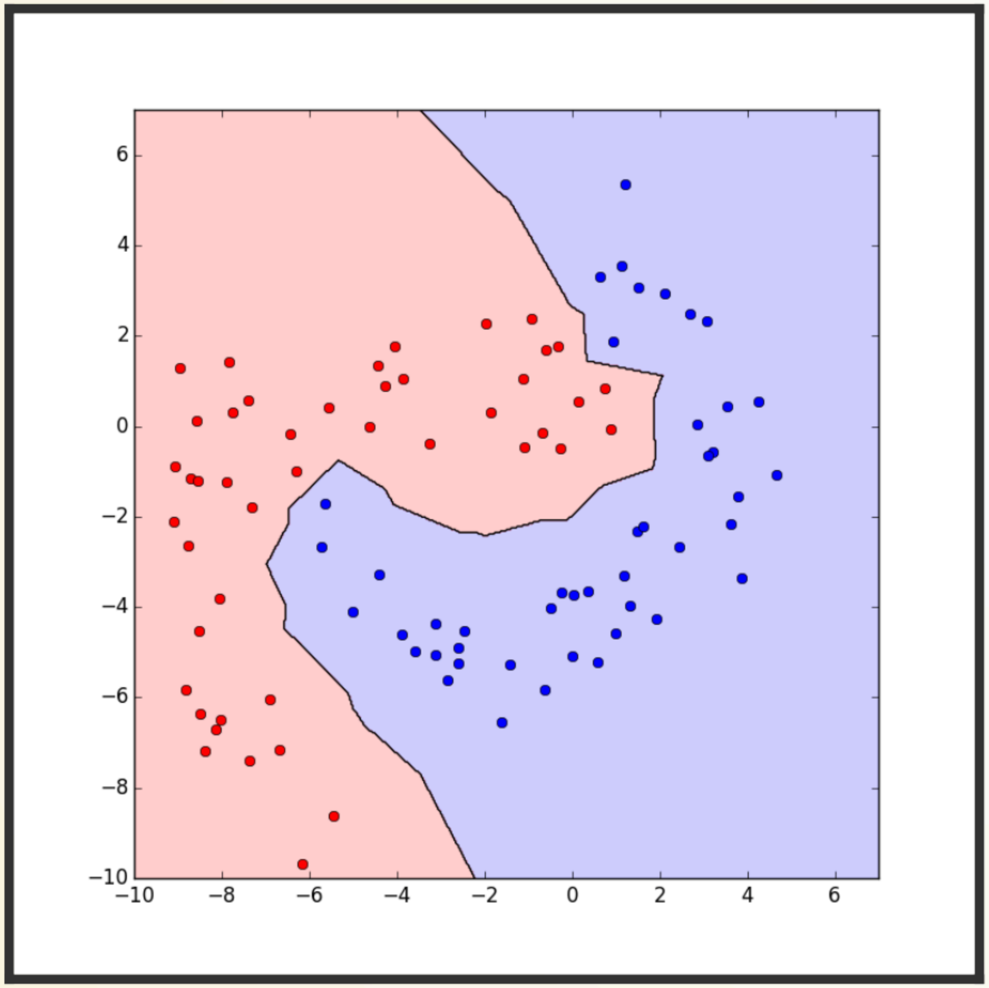


# K-nearest Neighbor classifier

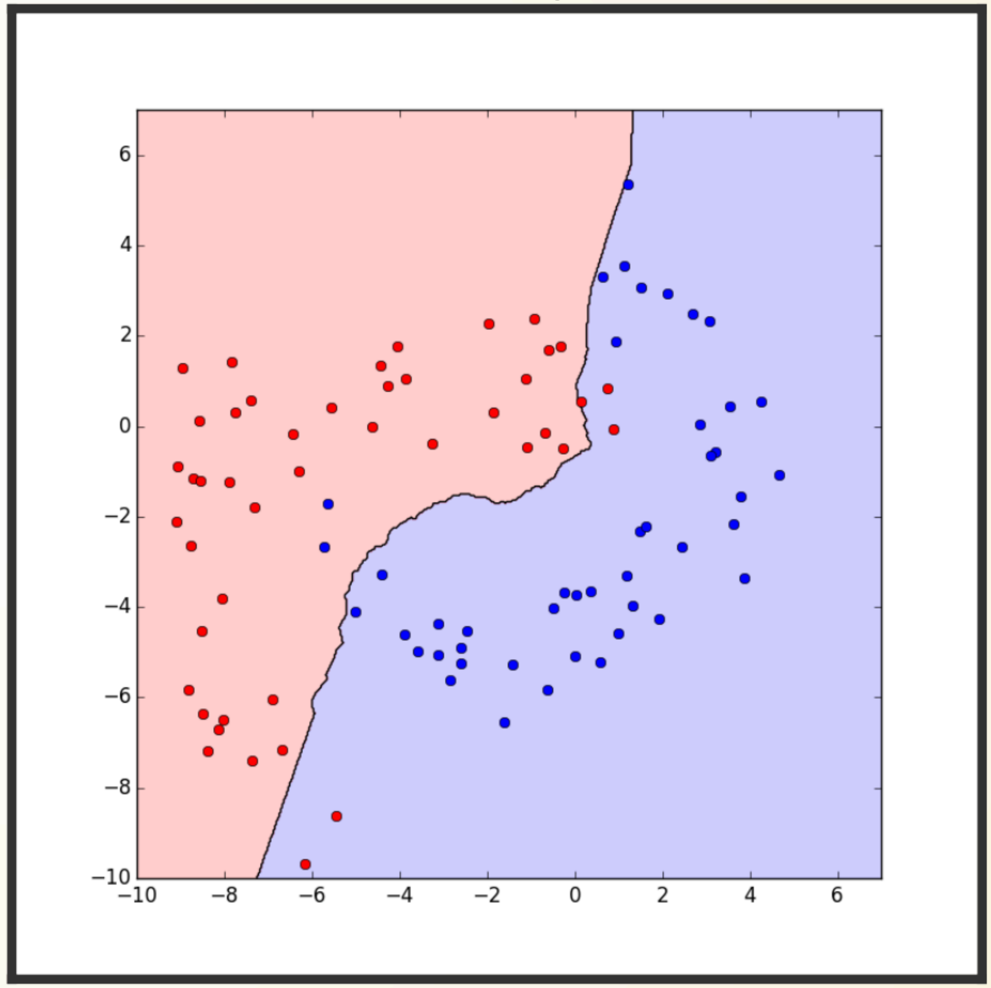


# K-nearest Neighbor classifier

K=1

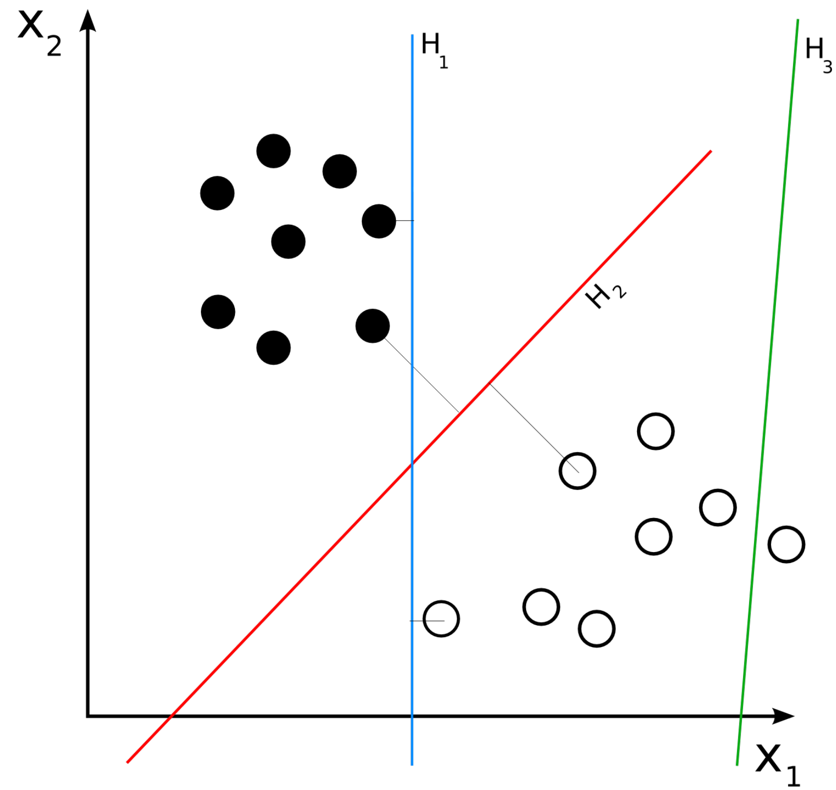



K=25



# Linear Classifier

$$y = f\left(\sum_j w_j x_j\right)$$

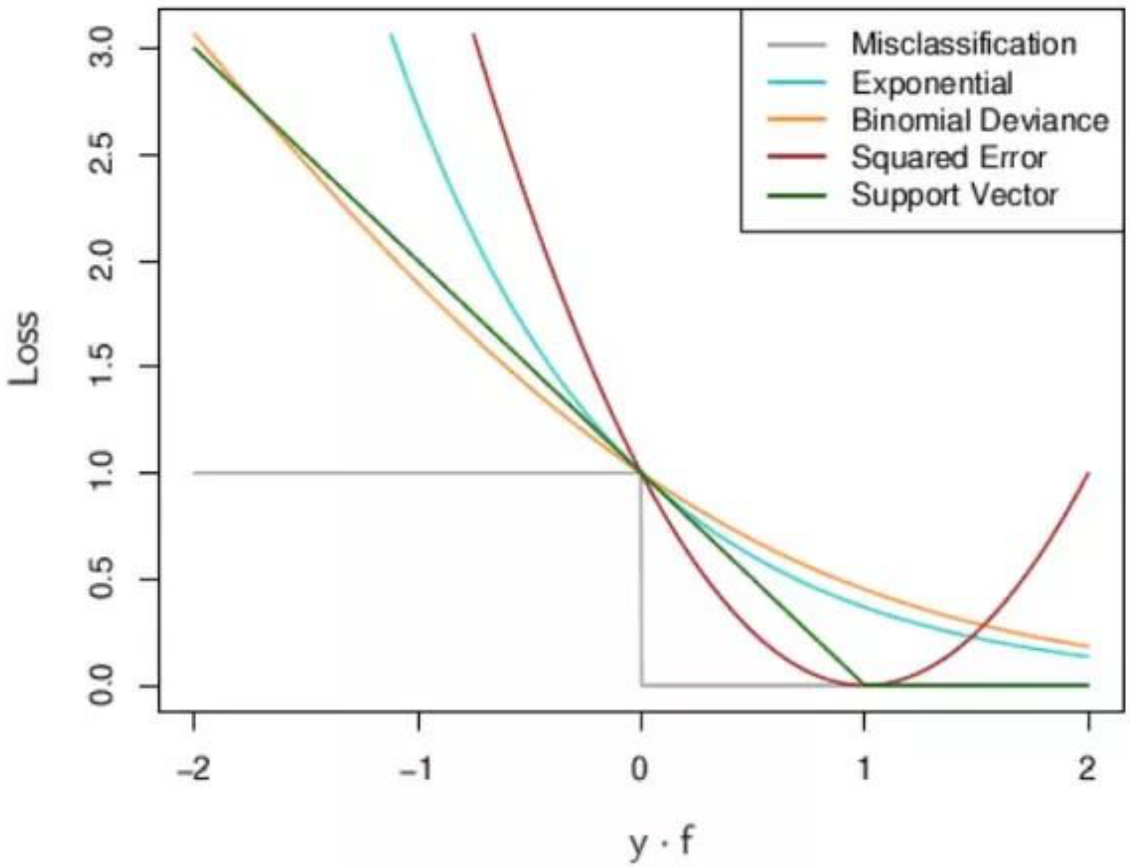


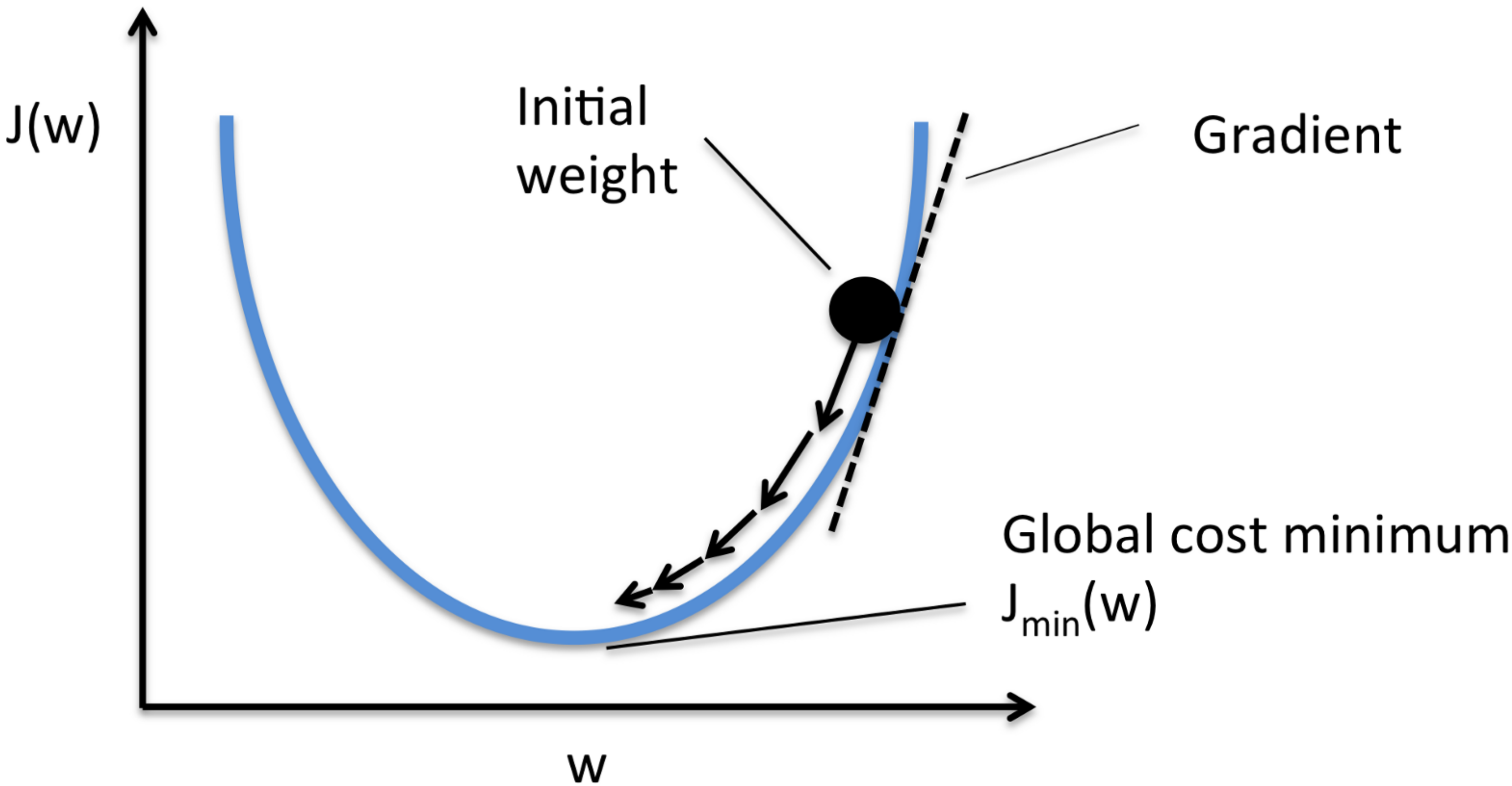

**Discriminative Learning**

$$\underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^{N} L(y_i, \mathbf{w}^\top \mathbf{x}_i)$$

Loss function
(difference between prediction and groundtruth)

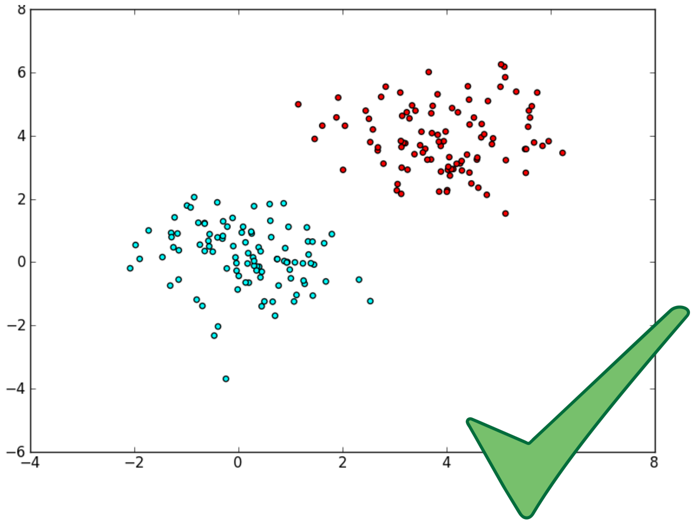
# Loss functions

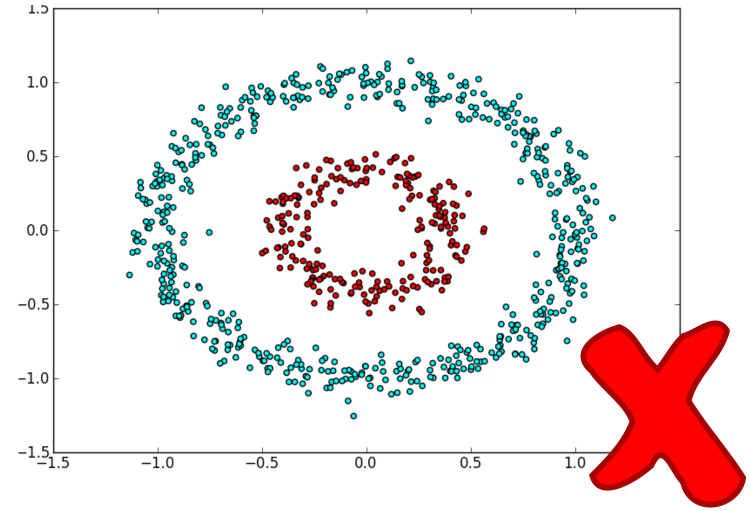




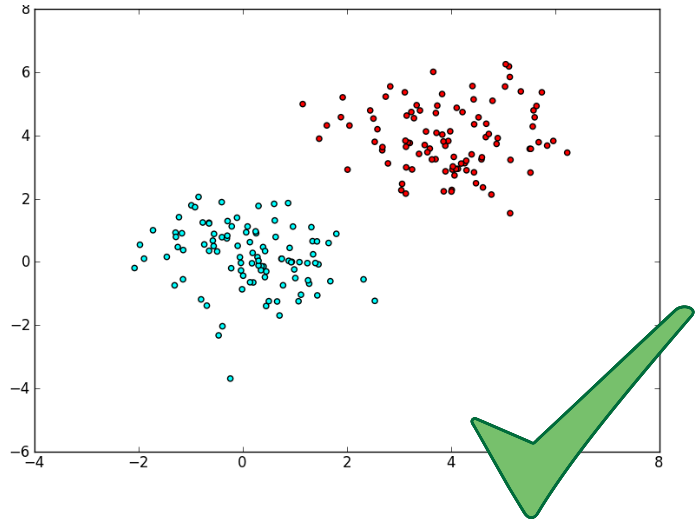
J(w)
Initial
weight
Gradient
Global cost minimum
$J_{min}(w)$
w

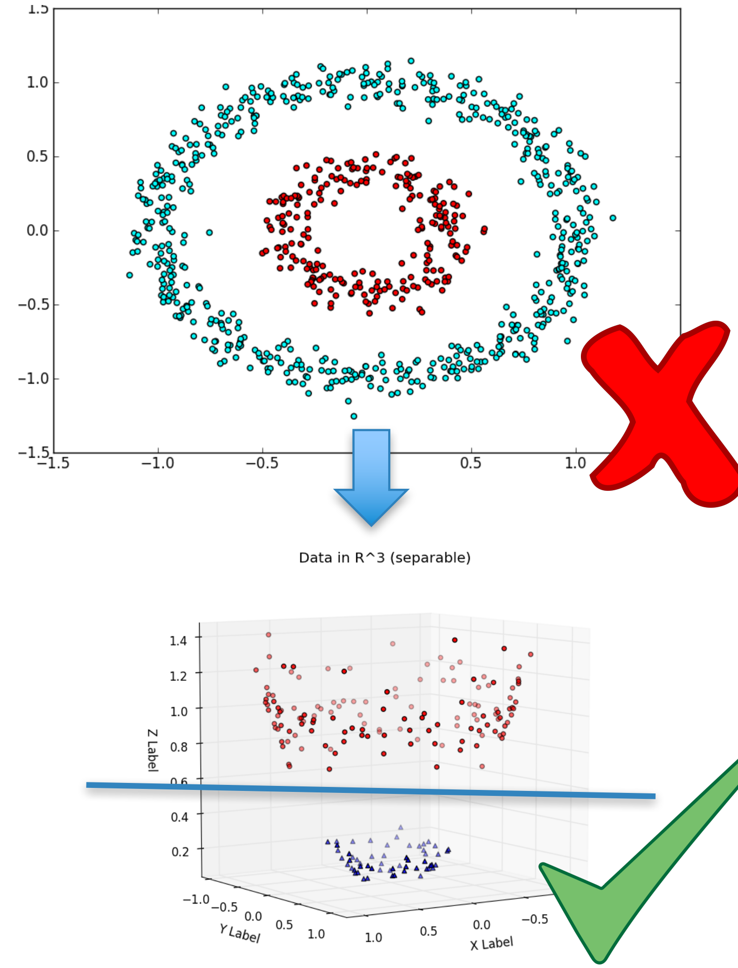
# Linear Classifier



# Kernel Projection + Linear Classifier





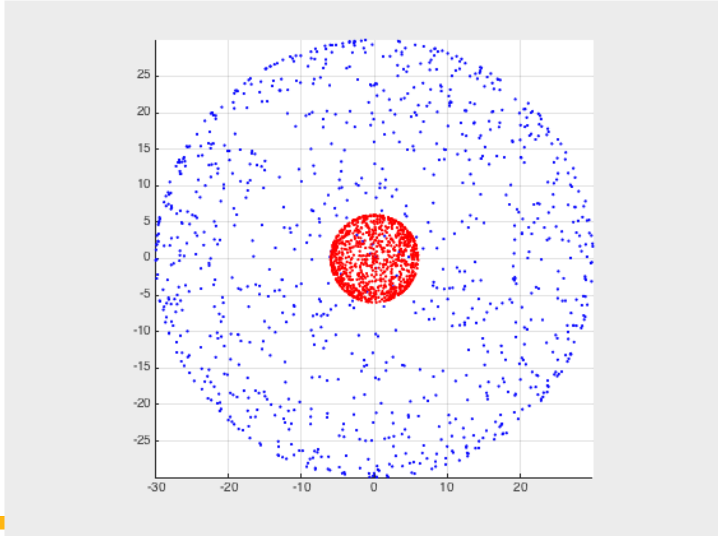

$$[x_1, x_2] = [x_1, x_2, {x_1}^2 + {x_2}^2]$$

# Kernel Projection + Linear Classifier

**Radial Basis Function (RBF) Kernel**

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$$

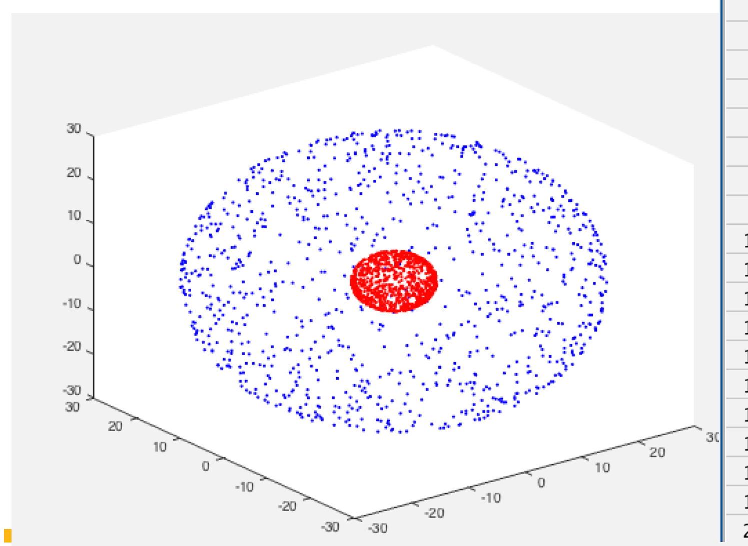
**Trick:** represent each datapoint as a vector of weights to all the other datapoints



# Kernel Projection + Linear Classifier

Represents each datapoint as a vector of weights to all the datapoints

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$$




datapoints

2000x3 double

| | 1 | 2 | 3 |
|---|---|---|---|
| 1 | -0.1503 | -4.6598 | 3.7767 |
| 2 | 2.7325 | -2.1957 | 4.8695 |
| 3 | 0.1856 | 3.9912 | -4.4762 |
| 4 | 2.3168 | 2.4547 | 4.9605 |
| 5 | 0.9027 | 5.7151 | 1.5883 |
| 6 | -2.0929 | 2.8802 | -4.8295 |
| 7 | -1.2421 | 5.2338 | -2.6580 |
| 8 | 5.3642 | -2.6284 | 0.5626 |
| 9 | 2.2958 | 0.7670 | 5.4901 |
| 10 | -1.8448 | -1.2146 | 5.5787 |
| 11 | 2.2753 | 3.7339 | -4.1086 |
| 12 | 1.0691 | -1.7225 | 5.6471 |
| 13 | 1.2030 | 2.1110 | 5.4860 |
| 14 | -5.9969 | -0.0829 | -0.1755 |
| 15 | 4.7974 | -0.0202 | 3.6034 |
| 16 | -2.5748 | 3.3020 | -4.2974 |
| 17 | 5.6687 | 1.7275 | -0.9389 |
| 18 | 0.7545 | 3.2469 | 4.9888 |
| 19 | -3.8997 | 2.9150 | 3.5065 |
| 20 | -1.1872 | 2.0463 | 5.5139 |

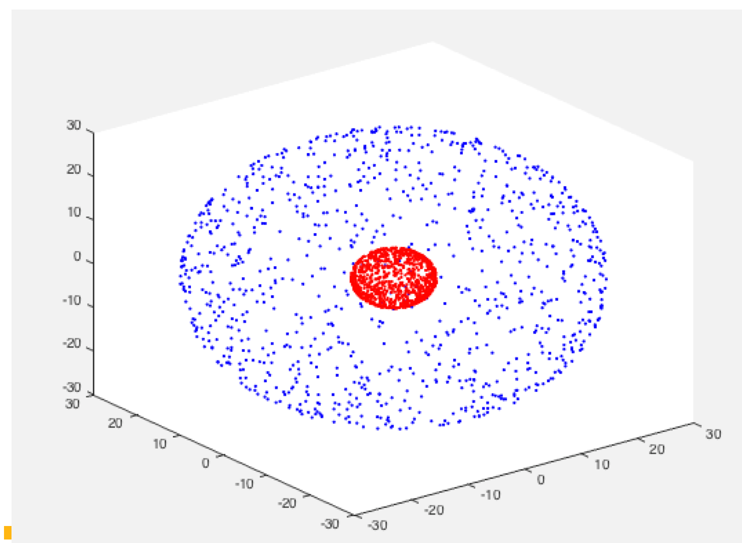
Class 1

Class 2

# Kernel Projection + Linear Classifier

Represents each datapoint as a vector of weights to all the datapoints

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$$



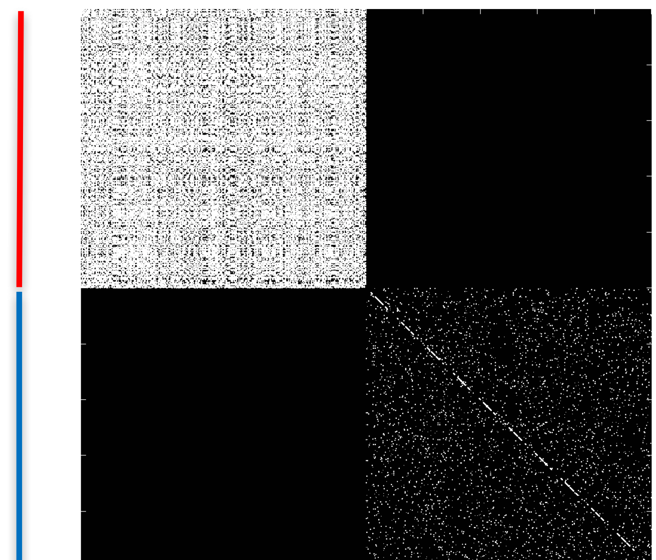

datapoints

2000x3 double

| | 1 | 2 | 3 |
|---|---|---|---|
| 1 | -0.1503 | -4.6598 | 3.7767 |
| 2 | 2.7325 | -2.1957 | 4.8695 |
| 3 | 0.1856 | 3.9912 | -4.4762 |
| 4 | 2.3168 | 2.4547 | 4.9605 |
| 5 | 0.9027 | 5.7151 | 1.5883 |
| 6 | -2.0929 | 2.8802 | -4.8295 |
| 7 | -1.2421 | 5.2338 | -2.6580 |
| 8 | 5.3642 | -2.6284 | 0.5626 |
| 9 | 2.2958 | 0.7670 | 5.4901 |
| 10 | -1.8448 | -1.2146 | 5.5787 |
| 11 | 2.2753 | 3.7339 | -4.1086 |
| 12 | 1.0691 | -1.7225 | 5.6471 |
| 13 | 1.2030 | 2.1110 | 5.4860 |
| 14 | -5.9969 | -0.0829 | -0.1755 |
| 15 | 4.7974 | -0.0202 | 3.6034 |
| 16 | -2.5748 | 3.3020 | -4.2974 |
| 17 | 5.6687 | 1.7275 | -0.9389 |
| 18 | 0.7545 | 3.2469 | 4.9888 |
| 19 | -3.8997 | 2.9150 | 3.5065 |
| 20 | -1.1872 | 2.0463 | 5.5139 |

Class 1

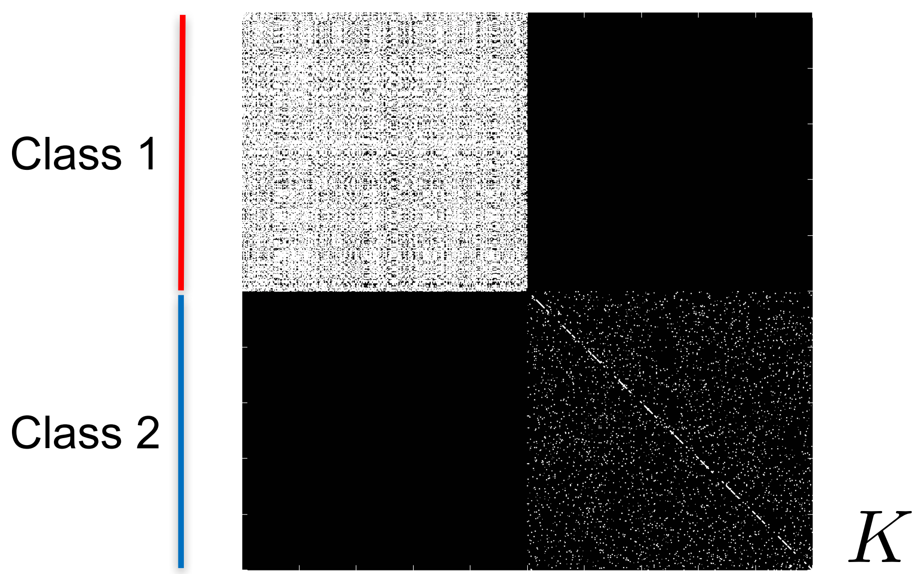
Class 2



$K$

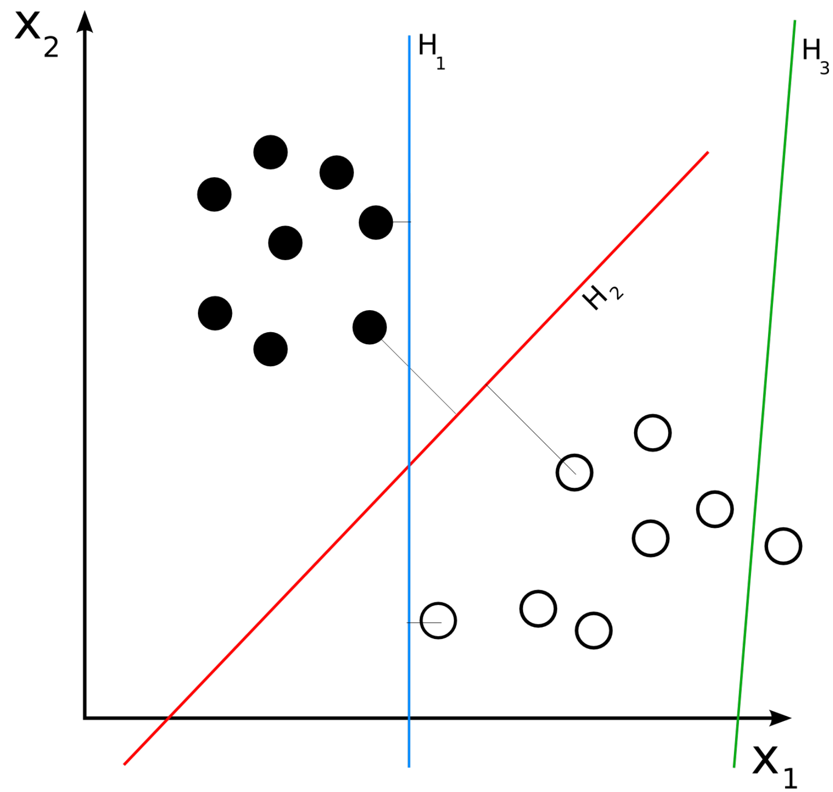
# Kernel Projection + Linear Classifier

$$\underset{\mathbf{w}}{\operatorname{argmin}}\; R(\mathbf{w}) + C\sum_{i=1}^{N} L(y_i, \mathbf{w}^{\mathsf{T}} K(\mathbf{x}_i, \mathbf{x}))$$



# Linear Classifier

$$y = f\left(\sum_j w_j x_j\right)$$



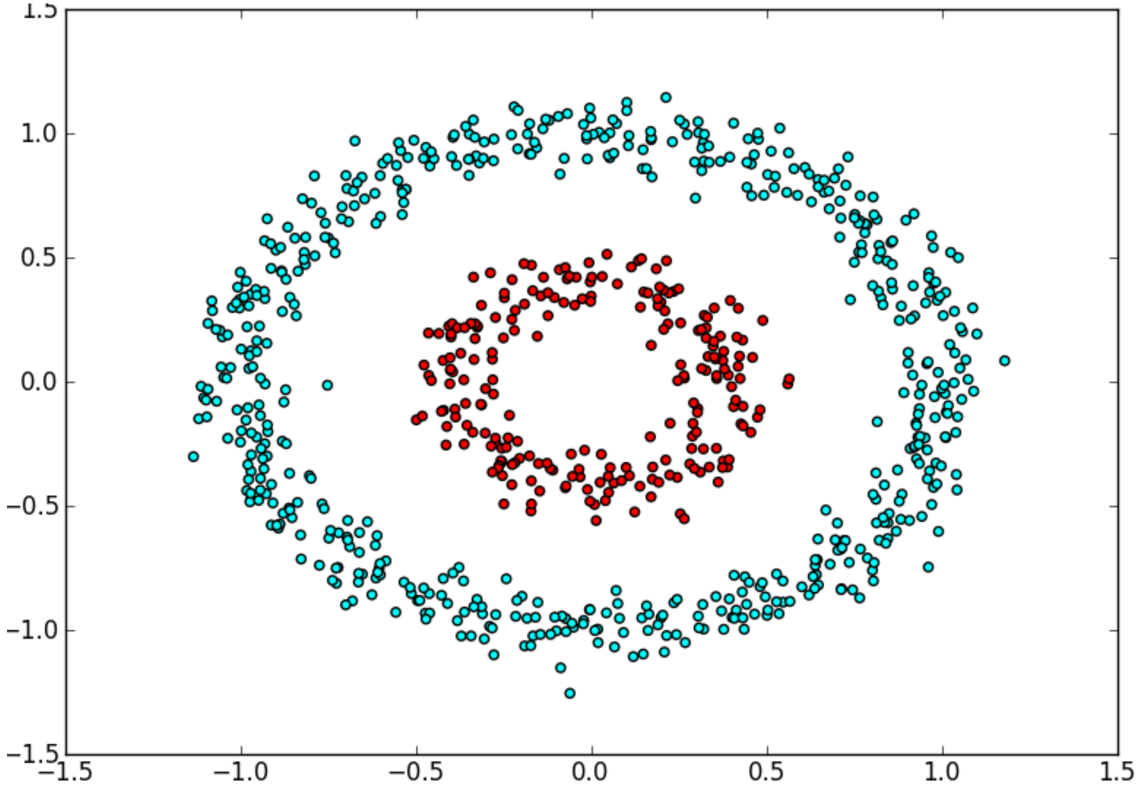

**Discriminative Learning**

$$\underset{\mathbf{w}}{\operatorname{argmin}}\ \underbrace{R(\mathbf{w})}_{\text{Regularization (prevents overfitting)}} + C \sum_{i=1}^{N} \underbrace{L(y_i, \mathbf{w}^{\mathsf{T}} \mathbf{x}_i)}_{\text{Loss (difference between prediction and groundtruth)}}$$

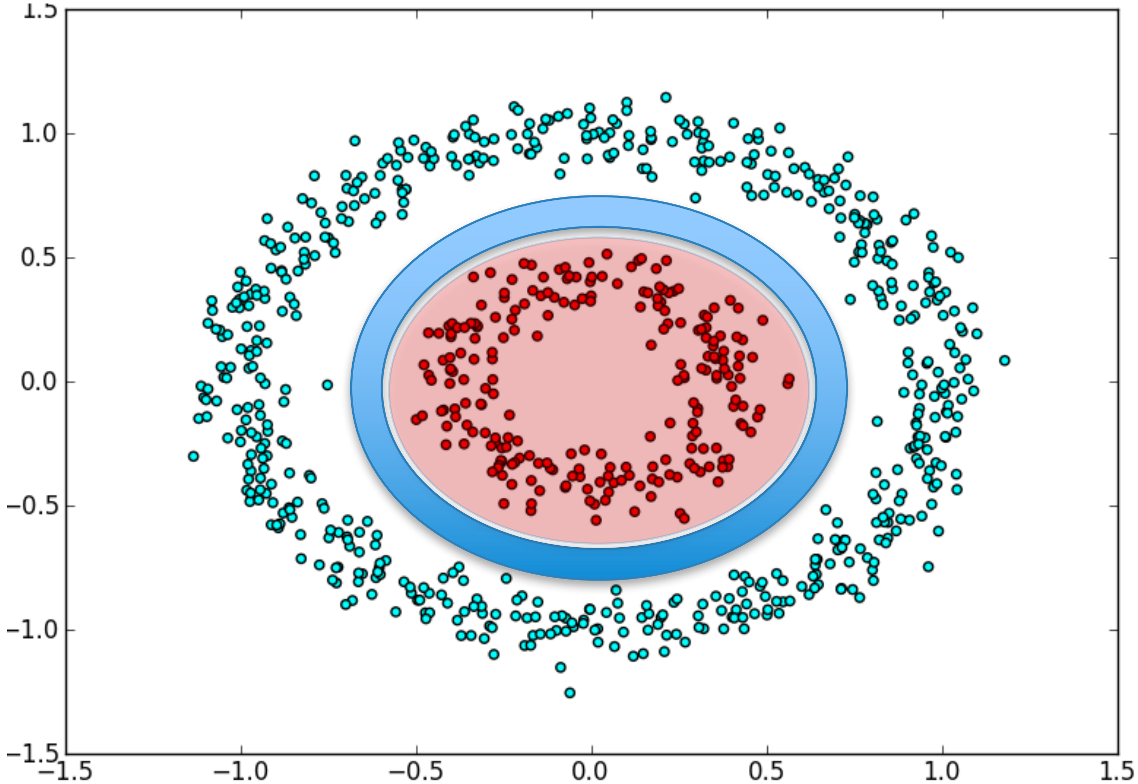
Regularization
(prevents overfitting)

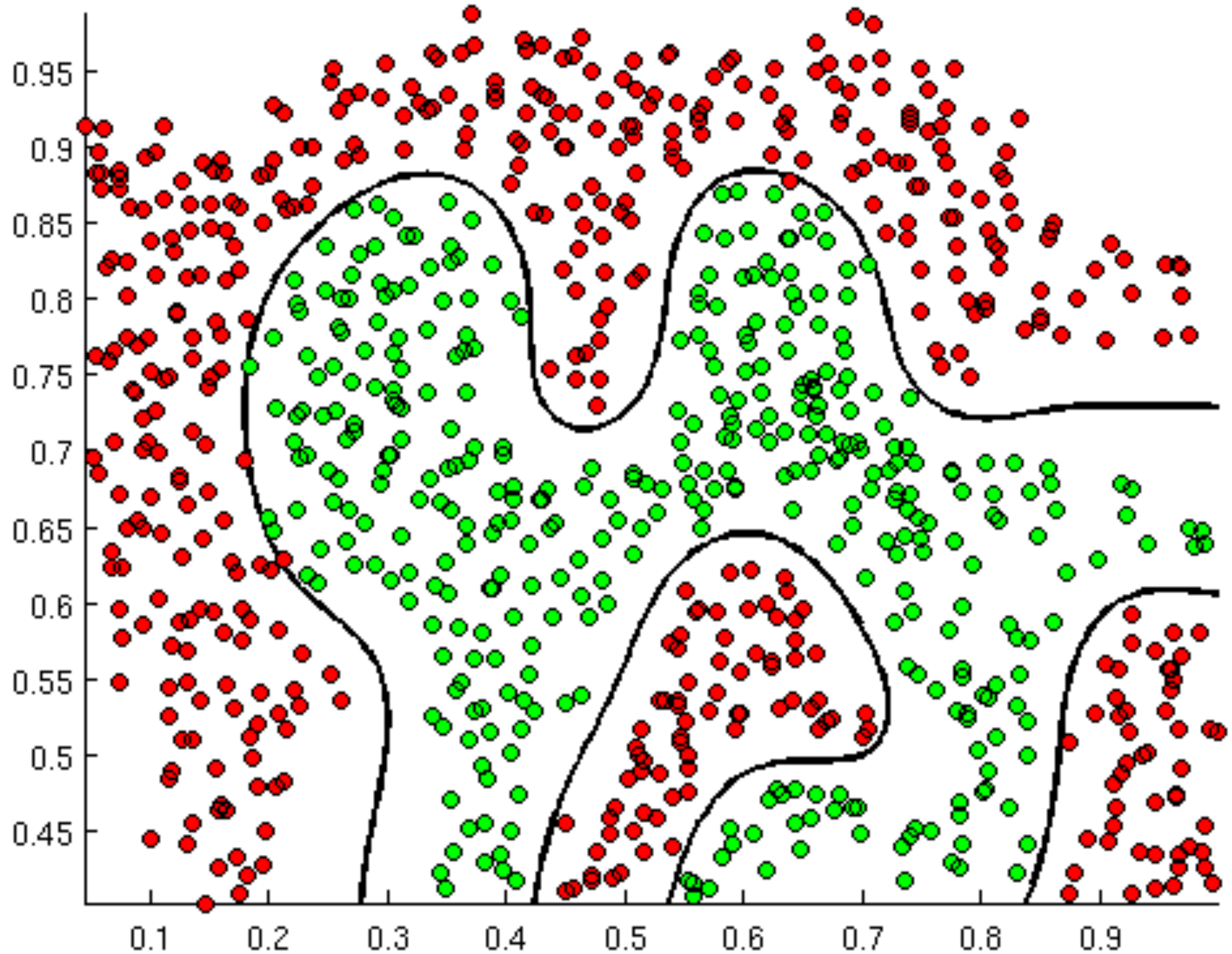
Loss
(difference between prediction
and groundtruth)

# Kernel Classifier



# Kernel Classifier





0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0.45
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9