UNIVERSITÉ DE FRIBOURG
UNIVERSITÄT FREIBURG

MASTER IN COMPUTER SCIENCE

# Pattern Recognition

## Lecture 1 : Classification

Dr. Andreas Fischer

andreas.fischer@unifr.ch

# Personal Background

- 2008-12: PhD in Computer Science
  - University of Bern
- 2012-15: PostDoc
  - University of Fribourg
  - Concordia University, Montréal
  - Ecole Polytechnique de Montréal
- Current:
  - Professor: University of Applied Sciences and Arts Western Switzerland
  - Lecturer: University of Fribourg

# Current Research Topics


1896. SCHWEIZERISCHE LEHRERZEITUNG. № 2.

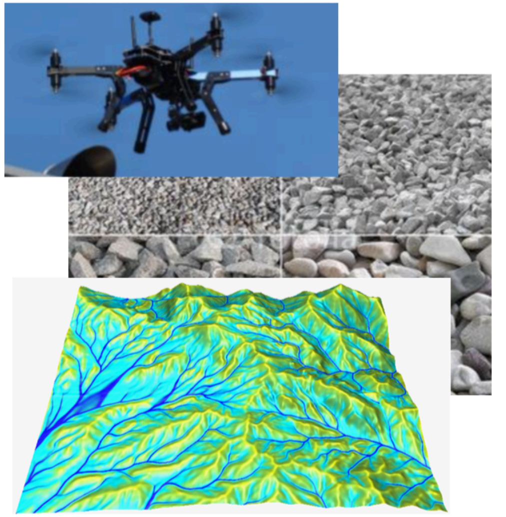
HEINRICH PESTALOZZI.

An Vater Pestalozzi.

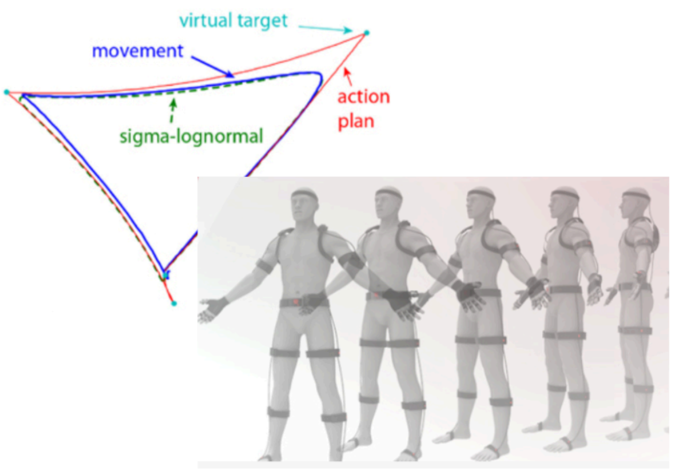
Zur Feier seines 150jährigen Geburtstages

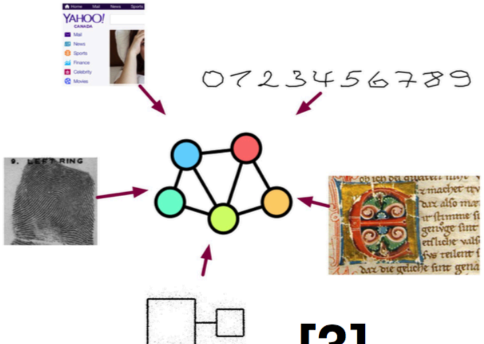
von EMIL FALLER.

Es gibt noch heil'ge Menschen hier auf Erden,
Es gibt noch Christussöhne selt'ner Art.
So soll ein Weiser heut gefeiert werden,
Der Weisheit mit der reinsten Lieb' gepaart.
„O lasst, o lasst die Kindlein zu mir kommen!
Ich geb' von Herzen, was der Schar mag frommen."
So sprach er sanft, dem grossen Heiland gleich,
Und wie ein Bote aus dem Himmelreich.

**Pattern Recognition**
**& Artificial Intelligence**



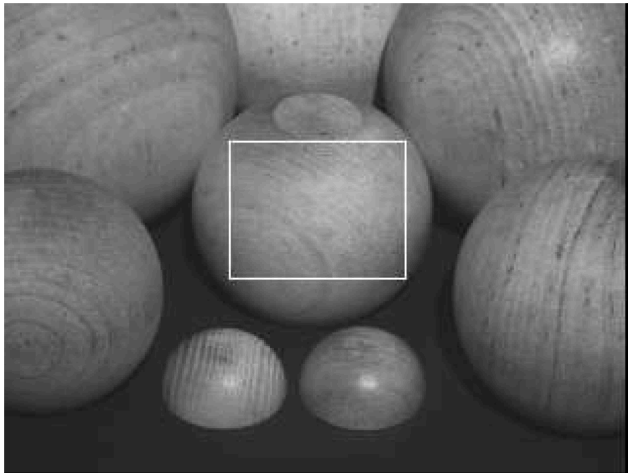

[3]

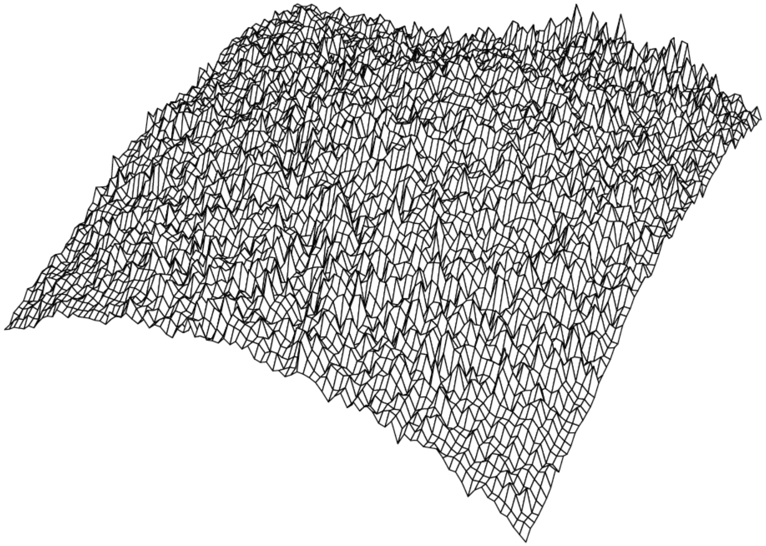
# Goals

# Goals

- You will be able to perform data analysis based on pattern recognition and machine learning.
- In particular, you will learn how to classify objects represented with:
  - real-valued numbers (statistical pattern recognition)
  - strings and graphs (structural pattern recognition)
- The methods studied are applicable to a wide range of data:
  - sensory data (image, video, audio, location, …)
  - born-digital data (text, network traffic, chemical formulas, …)
- Relevant in research as well as industry.

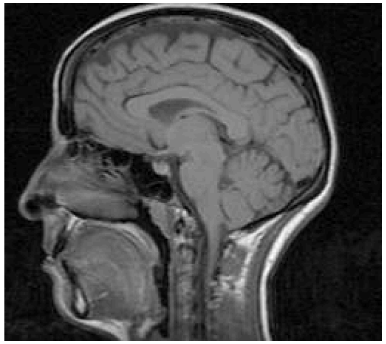
# Representation

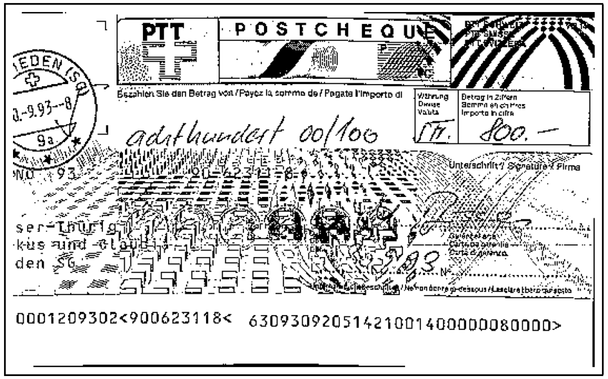


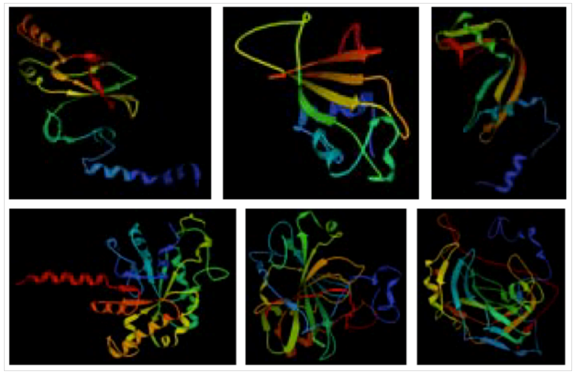


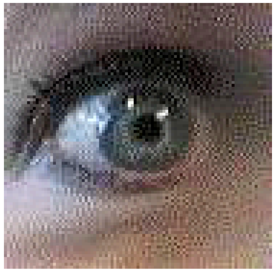
| | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 118 | 119 | 129 | 120 | 125 | 112 | 117 | 123 | 117 | 135 | 123 | 134 | 143 | 153 | 151 | 150 | 137 | 149 | 147 | 153 | 145 | 140 | 140 |
| 117 | 126 | 118 | 132 | 112 | 120 | 130 | 137 | 130 | 136 | 136 | 137 | 157 | 156 | 161 | 148 | 157 | 155 | 161 | 152 | 145 | 142 | 144 |
| 122 | 122 | 120 | 132 | 125 | 125 | 136 | 133 | 130 | 143 | 152 | 148 | 153 | 156 | 157 | 172 | 170 | 162 | 168 | 161 | 142 | 142 | 148 |
| 132 | 135 | 136 | 142 | 118 | 137 | 145 | 144 | 142 | 139 | 153 | 146 | 158 | 157 | 169 | 170 | 175 | 170 | 163 | 165 | 158 | 151 | 155 |
| 131 | 132 | 140 | 141 | 138 | 140 | 146 | 147 | 159 | 151 | 157 | 156 | 170 | 171 | 181 | 181 | 179 | 168 | 168 | 155 | 150 | 153 | 145 |
| 133 | 140 | 137 | 143 | 148 | 145 | 151 | 164 | 163 | 160 | 171 | 167 | 172 | 190 | 186 | 180 | 176 | 182 | 166 | 158 | 149 | 144 | 144 |
| 133 | 132 | 145 | 145 | 147 | 155 | 145 | 164 | 166 | 169 | 172 | 177 | 189 | 178 | 178 | 175 | 180 | 177 | 172 | 157 | 165 | 150 | 141 |
| 126 | 137 | 136 | 143 | 149 | 151 | 153 | 165 | 166 | 155 | 167 | 181 | 177 | 178 | 179 | 174 | 179 | 166 | 170 | 161 | 155 | 151 | 146 |
| 125 | 131 | 126 | 131 | 144 | 150 | 157 | 163 | 161 | 165 | 169 | 177 | 180 | 168 | 174 | 171 | 162 | 171 | 167 | 163 | 166 | 150 | 142 |
| 127 | 132 | 136 | 155 | 145 | 147 | 139 | 160 | 156 | 168 | 161 | 180 | 164 | 164 | 159 | 171 | 169 | 161 | 172 | 157 | 165 | 151 | 140 |
| 122 | 131 | 129 | 149 | 146 | 147 | 171 | 144 | 159 | 173 | 173 | 184 | 178 | 173 | 165 | 170 | 167 | 169 | 159 | 155 | 149 | 140 | 126 |
| 120 | 136 | 142 | 143 | 134 | 144 | 152 | 154 | 149 | 160 | 154 | 163 | 164 | 170 | 166 | 165 | 148 | 144 | 152 | 146 | 151 | 129 | 129 |
| 107 | 118 | 121 | 125 | 138 | 147 | 147 | 160 | 161 | 147 | 152 | 160 | 150 | 147 | 165 | 163 | 163 | 145 | 134 | 139 | 143 | 127 | 113 |
| 102 | 111 | 118 | 129 | 136 | 134 | 147 | 129 | 155 | 157 | 138 | 156 | 160 | 154 | 155 | 152 | 143 | 134 | 131 | 139 | 134 | 111 | 115 |
| 92 | 99 | 116 | 115 | 130 | 133 | 132 | 140 | 142 | 146 | 132 | 133 | 151 | 140 | 142 | 136 | 134 | 136 | 124 | 118 | 118 | 104 | 100 |
| 91 | 92 | 104 | 118 | 125 | 130 | 133 | 138 | 138 | 135 | 137 | 141 | 137 | 137 | 147 | 126 | 135 | 119 | 129 | 116 | 97 | 109 | 85 |
| 82 | 95 | 104 | 102 | 115 | 132 | 131 | 126 | 133 | 131 | 135 | 134 | 120 | 135 | 124 | 127 | 127 | 119 | 120 | 113 | 98 | 89 | 85 |
| 76 | 87 | 95 | 106 | 106 | 113 | 120 | 123 | 126 | 129 | 130 | 134 | 118 | 125 | 120 | 125 | 117 | 113 | 116 | 98 | 89 | 83 | 68 |

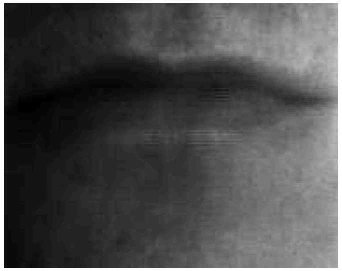
# Applications

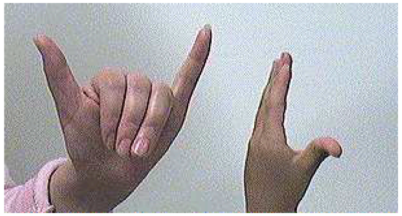








# Literature

- Requirements:
  - Basic knowledge in mathematics: linear algebra, vector spaces.
  - Basic knowledge in programming: any programming language.
- Literature (not required but helpful):
  - R. Duda, P. Hart, and D. Stork. Pattern Classification. Wiley Interscience, 2nd edition, 2000.
  - Some suggested scientific journals: Pattern Recognition (PR), Pattern Recognition Letters (PRL), IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI).
- This course is based, in large part, on the pattern recognition course and script of Prof. Em. Horst Bunke.

# Organization

# Organization

- Lecturer: Dr. Andreas Fischer <andreas.fischer@unifr.ch>
- Teaching assistants:
  - Angelika Garz <angelika.garz@unifr.ch>
  - Marcel Würsch <marcel.wuersch@unifr.ch>
  - Paul Märgner <paul.maergner@unifr.ch>
- General information:
  - ILIAS website: ilias.unibe.ch with password "**PatRec2017**"
  - 2h course & 1h exercises: Monday, 14:15 - 17:00, D230
  - Home work: 6-8h per week
  - Credits: 5 ECTS
  - Advanced Information Processing (T3) and Data Science (T6)

# Exercises and Exam

- Exercises:
  - Have to be *reasonably solved* to be allowed to the exam
  - 2 small individual tasks
  - 4 larger team tasks, at least 3 have to be solved
  - Team presentation at the end
- Exam:
  - 1h written exam
  - Date (tentative): *Monday, 12.6.2017*
  - Open books: annotated lecture notes, but no electronic devices
  - Repetition possible if failed: oral exam in September

# Semester Plan (tentative)

| Week | Lecture | Exercises | Due |
|---|---|---|---|
| W01 - 20.2 | L01 - Classification | E1a (individual) | W03 |
| W02 - 27.2. | L02 - Bayes Classifier | | |
| W03 - 06.3. | L03 - Clustering | E1b (individual) | W05 |
| W04 - 13.3. | L04 - Support Vector Machine | | |
| W05 - 20.3. | L05 - Artificial Neural Networks | E2 (team) | W08 |
| W06 - 27.3. | L06 - Classifier Optimization | | |
| W07 - 03.4 | L07 - String Matching I | | |
| W08 - 10.4. | L08 - String Matching II | E3 (team) | W11 |
| W09 - 17.4. | *no lecture (Easter)* | | |
| W10 - 24.4. | L09 - Graph Matching I | | |
| W11 - 01.5. | L10 - Graph Matching II | E4/5 (team) | W14 |
| W12 - 08.5. | *no lecture (Team Work)* | | |
| W13 - 15.5. | L11 - Kernel Methods | | |
| W14 - 22.5. | L12 - Repetition | Presentation | |
| W15 - 29.5. | *no lecture (Q & A)* | | |

# Pattern Classification

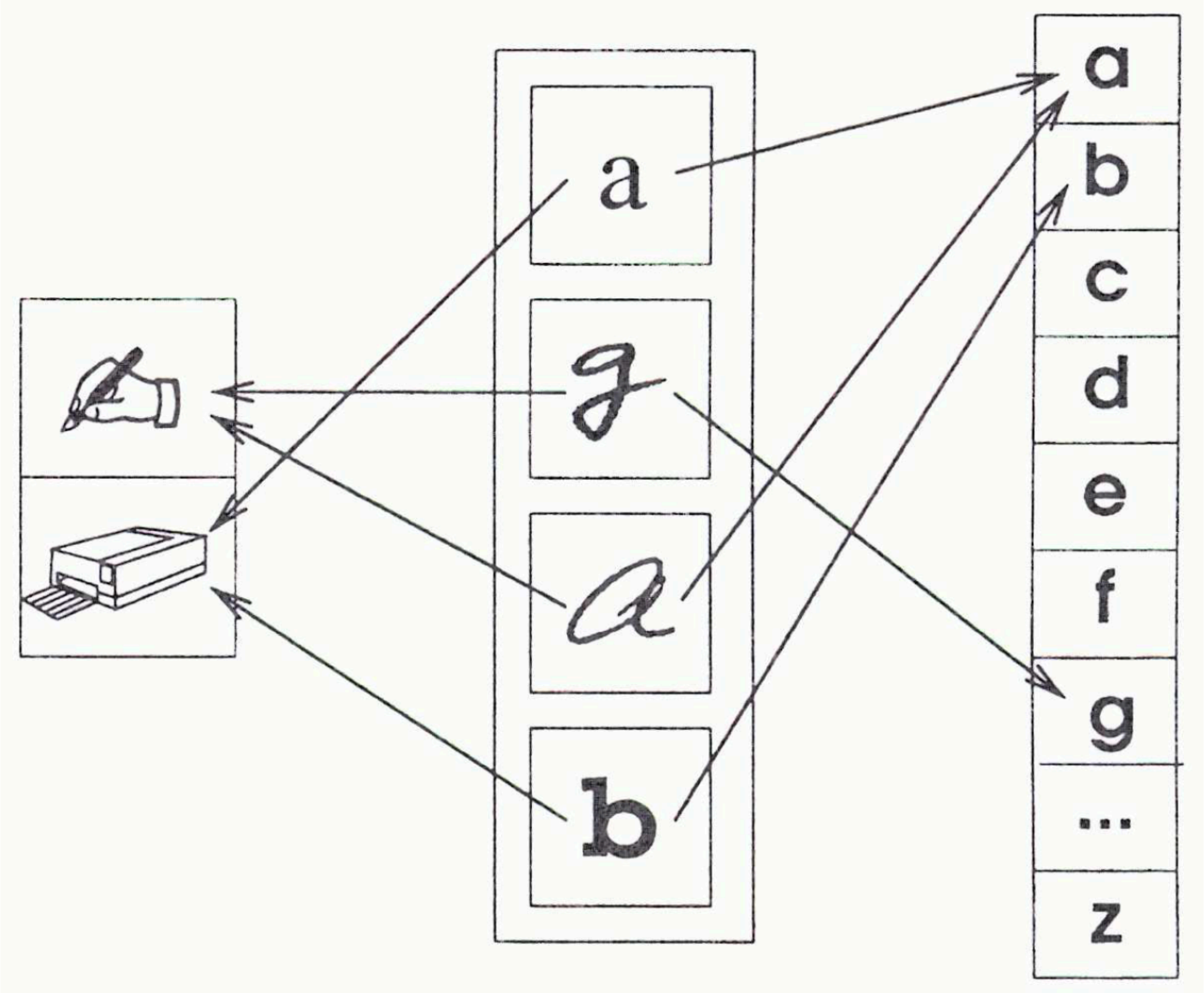
# Pattern Classification and Analysis

- Pattern classification
  - Assign a class y to a pattern x:

$$f_\theta(x) = y,\ x \in X,\ y \in Y$$

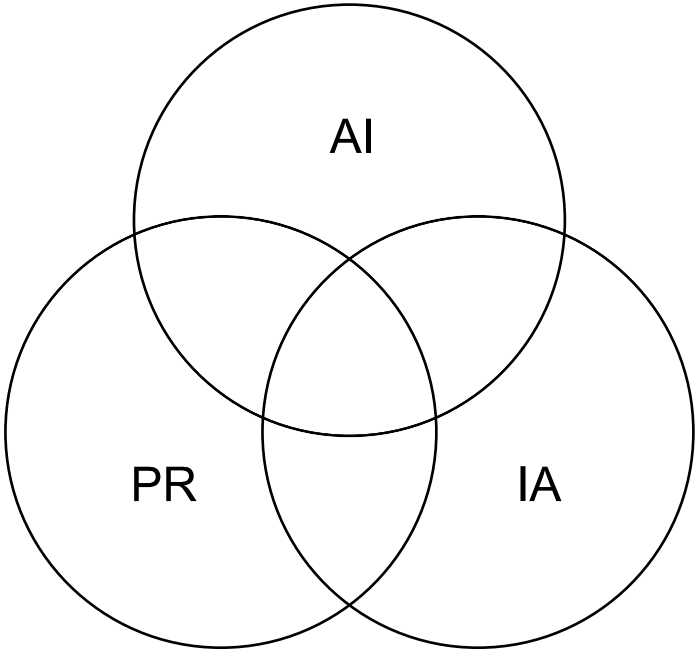
  - x : abstract pattern representation, often $X = R^n$
  - y : class label, Y = {a,b,…}
  - θ : class model, typically based on learning samples
- Pattern analysis
  - Provide a general description of the pattern, e.g. by identifying its parts and their relation.

# Example: Character Recognition



# General Context

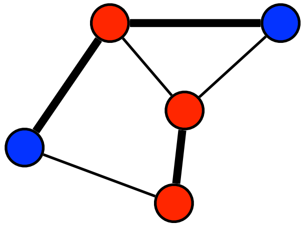
- Different disciplines are involved, their boundaries are not clearly defined. Some typical distinctions:
- Image analysis (IA)
  - preprocessing
  - segmentation
  - representation
- Pattern recognition (PR)
  - classification
  - analysis
- Artificial intelligence (AI)
  - search algorithms
  - learning algorithms



# Statistical and Structural Approach

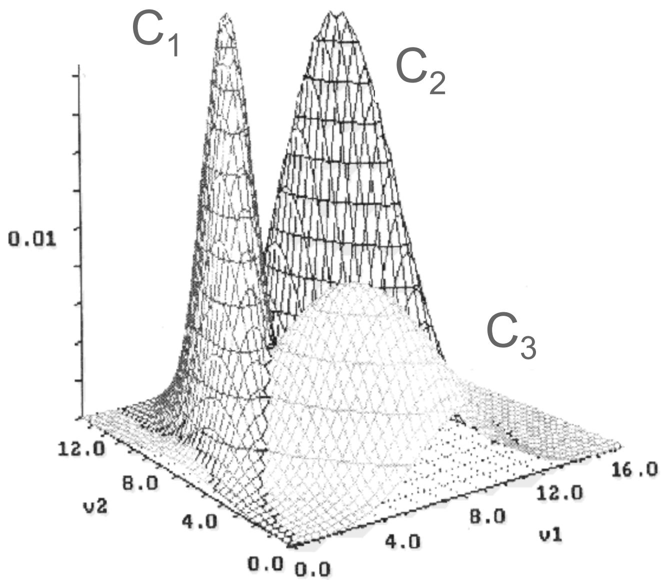
- Statistical pattern recognition
  - X : vector space $R^n$
  - θ : learned class boundaries in $R^n$
  - classification
- Structural pattern recognition
  - X : strings, trees, graphs
  - θ : comparison with known, prototypical patterns
  - classification and analysis

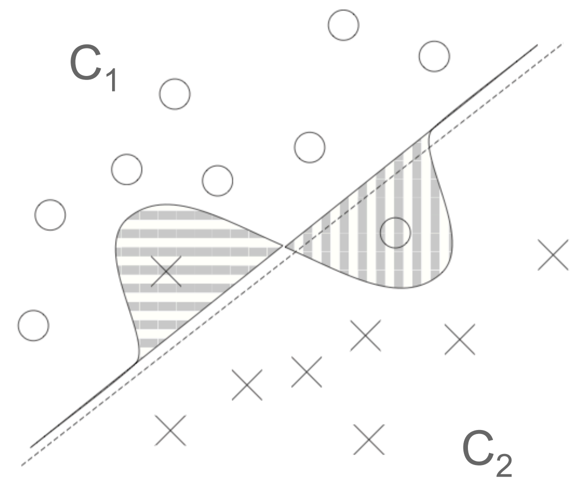
$$(x_1,\ldots,x_n)$$



# Generative and Discriminative Classification

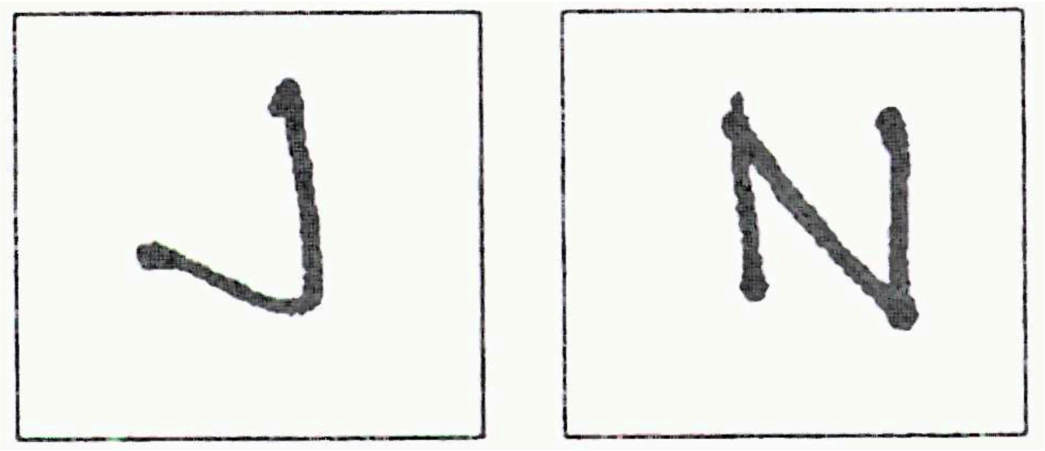
- Generative classification
  - θ : based on estimated class-conditioned feature distribution
  - allows to generate new samples
- Discriminative classification
  - θ : based on class boundaries
  - can be more accurate if the distribution is hard to estimate



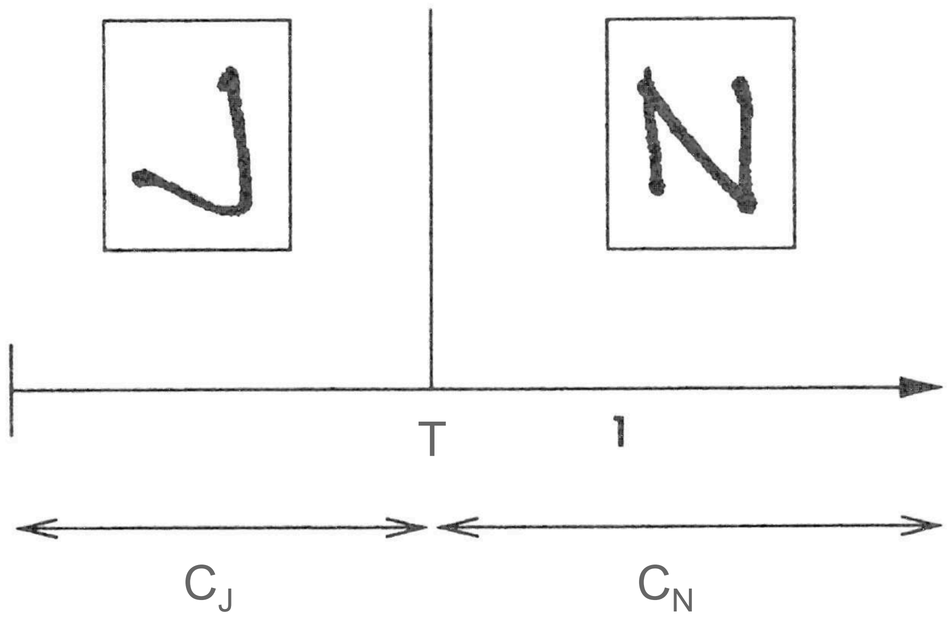
# Example: Character Recognition

- segmented images
- binarized: matrix of black/white pixels
- Y = { “J”, “N” }



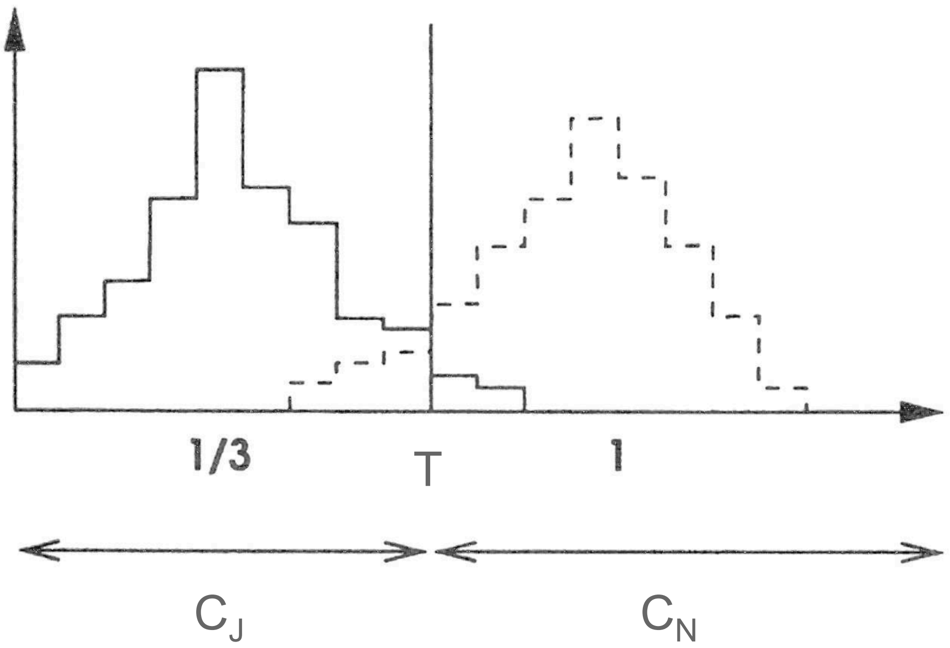
# Statistical Representation

- raw data
  - 25 x 20 = 500 pixels
  - $2^{500} \approx 10^{150}$ possible images
- x = #left / #right : ratio of black pixels in the left/right half



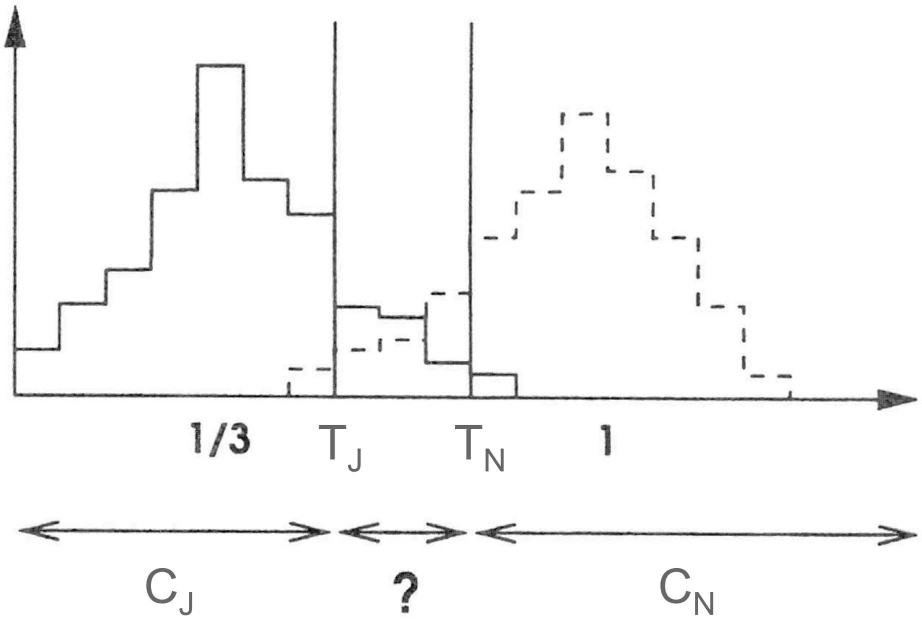
# Generative Classification

- $\theta$ : estimate the class-dependent feature distribution on intervals
  - histogram-based estimation using learning samples
- $f_\theta$ : apply the optimal threshold T with respect to $\theta$
  - minimize probability of misclassification



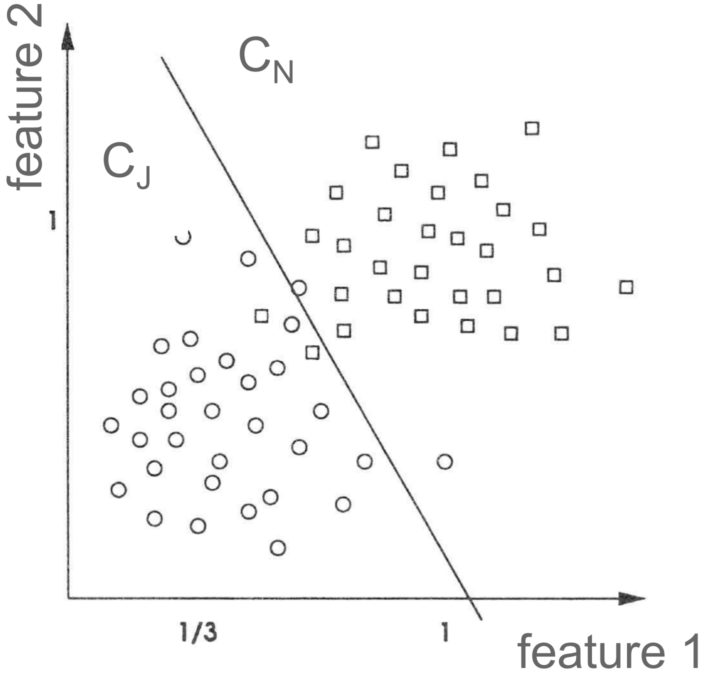
# Classification with Rejection

- The classifier can also reject inputs to improve the reliability.
- Rejected samples could then be classified by a human.
  - Example: postal address reading.

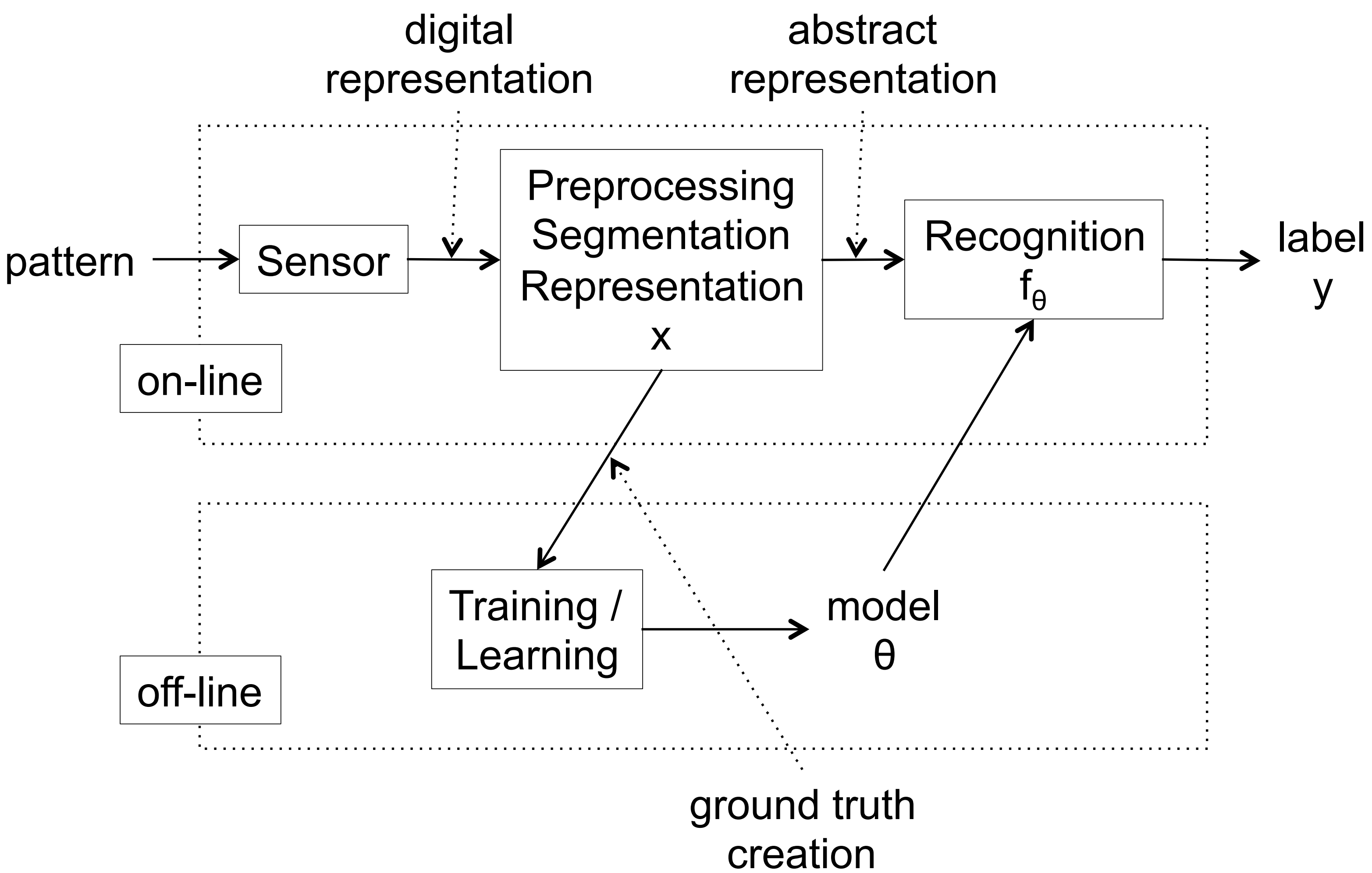


# Discriminative Classification

- $X = R^2$
  - second feature: #top / #bottom
- θ : linear class boundary ax + b
  - optimized with learning samples, e.g. using support vector machines (SVM), perceptrons
- $f_\theta$ : class assignment based on the sign of ax + b



# Pattern Recognition System



# K-Nearest Neighbor Classifier (KNN)

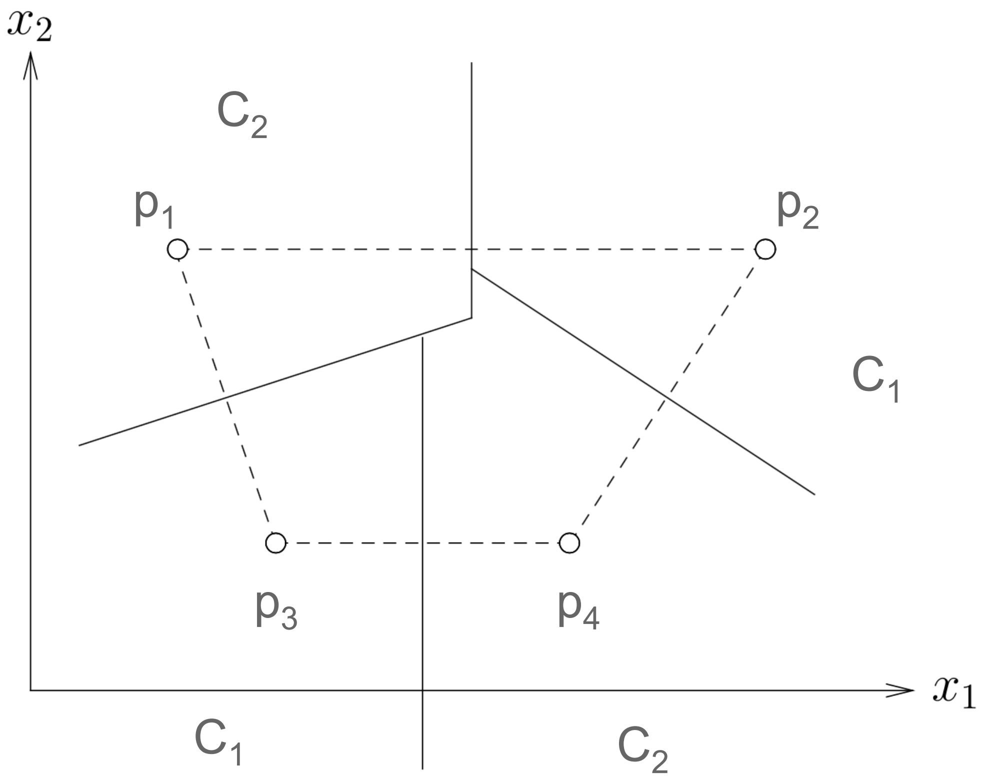
# Nearest Neighbor Classifier (NN)

- Discriminative classifier.
- In the case of statistical representation:
  - $X = R^n$
  - $\theta$ = { learning samples }
  - $f_\theta$ : assign the class label of the most similar known sample

$$f_\theta(x) = C_i \Leftrightarrow \underset{p \in \theta}{\operatorname{argmin}} \left\| x - p \right\| \in C_i$$

## Class Boundary

- NN rule leads to a Voronoi partition of the Euclidean space.
- Possible tie resolution on the boundary: rejection or random classification.



# Distance Measure

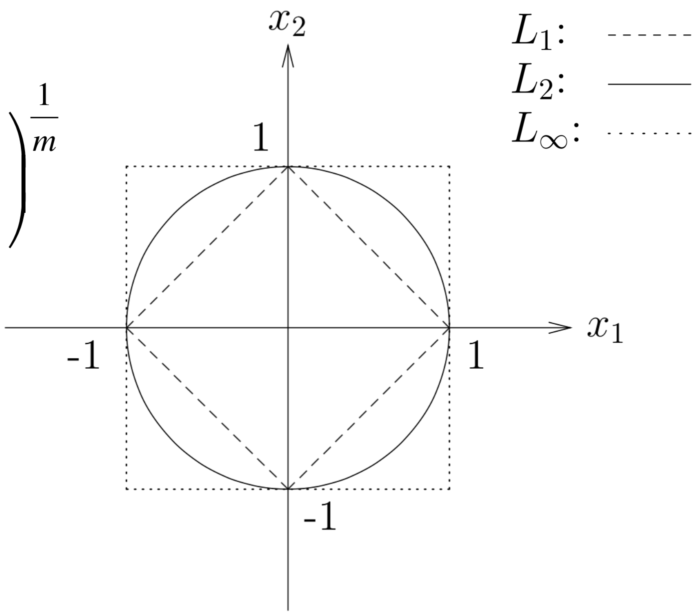
- NN not constrained to the Euclidean distance $L_2(x,p)$
  - $L_m(x,p)$: Minkowski distance
  - $L_1(x,p)$: Manhattan distance

$$f_\theta(x) = C_i \Leftrightarrow \underset{p \in \theta}{\operatorname{argmin}}\ L_m(x,p) \in C_i$$

$$L_m(x,p) = \left( \sum_{i=1}^{n} \left| x_i - p_i \right|^m \right)^{\frac{1}{m}}$$

$$L_1(x,p) = \sum_{i=1}^{n} \left| x_i - p_i \right|$$

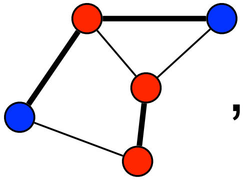
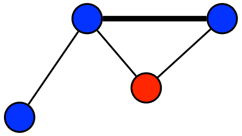
$$L_\infty(x,p) = \max_{i=1}^{n} \left| x_i - p_i \right|$$



# Structural Pattern Recognition

- NN widely used in structural pattern recognition
- X : strings, trees, graphs
- d(x,p) : dissimilarity measure, e.g.
  - string edit distance (Levenshtein distance)
  - graph edit distance

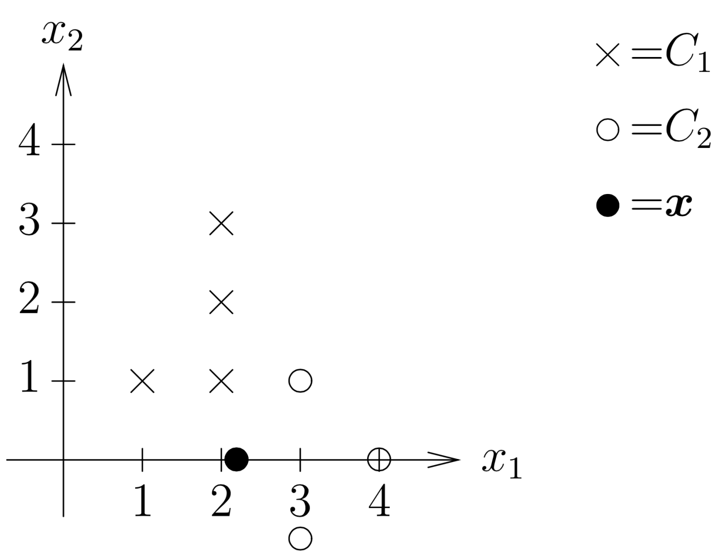
$$f_\theta(x) = C_i \Leftrightarrow \underset{p \in \theta}{\operatorname{argmin}}\, d(x,p) \in C_i$$

d(  ,  )

# K-Nearest Neighbor Classifier (KNN)

- $f_\theta$ : assign the *most frequent* class label among the *K most similar* known samples
- Possible tie resolution:
  - Assign class of the 1-NN.
  - Assign class with the closest center (mean vector).
- High computational complexity:
  - Efficient indexing needed for large-scale problems.

1-NN: $C_1$
3-NN: $C_2$



# Condensing

- Reduce training set by focusing on significant elements.

**Require:** training set $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$
**Ensure:** condensed training set $R \subseteq S$

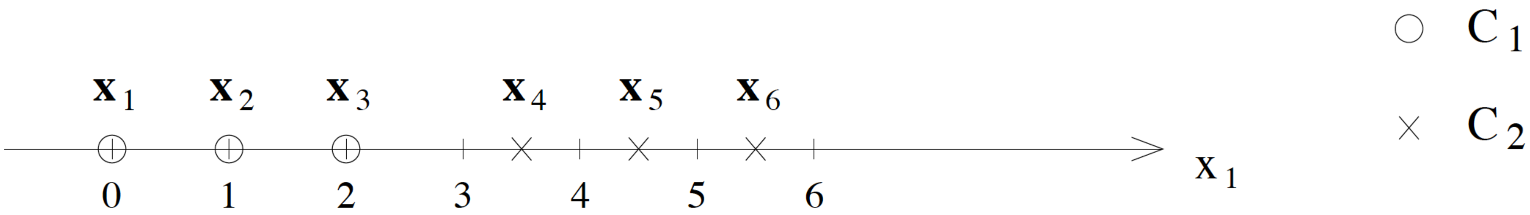
$R = \{\mathbf{x}_1\}$; $S = S - \{\mathbf{x}_1\}$; changes = true
**while** changes = true **do**
  changes = false
  **for all** $\mathbf{x}_i \in S$ **do**
    classify $\mathbf{x}_i$ based on $R$ with 1NN
    **if** $\mathbf{x}_i$ was wrongly classified **then**
      $R = R \cup \{\mathbf{x}_i\}$ *# add significant element*
      $S = S - \{\mathbf{x}_i\}$
      changes = true
    **end if**
  **end for**
**end while**

# Editing

- Reduce training set by removing outliers.

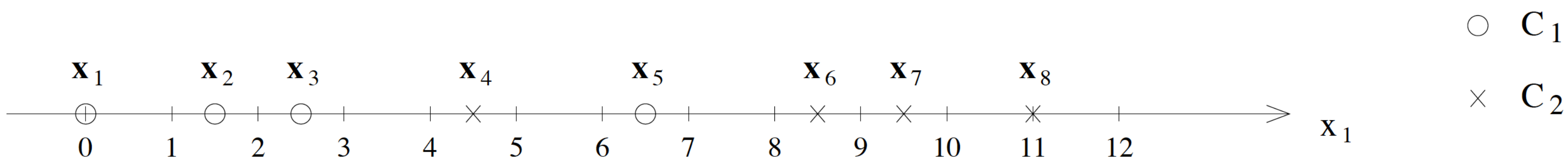
**Require:** training set $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$
**Ensure:** edited training set $R \subseteq S$
1: **for all** $i = 1, \ldots, N$ **do**
2: classify $\mathbf{x}_i$ based on $S - \{\mathbf{x}_i\}$ with 3NN
3: **if** $\mathbf{x}_i$ was wrongly classified **then**
4: mark $\mathbf{x}_i$ for deletion *# remove outlier*
5: **end if**
6: **end for**
7: $R = S - \{\mathbf{x} | \mathbf{x} \text{ was marked for deletion}\}$

# Example 1




- Condensing:
  - First iteration: R = $\{x_1,x_4\}$
  - Second iteration: R = $\{x_1,x_3,x_4\}$
- Editing:
  - Marked: none
  - Result: R = $\{x_1,x_2,x_3,x_4,x_5,x_6\}$

# Example 2




- Condensing:
  - First iteration: R = {$x_1$,$x_4$,$x_5$,$x_6$}
  - Second iteration: R = {$x_1$,$x_3$,$x_4$,$x_5$,$x_6$}
- Editing:
  - Marked: $x_4$,$x_5$
  - Result: R = {$x_1$,$x_2$,$x_3$,$x_6$,$x_7$,$x_8$}

# Example 3