Approaches in Cognitive Science

Machine Learning as an Application Field of Methods in Cognitive Science

Bastian Tenbergen
(tenberge@oswego.edu)

Human Computer Interaction M.A. Program
State University of New York at Oswego

formerly
School of Human Sciences, Cognitive Science B.Sc. Program
University of Osnabrück, Germany

Outline

- 1. Introduction
  - 1.1 What is Cognitive Science?
  - 1.2 CogSci Approaches
- 2. Machine Learning Methodology
  - 2.1 DT Algorithms
  - 2.2 ILP Systems
  - 2.3 Hypothesis Searching
- 3. Computational Neuroscience
  - 3.1 Multi-Layer Perceptrons
  - 3.2 Kohonen Networks
- 4. A Research Example...

1.1 What is Cognitive Science?

- CogSci aims at understanding
  - Cognitive processes, learning
  - Perception, thinking
  - Language and motor functions

- A research field with many different sectors
  - One single approach would not cover every notion of the brain

  Neurobiology, Neurobiopsychology, Psychology, Artificial Intelligence, Philosophy, Linguistic, Computational Neuroscience...

1.1 What is Cognitive Science?

- The brain is the most complex (known) system in the universe
- The brain is the center of all these cognitive functions
- We need to know how it works, before we can copy it!
- Ethic dilemma:
  - Should we create a **real** AI, even if we could?
  - AI serving humans: Is that slavery?
  - If you are interested in that, please attend a Philosophy lecture ;-)

1.2 CogSci Approaches

- Neurobiopsychology
- Cognitive Psychology
  - Use brain imaging techniques
- Philosophy
  - Mind/Body Problem
- Linguistics
  - Develop theory of language
- Computer Science
- Machine Learning / Artificial Intelligence
  - Make a computer “understand”
- Computational Neuroscience
  - Model neuronal structure

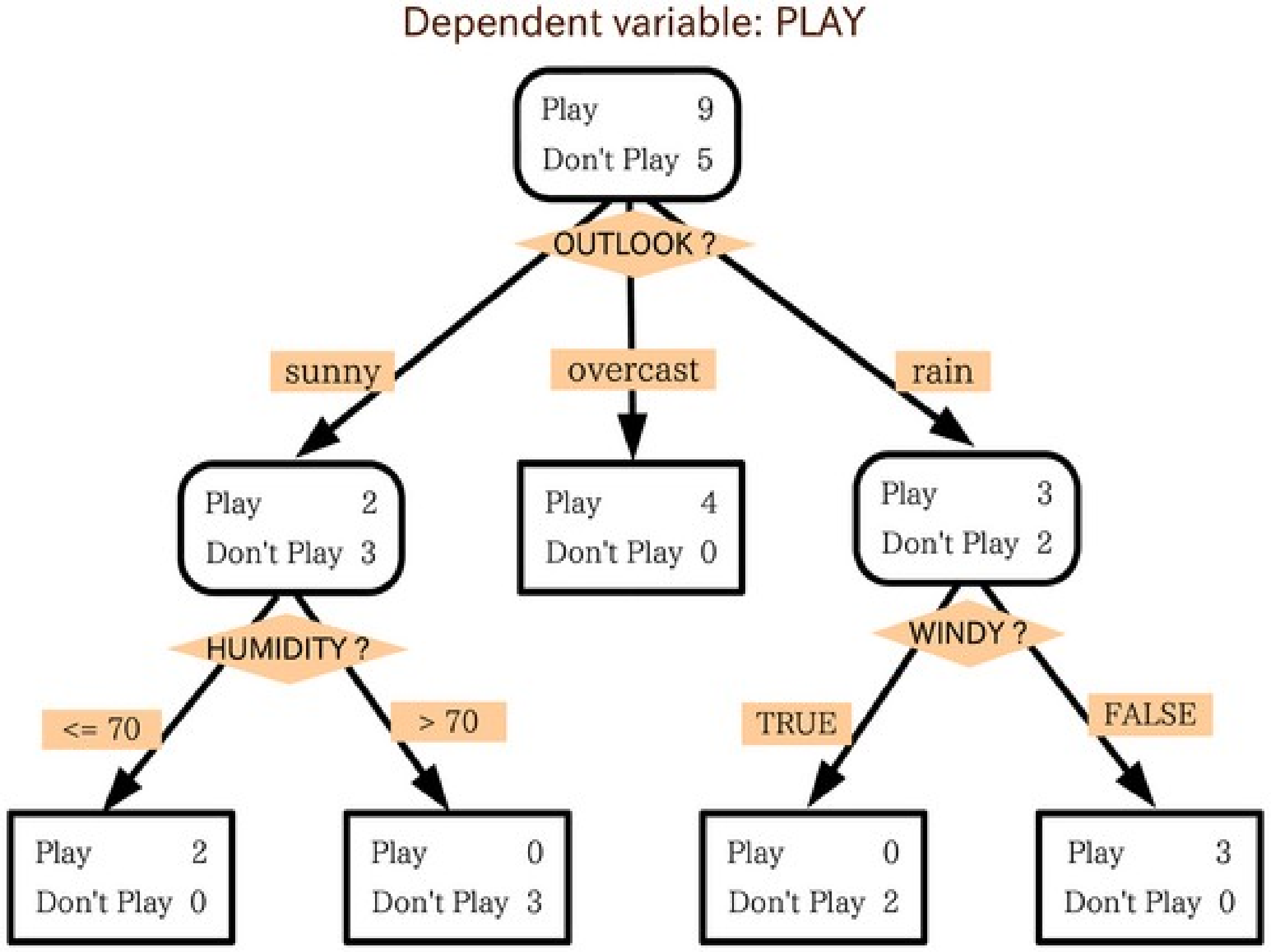
2. Machine Learning

- In the universe, there are
  - entities and
  - relations between entities
- Goal of ML: make a machine reason about the universe
- A computer cannot generalize or abstract
  - How can it actually “learn” then?
- -> “furniture”-problem

2.1 Machine Learning Methodology

- Decision Tree Algorithms
  - Examples:
    - CAL2
    - C4.5
    - ID3
- Learn general rules from given training sets.

  “IF
  outlook == sunny AND
  humidity == low AND
  wind == weak
  THEN
  play tennis == yes”


Dependent variable: PLAY
Play 9
Don't Play 5
OUTLOOK ?
sunny
overcast
rain
Play 2
Don't Play 3
Play 4
Don't Play 0
Play 3
Don't Play 2
HUMIDITY ?
WINDY ?
<= 70
> 70
TRUE
FALSE
Play 2
Don't Play 0
Play 0
Don't Play 3
Play 0
Don't Play 2
Play 3
Don't Play 0

2.1 Machine Learning Methodology

- Inductive Logical Programming Systems
  - Examples:
    - FOIL
    - CProgol

- Generate rules from training sets, either top-down (refinement) or bottom-up (generalization)

  mother(maria, anna).
  female(anna).
  female(maria).

  Learn the predicate
  daughter(X,Y).

2.1 Machine Learning Methodology

- Hypothesis Space Exploration Techniques
  - Examples:
    - Search Algorithms (Beam, Hill-climbing, etc)
    - Sort Algorithms (Quicksort, Mergesort, Radixsort, etc)
- Idea: Go through hypothesis space by searching or sorting existing hypothesis for optimal solutions or generate solutions and measure quality.
- Genetic Algorithms / Evolutionary Programming

2.1 Machine Learning Methodology

- Genetic Algorithms / Evolutionary Programming
- Idea:
  - Generate a generation of hypothesis
  - Evaluate quality of every hypothesis
  - Take the n best hypothesis
  - And generate new generation of hypothesis
  - Repeat until optimal solution found or early stopping criterion fulfilled.
- Hill-climbing like approach
- Pros? Cons?

3. Computational Neuroscience

- Solve problems in CS or in Real-World by modeling neuronal architectures
- Neuronal Structures are of artificial nature (no real-tissue! ;-))
- Tiniest constituent: Perceptrons
- “Neural Networks” are a perfect example of “Emergentialism”:

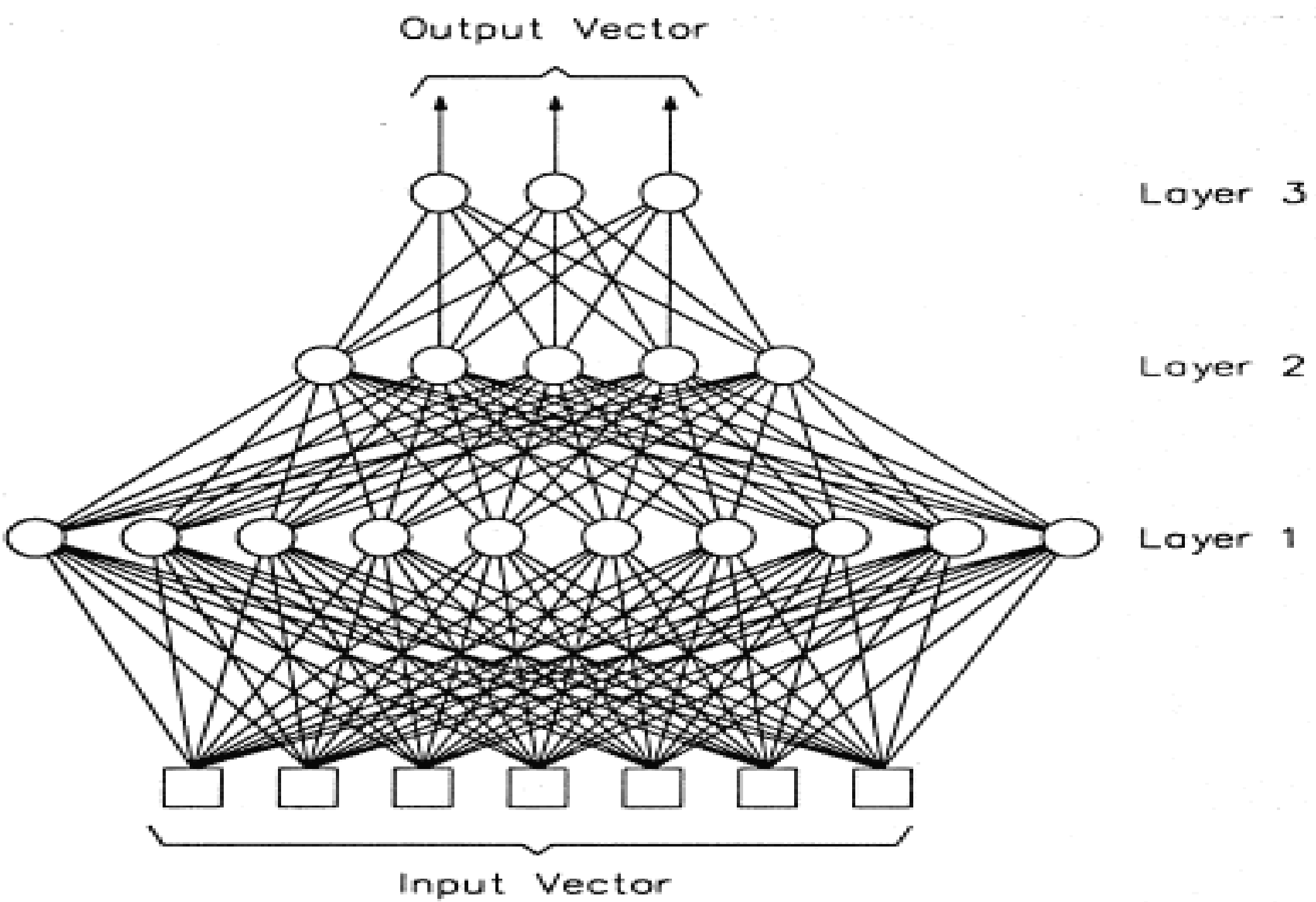
  The whole is more than the sum of its parts!

3.1 Multi-Layer Perceptrons

- m input neurons
- n output neurons
- h number of layers between input- and output-layer with (hidden layers)
- k(l) neurons in each layer (l)

- Network is a directed, feed-forward network without cycles
  (-> acyclic, mono-directed graphs)

- Take m-dimensional input patterns
- Propagate the input patterns through net
- Weights of perceptron change persistently with each pattern


Output Vector
Layer 3
Layer 2
Layer 1
Input Vector

3.2 Kohonen Networks

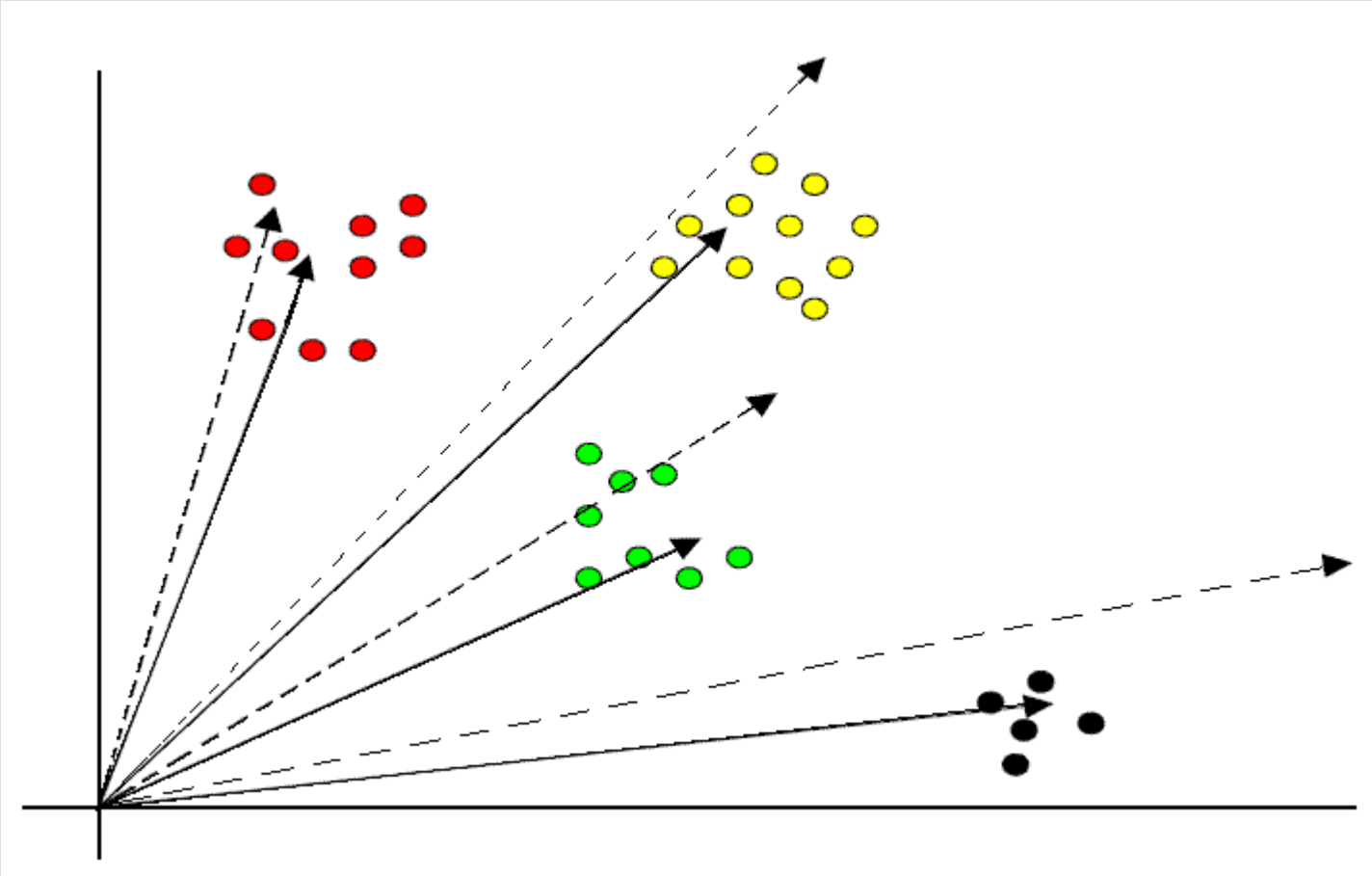
- Also known as: Self-Organizing Maps

- m X n – matrix
- Every neuron receives input
- KWTA-output principle
- Only winner neuron(s) and neighbours updated
- Updated neurons updated according to current pattern

  -> map changes topography according to input patterns

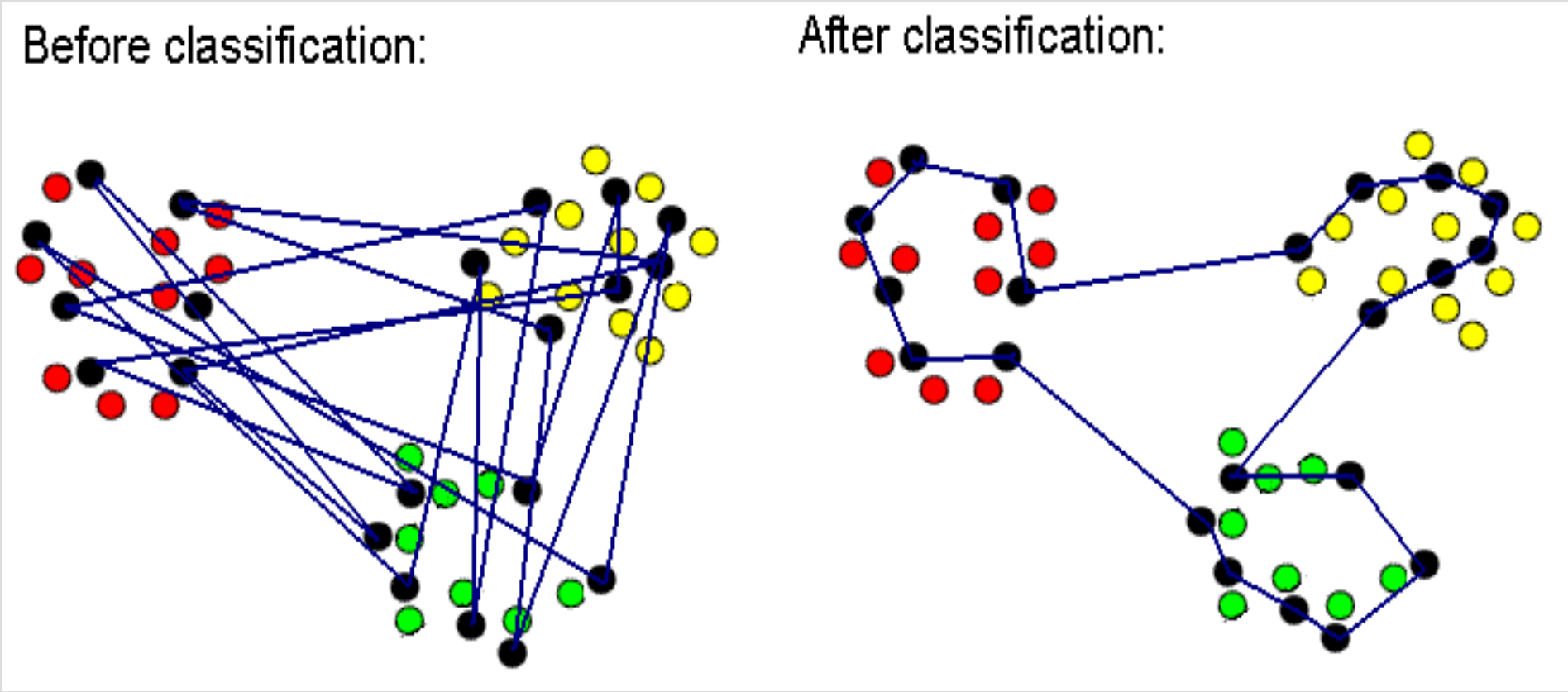
3.2 Kohonen Networks

- Prototyping
  - “Pull” vectors into direction of patterns



3.2 Kohonen Networks

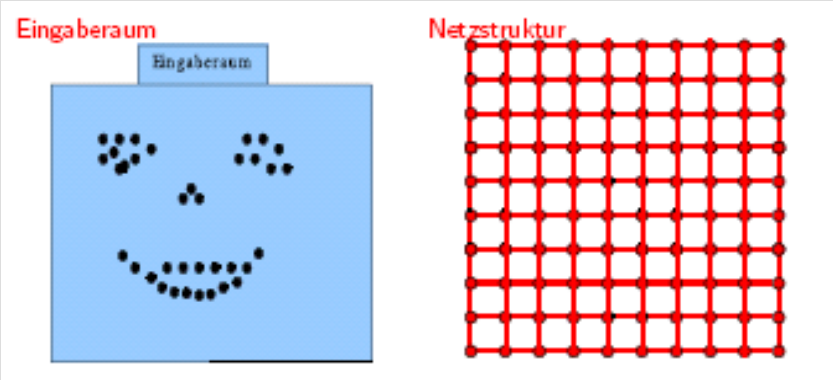
- Clustering
  - Lay network over input patterns

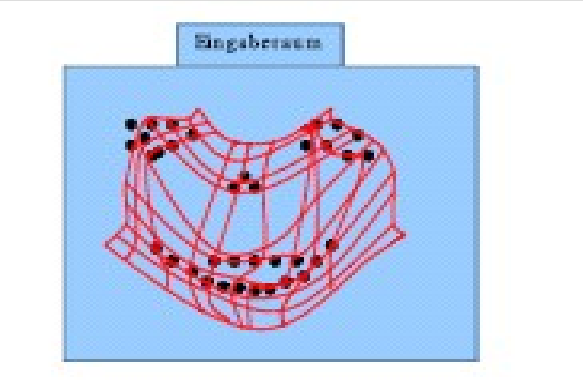


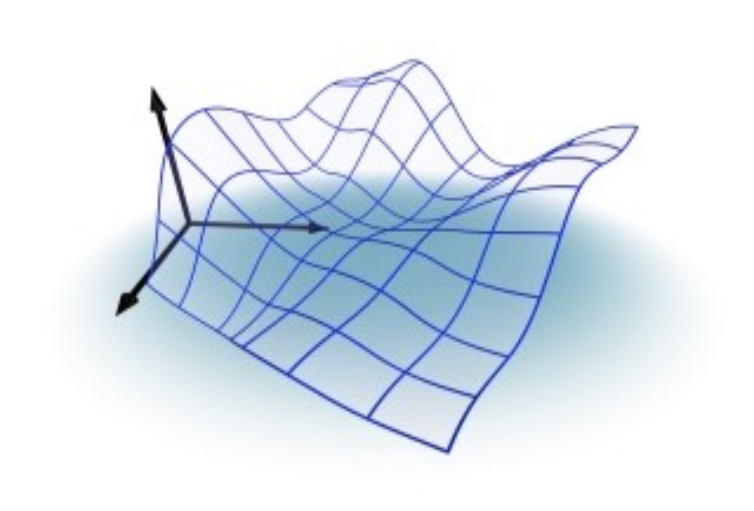
here: Travelling-Salesman-Approximation

3.2 Kohonen Networks

- Real cool things can happen...

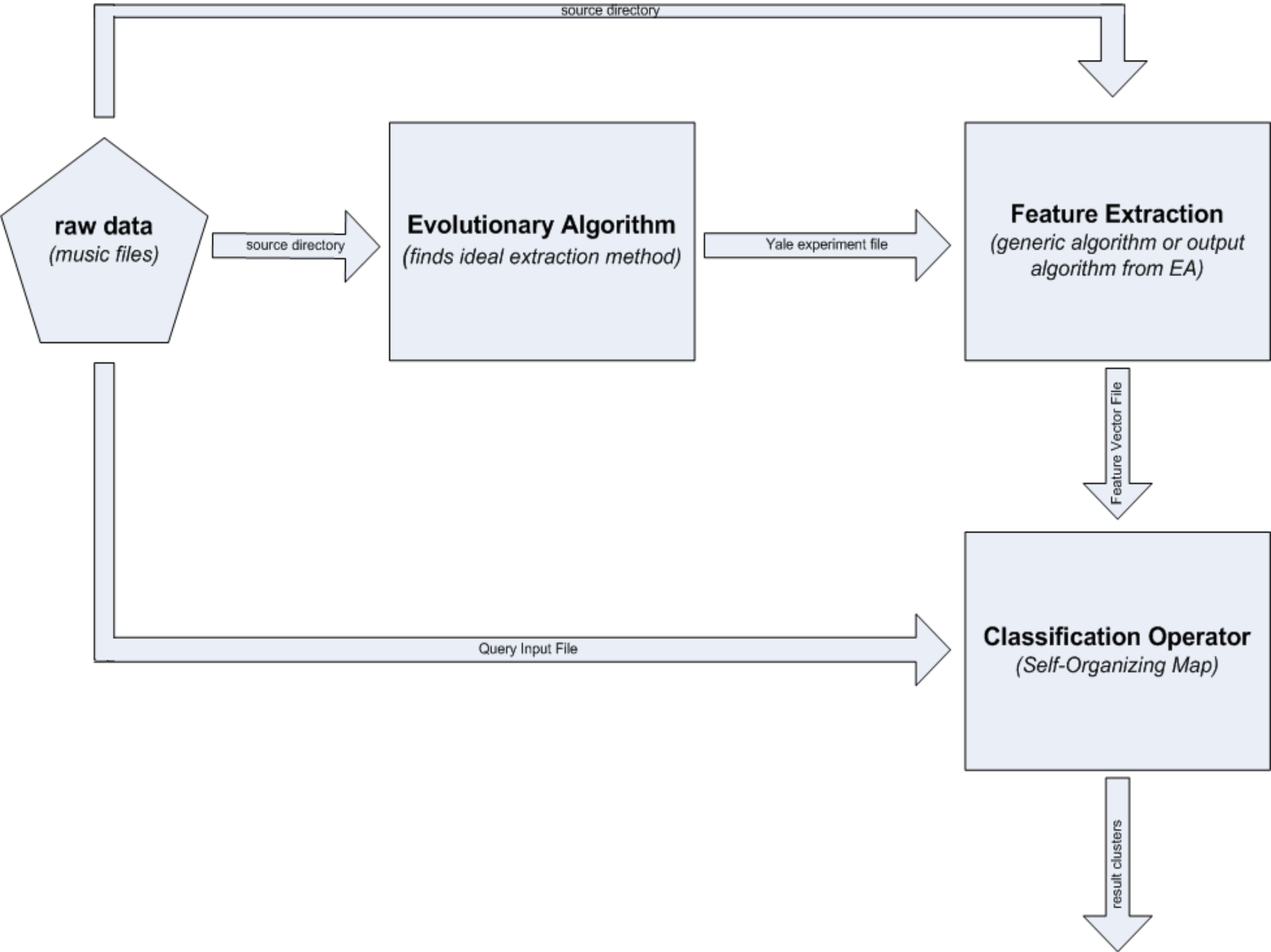






4. A Research Example...

- Music Similarity Clustering According to User Preference
- Extract audio features from mp3-files
  - use GA to find optimal extraction algorithm
  - extracted features depend on corpus
- Cluster feature vectors with SOM
  - “perceptionally” similar files in same cluster
  - represented by same prototype in map
- Allows for structuring **any** music database according to groups that are not known a priori


source directory
raw data
(music files)
source directory
Evolutionary Algorithm
(finds ideal extraction method)
Yale experiment file
Feature Extraction
(generic algorithm or output algorithm from EA)
Feature Vector File
Query Input File
Classification Operator
(Self-Organizing Map)
result clusters

4. A Research Example...

- Very good results:
  - 75% accuracy
  - accuracy improves with larger copora
  - problem: GA did not contribute to optimal clustering performance
    - Same performance with GA-developed extraction algorithms as with generic feature sets
- Can help to visualize music similarity
  - research in music theory
- Generate playlists content-based on user preferrence
  - not on basis of meta-data
- Disambiguate and structure music databases
  - On-the-fly analysis of new patterns supported!

Thank you for your attention!

Any questions?