Advanced Techniques for Mobile Robotics

Clustering & EM

Wolfram Burgard, Cyrill Stachniss,

Kai Arras, Maren Bennewitz

UNI FREIBURG

Motivation

- Common technique for statistical data analysis to detect structure (machine learning, data mining, pattern recognition, ...)
- Classification of a data set into subsets (clusters)
- Efficient representation of data

Example Application: Image Segmentation / Data Compression

- Each pixel is a point in the RGB space
- Represent the image using only K colors
- The corresponding colors are obtained by a clustering of the input data

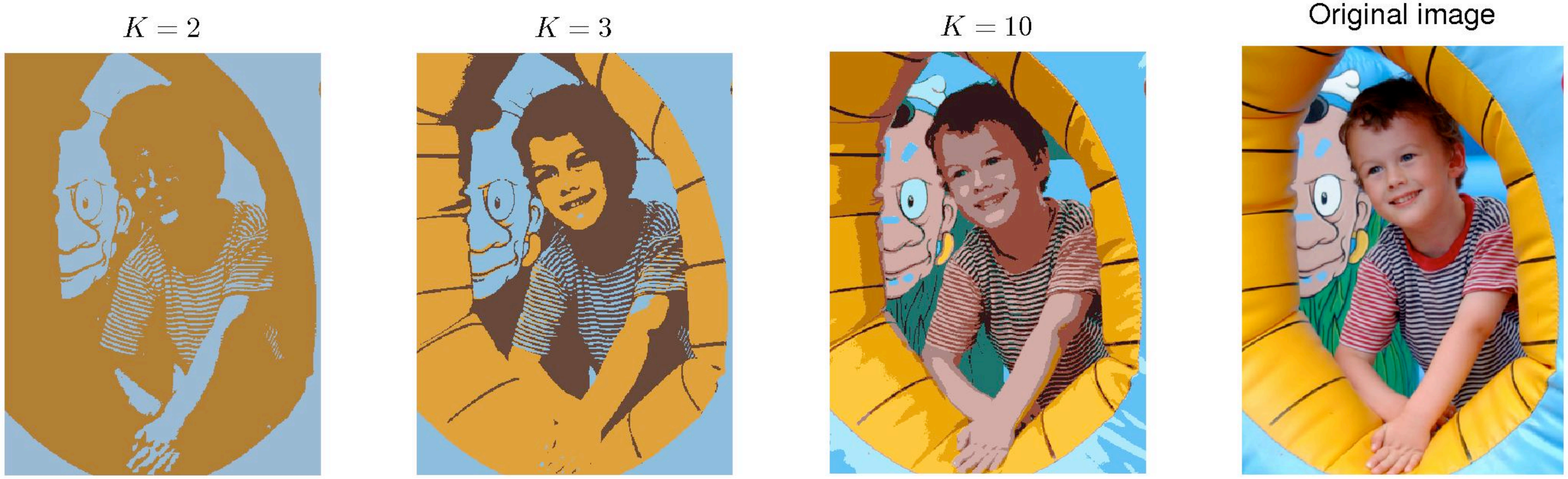


image source: C. M. Bishop

Clustering

- Needed: Distance function (similarity / dissimilarity), e.g., Euclidian distance
- Objectives
  - Maximize inter-clusters distance
  - Minimize intra-clusters distance
- The quality of the clustering result depends on
  - The clustering algorithm
  - The distance function
  - The application (data)

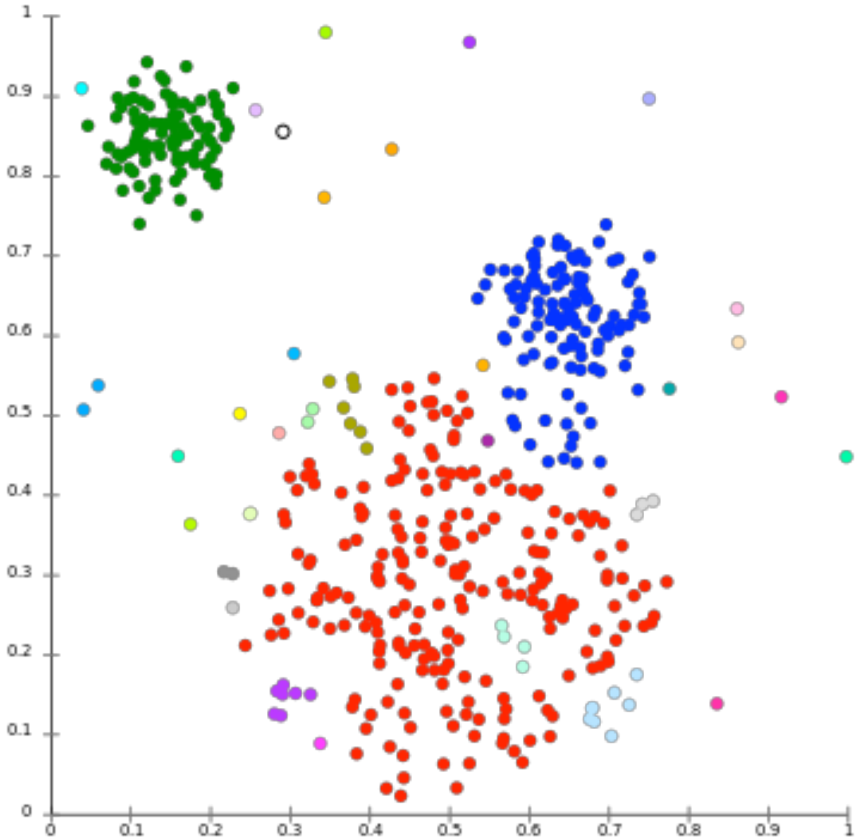
Types of Clustering

- Hierarchical Clustering
  - Agglomerative Clustering (bottom up)
  - Divisive Clustering (top-down)
- Partitional Clustering
  - K-Means Clustering (hard & soft)
  - Gaussian Mixture Models

Hierarchical Clustering

- Connects data points to clusters / separates points from clusters based on their distance
- In addition to the distance function, one also needs to specify the linkage criterion (which clusters to merge / separate)
- Produces a hierarchy of partitionings one has to choose from

Example: Divisive Clustering



- Data generated from three Gaussians
- Single-linkage (distance between the two closest elements of different clusters)
- Currently, 35 clusters
- The biggest cluster starts fragmenting into smaller parts

image source: wikipedia

Weaknesses
Hierarchical Clustering

- Once connected, clusters cannot be partitioned again (agglomerative clustering)
- The order in which clusters are formed is crucial (depends on the linkage criterion)
- Sensitive to outliers (either leads to additional clusters or can cause other clusters to merge)
- Unclear which partitioning to choose
- Too slow for large data sets

K-Means Clustering

- Clusters are represented by centroids, which do not need to be members of the cluster
- Partitions the data into *k* clusters (*k* needs to be specified by the user!)
- Objective: Find the *k* cluster centers and assign the data points to the nearest cluster, such that the squared distances from the cluster centroids are minimized

K-Means Clustering Algorithm (Informally)

- Iterative procedure
- Initialization: Choose *k* arbitrary centroids (cluster means)
- Repeat until convergence
  - Assign each data point to the closest centroid
  - Adjust the centroids of the clusters to the mean of the data points assigned to them

K-Means Clustering (More Formally)

- Find *k* reference vectors (centroids) $m_j, j = 1, \ldots, k$ that best explain the data ***X***
- Assign data vectors to the nearest (most similar) reference vector $\boldsymbol{m_i}$

$$\|\mathbf{x}^t - \mathbf{m}_i\| = \min_j \|\mathbf{x}^t - \mathbf{m}_j\|$$

r-dimensional data vector
in a real-valued space

reference vector
(centroid / mean)

K-Means Clustering

- Optimize the following objective function:

$$R = \sum_t \sum_i b_i^t \|\mathbf{x}^t - \mathbf{m}_i\|^2$$

with $$b_i^t = \begin{cases} 1 & \text{if } i = \arg\min_j \left\|\mathbf{x}^t - \mathbf{m}_j\right\| \\ 0 & \text{otherwise} \end{cases}$$

- Find reference vectors that maximize *R*
- Taking the derivative with respect to $\mathbf{m}_i$ and setting it to 0 leads to:

$$\mathbf{m}_i = \frac{\sum_t b_i^t \mathbf{x}^t}{\sum_t b_i^t}$$

K-Means Algorithm

Initialize $\boldsymbol{m}_i, i = 1, \ldots, k$, for example, to k random $\boldsymbol{x}^t$

Repeat

For all $\boldsymbol{x}^t \in \mathcal{X}$

$$b_i^t \leftarrow \begin{cases} 1 & \text{if } \|\boldsymbol{x}^t - \boldsymbol{m}_i\| = \min_j \|\boldsymbol{x}^t - \boldsymbol{m}_j\| \\ 0 & \text{otherwise} \end{cases}$$

For all $\boldsymbol{m}_i, i = 1, \ldots, k$

$$\boldsymbol{m}_i \leftarrow \sum_t b_i^t \boldsymbol{x}^t / \sum_t b_i^t$$

Until $\boldsymbol{m}_i$ converge

Re-compute the cluster means $\boldsymbol{m}_i$ using the current cluster memberships

Assign each $\boldsymbol{x}^t$ to the closest cluster

K-Means Example (1)

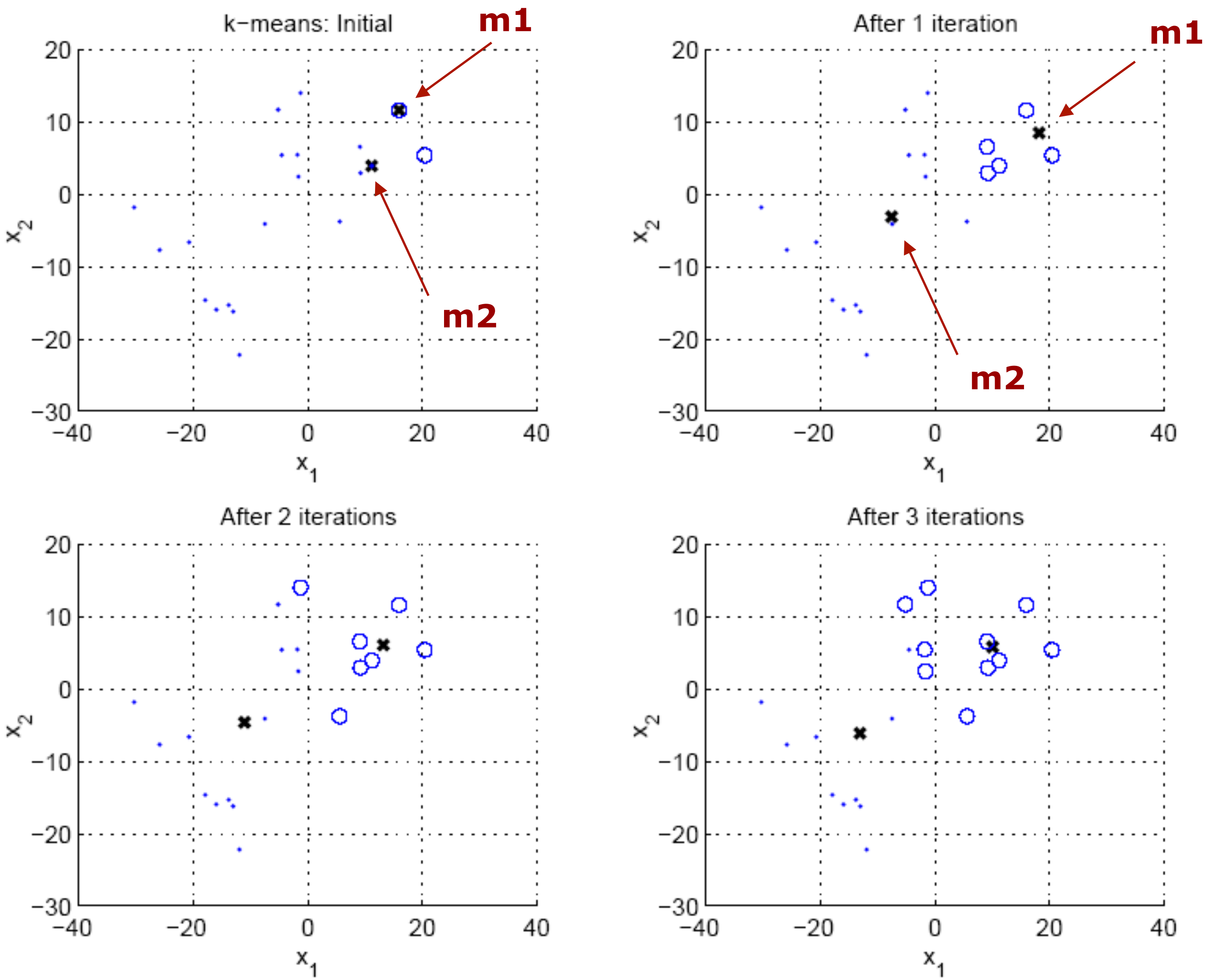


image source: Alpaydin, Introduction to Machine Learning

K-Means Example (2)

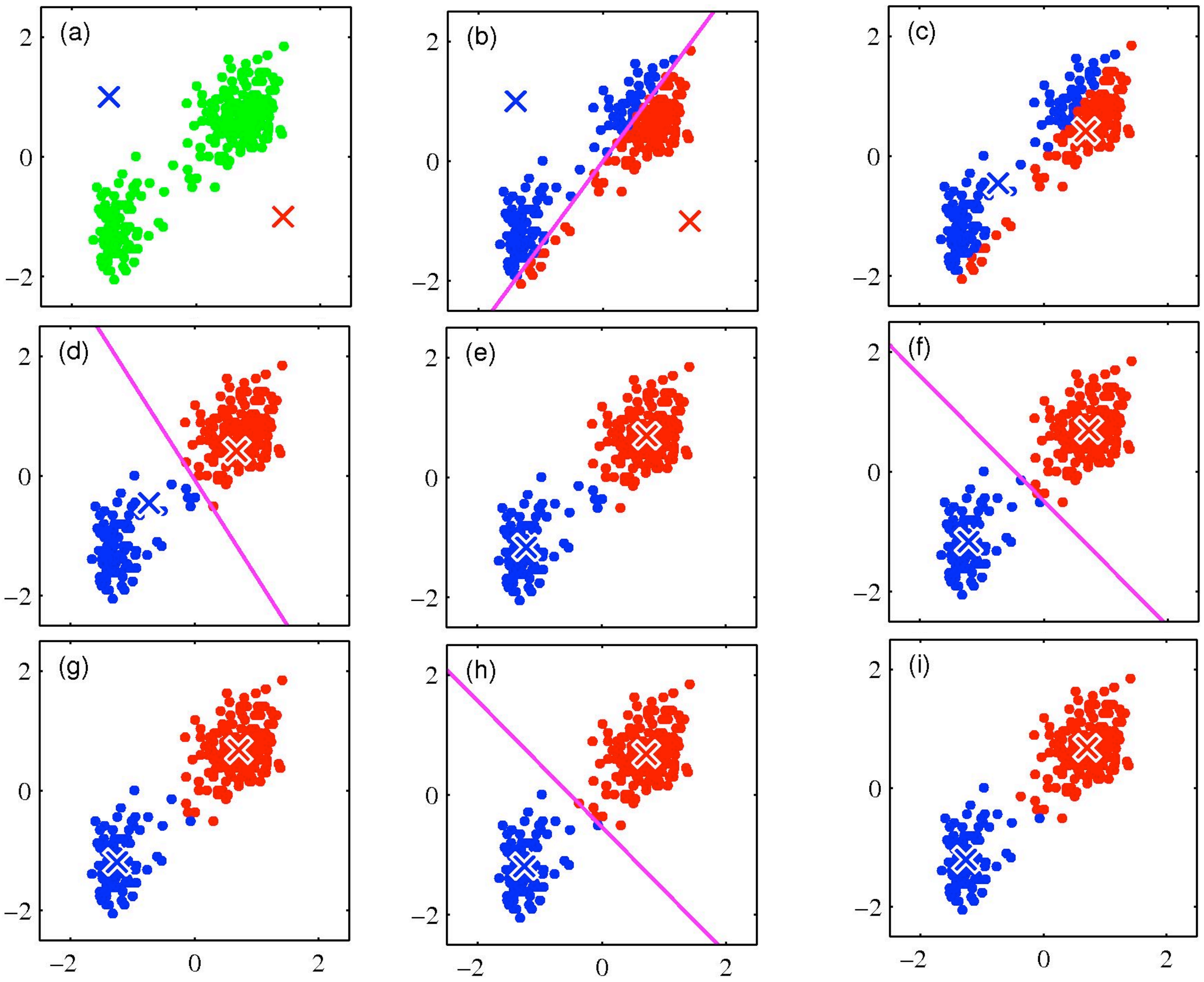


image source: Bishop, Pattern Recognition and Machine Learning

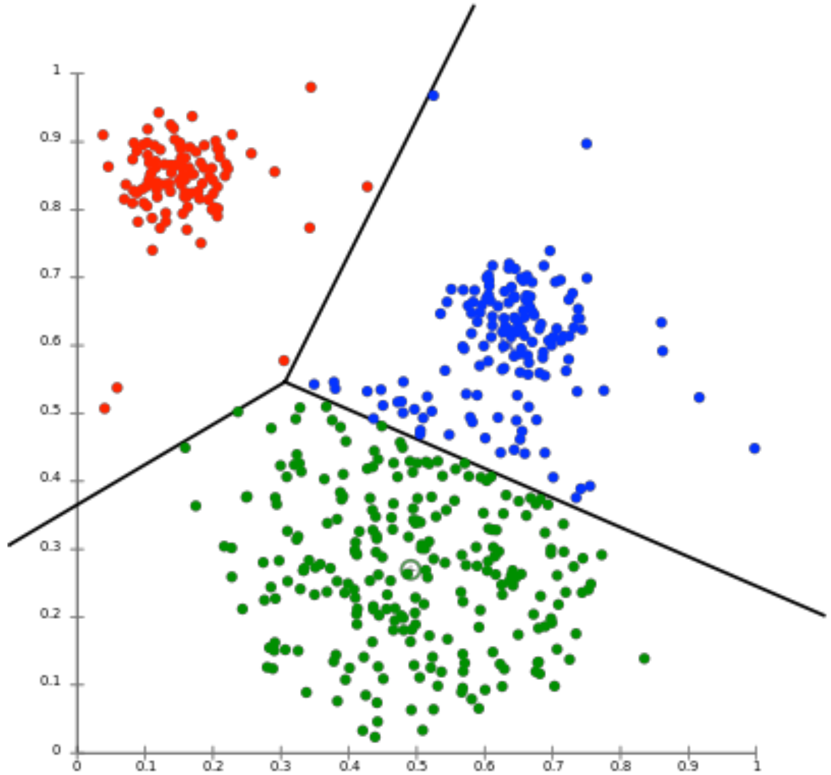
Strength of K-Means

- Easy to understand and to implement
- Efficient O(*nkt*)
  n = #iterations, *k* = #clusters, *t* = #data points
- Converges quickly to a local optimum
- Most popular clustering algorithm

Weaknesses of K-Means

- User needs to specify #clusters (*k*)
  Later introduced: Method to estimate *k*
- Sensitive to initialization
  Strategy: Use different seeds
- Sensitive to outliers since all data points contribute equally to the mean
  Strategy: Try to eliminate outliers
- Prefers clusters of approximately similar size (objects are assigned to the nearest centroid)

Example: Problem with K-Means



- Dataset generated from three Gaussians
- K-means prefers equally sized clusters

image source: wikipedia

Soft Assignments

- So far, each data point was assigned to exactly one cluster
- A variant called soft k-means allows for making fuzzy assignments
- Data points are assigned to clusters with certain probabilities

Soft K-Means (Informally)

- Choose *k* clusters centroids
- Assign randomly to each data point probabilities for being in the clusters
- Repeat until convergence
  - Compute the centroid for each cluster taking into account the membership probabilities
  - For each data point, re-compute its membership probabilities based on the distance to the centroids

Soft K-Means Clustering (1)

- Each data t point is given a soft assignment to all means k:

$$c_{tk} = \frac{\exp(-\beta\,||\mathbf{x}_t - \mathbf{m}_k||^2)}{\sum_i \exp(-\beta\,||\mathbf{x}_t - \mathbf{m}_i||^2)}, \; \sum_k c_{tk} = 1$$

- β is a “stiffness” parameter and plays a crucial role

Soft K-Means Clustering (2)

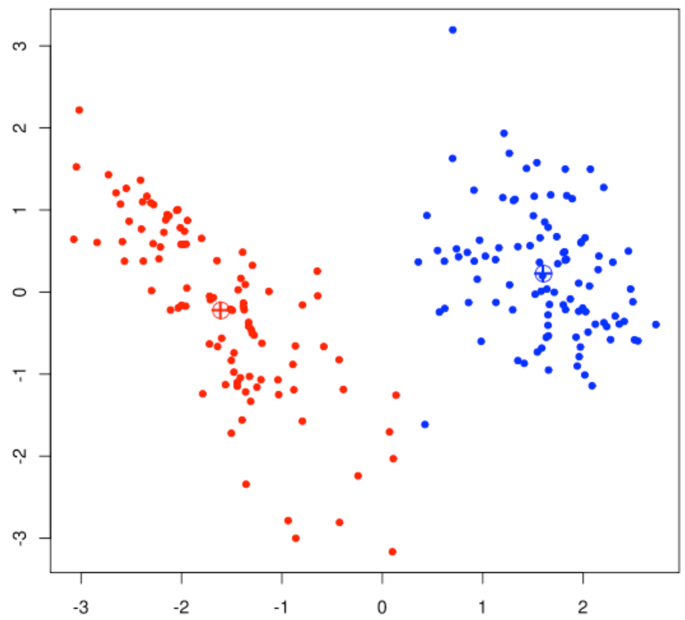
- Soft k-means optimizes the following objective function:

$$J = \sum_t \sum_k c_{tk} \|\mathbf{x}_t - \mathbf{m}_k\|^2$$

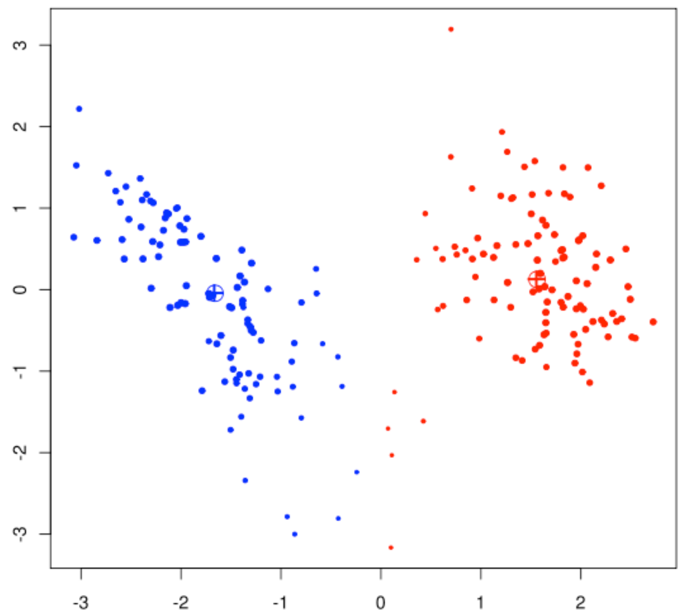
- Accordingly, the means are updated:

$$\mathbf{m}_k = \frac{\sum_t c_{tk}\mathbf{x}_t}{\sum_t c_{tk}}$$

Example: Hard vs Soft K-Means



hard k-means



soft k-means

image source: wikipedia

Properties of Soft K-Means Clustering

- Points between clusters get assigned to both of them (accounts for uncertainty)
- Additional parameter β
- Same problem as *k*-means: Local optimum; the result depends on the initial choice of membership probabilities
- Extension: Clusters with varying shapes can be treated in a probabilistic framework (mixtures of Gaussians, see next lecture)

Similarity of Soft K-Means and Expectation Maximization (EM)

- EM is a general method for finding the maximum-likelihood estimate of the parameters of the underlying distribution
- In case of Gaussian distributions, the parameters are the means (and variances)
- EM finds the model parameters that maximize the likelihood of the given, incomplete data

Expectation Maximization (EM) Basic Definitions

- Two sets of random variables
  - Observed data set *d*
  - Hidden variables *c*
    (assignment of data points to clusters)
- Since the joint likelihood (incl. the hidden variables!) cannot be determined, we work with its expectation

Expectation (Expected Value)

- … is the integral of the random variable with respect to its probability measure
- For discrete random variables this is equivalent to the probability-weighted sum of the possible values
- $E[x] = E_x[x] = \int_x xp(x)\, dx$
- $E_{x,y}[x] = \int_{x,y} xp(x,y)\, dx\, dy$
- $E_{x|y}[x] = E_x[x|y] = E_x[x|y] = \int_x xp(x \mid y)\, dx$
- $E_{x|y}[g(x)] = E_x[g(x) \mid y] = \int_x g(x)p(x \mid y)\, dx$

Expected Data Likelihood

- Observed data $d = \{d_1, \ldots, d_I\}$
- Correspondence variables (hidden)

$$c = \{c_1, \ldots, c_I\}$$

- Joint likelihood of *d* and *c* given model θ

$$P(d, c \mid \theta) = \prod_{i=1}^{I} P(d_i, c_i \mid \theta)$$

$$\ln P(d, c \mid \theta) = \sum_{i=1}^{I} \ln P(d_i, c_i \mid \theta)$$

- Since the values of *c* are hidden, optimize the expected value of the log likelihood

$$E_c[\ln P(d, c \mid \theta) \mid \theta, d] = E_c[\sum_{i=1}^{I} \ln P(d_i, c_i \mid \theta) \mid \theta, d]$$

Expectation Maximization

- Optimizing the expected log likelihood is usually not easy
- EM iteratively maximizes log likelihood functions
- EM generates a sequence of models $\theta^{[1]}, \theta^{[2]}, \ldots$ of increasing log likelihood
- EM converges to a (local) optimum

Expectation Maximization

- Use so-called Q-function to find the model with the maximum expected data likelihood
- Define the expected data log likelihood as a function of θ

$$Q(\theta \mid \theta^{[j]}) = E_c[\ln P(d, c \mid \theta) \mid \theta^{[j]}, d]$$

new parameters we optimize to increase Q

current parameter estimates used to evaluate the expectation

Expected value:

$$E_c[\ln P(d, c \mid \theta) \mid \theta^{[j]}, d] = \int_c \ln P(d, c \mid \theta) P(c \mid \theta^{[j]}, d) dc$$

Expectation Maximization

$$Q(\theta \mid \theta^{[j]}) = E_c[\ln P(d, c \mid \theta) \mid \theta^{[j]}, d]$$

Expectation (E) step

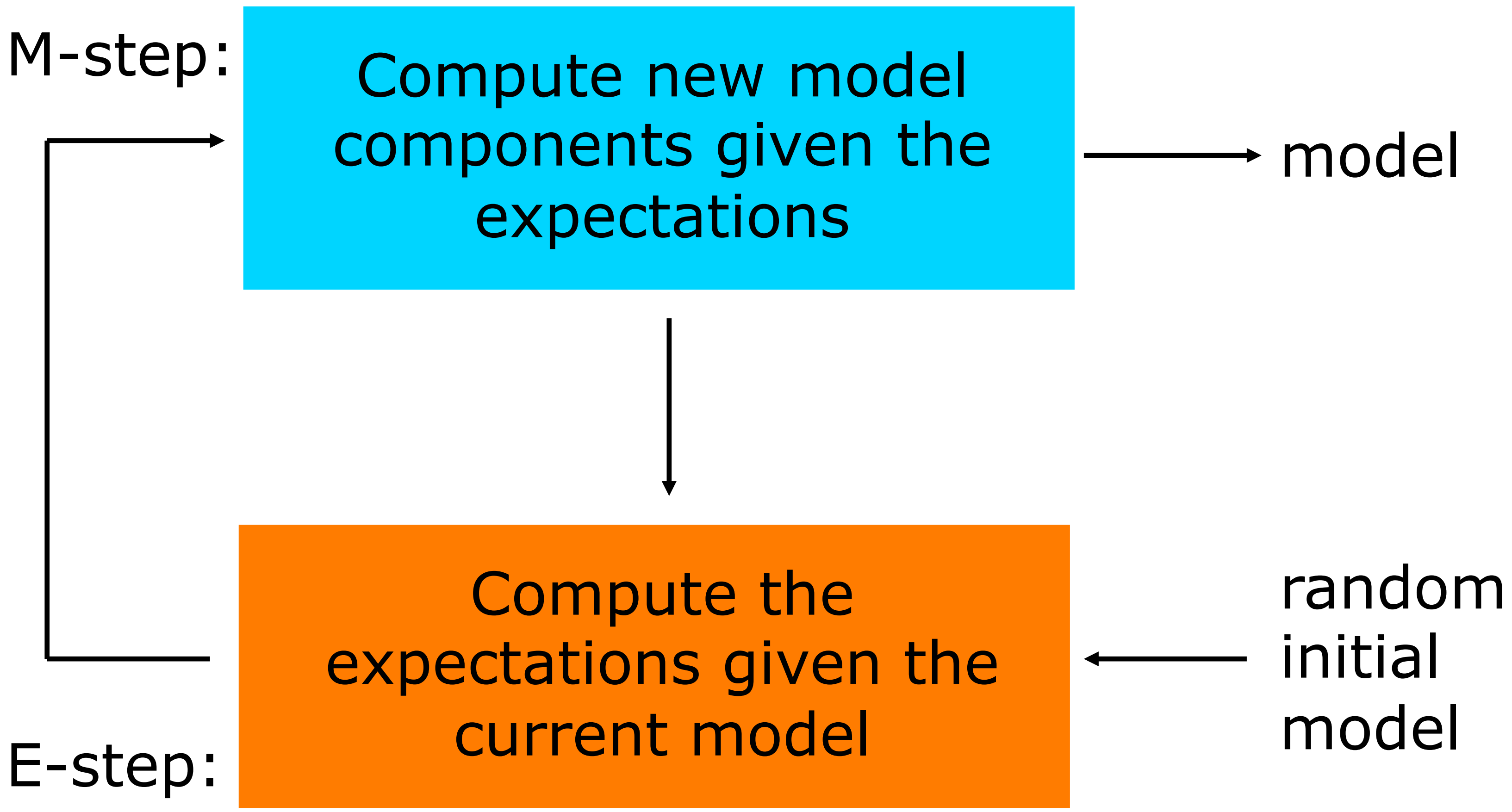
- Compute expected values for the hidden variables *c* given the current model $\theta^{[j]}$ and the observed data *d*

Maximization (M) step

- Maximize the expected likelihood

$$\theta^{[j+1]} = \operatorname*{argmax}_{\theta'} Q(\theta' \mid \theta^{[j]})$$

Iterating E and M Steps



Properties of EM

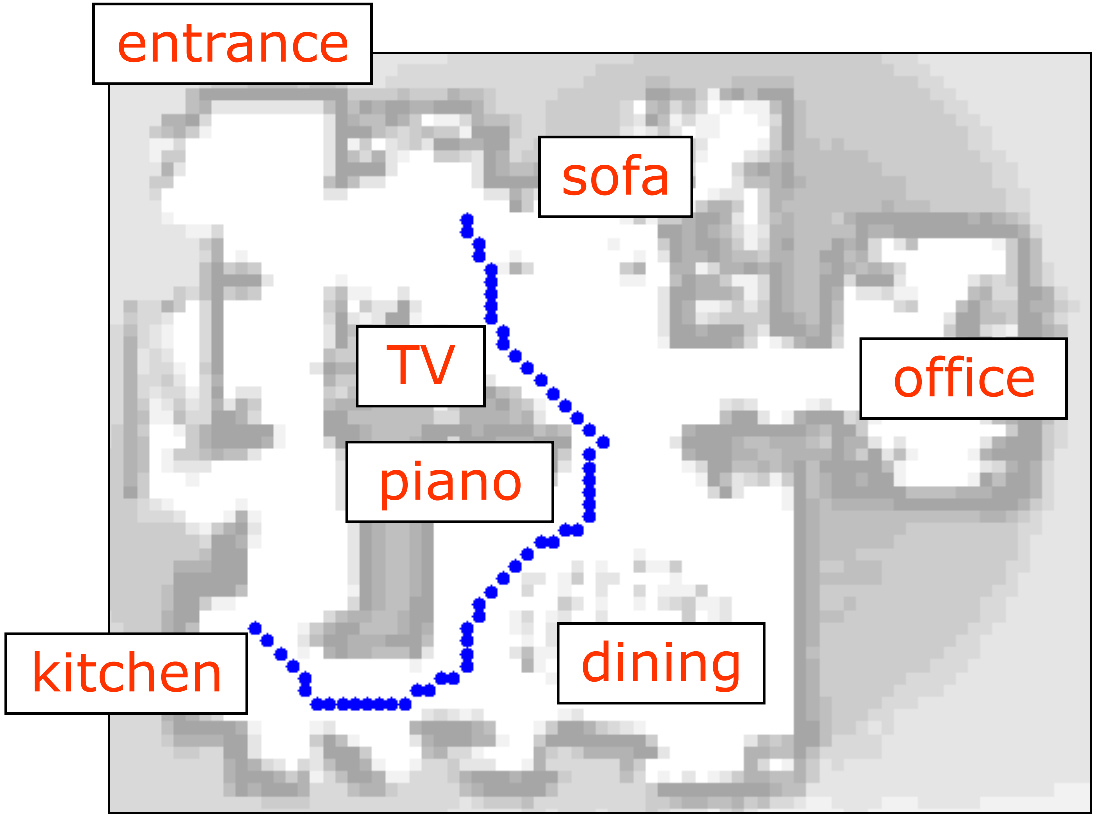
- Each iteration is guaranteed to increase the data log likelihood
- EM is guaranteed to converge to a local maximum of the likelihood function

Application: Trajectory Clustering

How to learn typical motion patterns of people from observations?

Application: Trajectory Clustering

- Input: Set of trajectories $d_1,\ldots,d_I$



$$d_i = \{x_i^1, x_i^2, \ldots, x_i^T\}$$

What we are looking for

- Clustering of similar trajectories into motion patterns $\theta_1, \ldots, \theta_M$
  (Note: From now on M = #clusters)
- Binary correspondence variables c_{im} indicating which trajectory d_i belongs to which motion pattern θ_m

Problem:

How can we estimate c_{im}?

Motion Patterns

- Use T Gaussians with fixed variance to represent each motion pattern (= model)
- If we knew the values of the c_{im}, the computation of the motion patterns would be easy
- But: These values are hidden
- Use EM to compute
  - Expected values for the c_{im}
  - The model θ (i.e., the set of motion patterns) which has the highest expected data likelihood

Likelihood of Trajectory d_i given Motion Pattern $\theta_m = \{\theta^1_m, \ldots, \theta^T_m\}$

$$\begin{aligned} P(d_i \mid \theta_m) &= \prod_{t=1}^{T} P(x_i^t \mid \theta_m^t) \\ &= \prod_{t=1}^{T} \exp\left(-\frac{1}{2\sigma^2} \|x_i^t - \mu_m^t\|^2\right) \end{aligned}$$

likelihood that the person is at location x_i^t after t observations given it is engaged in motion pattern θ_m with the means $\{\mu_m^1, \ldots, \mu_m^T\}$

Data Likelihood

- Joint likelihood of a single trajectory and its correspondence vector

$$P(d_i, c_i \mid \theta) \;=\; \prod_{t=1}^{T} \prod_{m=1}^{M} \exp\left(-\frac{1}{2\sigma^2} c_{im} \|x_i^t - \mu_m^t\|^2\right)$$

Note: c_{im}=1 for exactly one m

- Expected log likelihood

$$E_c[\ln P(d, c \mid \theta) \mid \theta, d] = E_c\left[\sum_{i=1}^{I} \ln P(d_i, c_i \mid \theta) \mid \theta, d\right]$$

$$= \; E_c\Big[\sum_{i=1}^{I} \ln \prod_{t=1}^{T} \prod_{m=1}^{M} \exp\left(-\frac{1}{2\sigma^2} c_{im} \|x_i^t - \mu_m^t\|^2\right) \mid \theta, d\Big]$$

$$= \; E_c\Big[-\frac{1}{2\sigma^2} \sum_{i=1}^{I} \sum_{t=1}^{T} \sum_{m=1}^{M} c_{im} \|x_i^t - \mu_m^t\|^2 \mid \theta, d\Big]$$

Expected Data Likelihood

$$E_c[\ln P(d, c \mid \theta) \mid \theta, d] = \ldots$$

$$= E_c\Big[-\frac{1}{2\sigma^2}\sum_{i=1}^{I}\sum_{t=1}^{T}\sum_{m=1}^{M} c_{im}\|x_i^t - \mu_m^t\|^2 \mid \theta, d\Big]$$

Expectation is a linear operator

$$= -\frac{1}{2\sigma^2}\sum_{i=1}^{I}\sum_{t=1}^{T}\sum_{m=1}^{M} E_c[c_{im}| \, \theta, d]\|x_i^t - \mu_m^t\|^2$$

$$= -\frac{1}{2\sigma^2}\sum_{i=1}^{I}\sum_{m=1}^{M} E[c_{im}| \, \theta, d]\sum_{t=1}^{T}\|x_i^t - \mu_m^t\|^2$$

Q-Function

$$Q(\theta' \mid \theta) \;=\; -\frac{1}{2\sigma^2}\sum_{i=1}^{I}\sum_{m=1}^{M} E[c_{im} \mid \theta, d]\sum_{t=1}^{T}\|x_i^t - \mu_m'^t\|^2$$

E-Step: Compute the Expectations Given the Current Model $\theta^{[j]}$

$$
\begin{aligned}
E[c_{im} \mid \theta^{[j]}, d] &= P(c_{im} \mid \theta^{[j]}, d) \\
&= P(c_{im} \mid \theta^{[j]}, d_i) \\
&\overset{\text{Bayes'}}{=} \eta P(d_i \mid c_{im}, \theta^{[j]}) P(c_{im} \mid \theta^{[j]}) \\
&\overset{\text{uniform prior}}{=} \eta' P(d_i \mid \theta_m^{[j]}) \\
&= \eta' \prod_{t=1}^{T} \exp\left(-\frac{1}{2\sigma^2} \|x_i^t - \mu_m^{t[j]}\|^2\right)
\end{aligned}
$$

Note: c_{im} can be 0 or 1

normalizer

likelihood that the i-th trajectory belongs to the m-th model component

M-Step: Maximize the Expected Likelihood

$$
\begin{aligned}
\theta^{[j+1]} & \\
&= \operatorname*{argmax}_{\theta'} Q(\theta' \mid \theta^{[j]}) \\
&= \operatorname*{argmax}_{\theta'} \Big\{ -\frac{1}{2\sigma^2} \sum_{i=1}^{I} \sum_{m=1}^{M} E[c_{im} \mid \theta^{[j]}, d] \sum_{t=1}^{T} \|x_i^t - \mu_m'^t\|^2 \Big\} \\
&= \operatorname*{argmin}_{\theta'} \Big\{ \sum_{i=1}^{I} \sum_{m=1}^{M} E[c_{im} \mid \theta^{[j]}, d] \sum_{t=1}^{T} \|x_i^t - \mu_m'^t\|^2 \Big\}.
\end{aligned}
$$

Compute partial derivative with respect to μ_m^t

M-Step: Maximize the Expected Likelihood

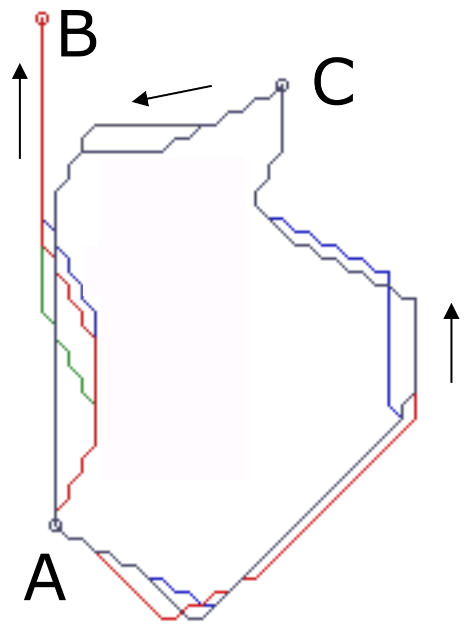
$$\sum_{i=1}^{I} E[c_{im} \mid \theta^{[j]}, d] \cdot (-2) \cdot (x_i^t - \mu_m^{t[j+1]}) \stackrel{!}{=} 0 \quad \Longleftrightarrow$$

$$\sum_{i=1}^{I} E[c_{im} \mid \theta^{[j]}, d]\, x_i^t \stackrel{!}{=} \sum_{i=1}^{I} E[c_{im} \mid \theta^{[j]}, d]\, \mu_m^{t[j+1]} \Longleftrightarrow$$

$$\mu_m^{t[j+1]} \stackrel{!}{=} \frac{\sum_{i=1}^{I} E[c_{im} \mid \theta^{[j]}, d]\, x_i^t}{\sum_{i=1}^{I} E[c_{im} \mid \theta^{[j]}, d]}$$

This is the mean update of soft k-means

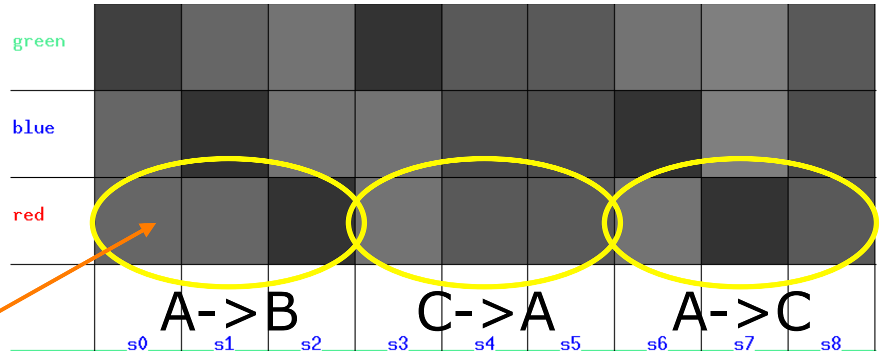
EM Application Example: 9 Trajectories of 3 Motion Patterns



EM: Example (step 1)

MP

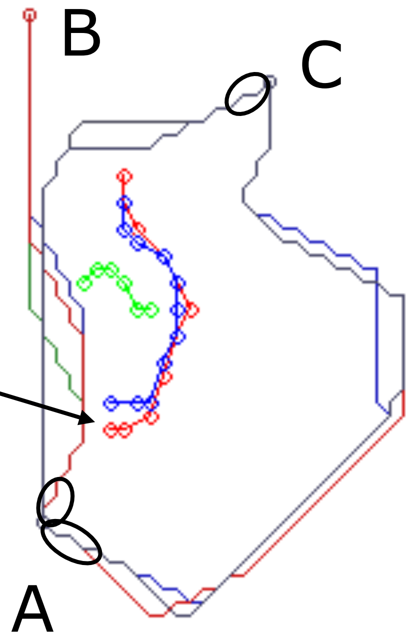
C_{im}:



trajectories

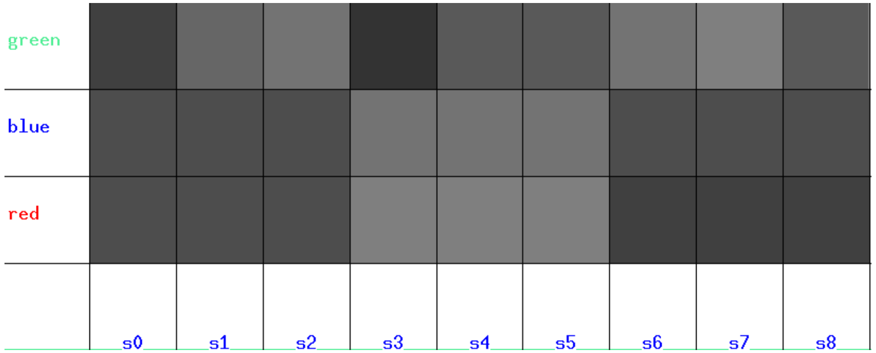
likelihood that s_0 belongs to the red motion pattern

θ^1:

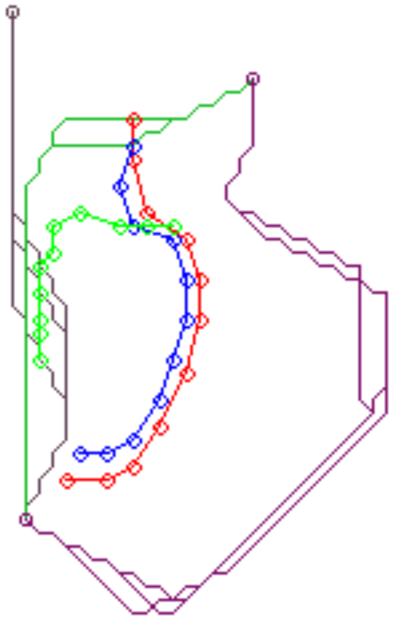


EM: Example (step 2)

C_{im}:

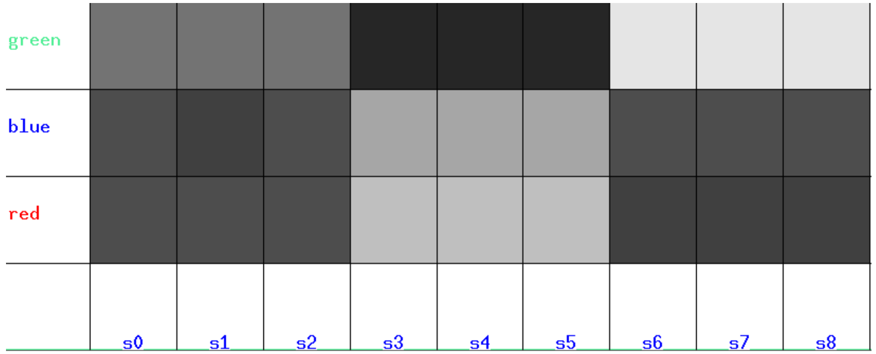


θ^2:

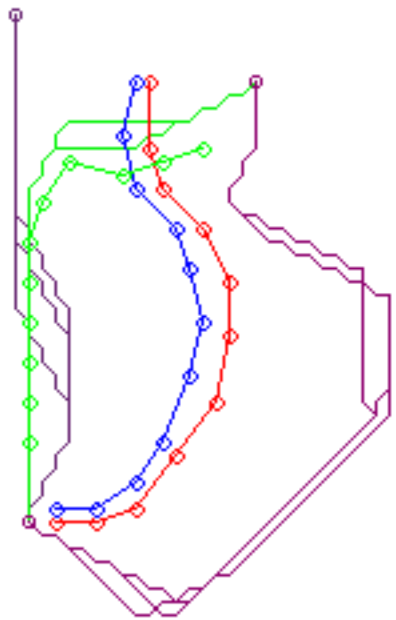


EM: Example (step 3)

C_{im}:

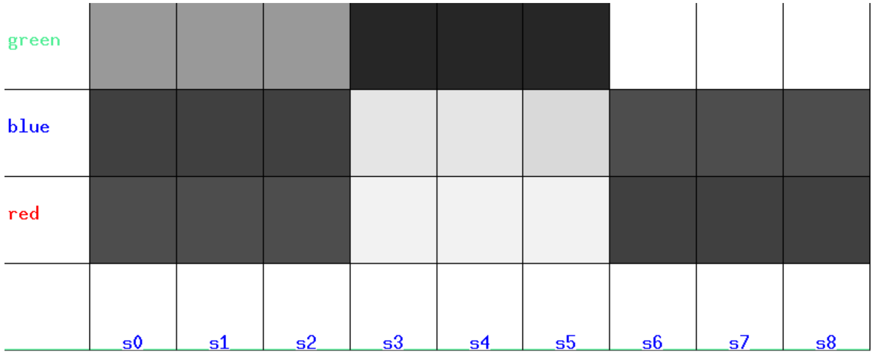


θ^3:

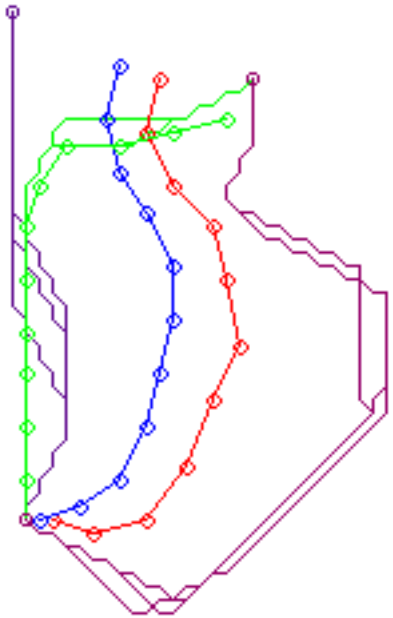


EM: Example (step 4)

C_{im}:

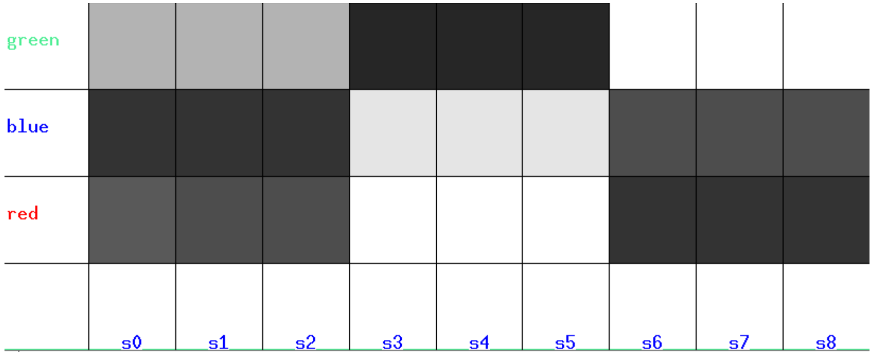


θ^4:

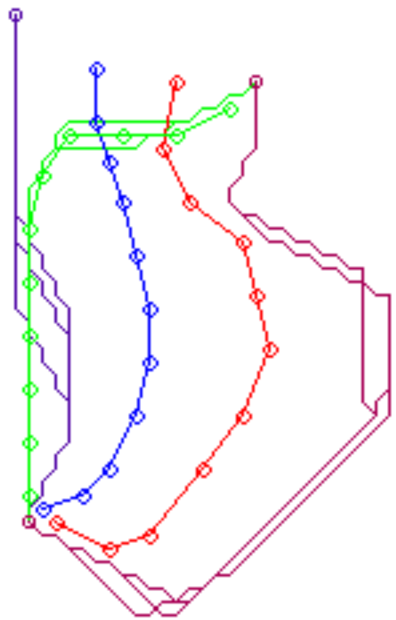


EM: Example (step 5)

C_{im}:

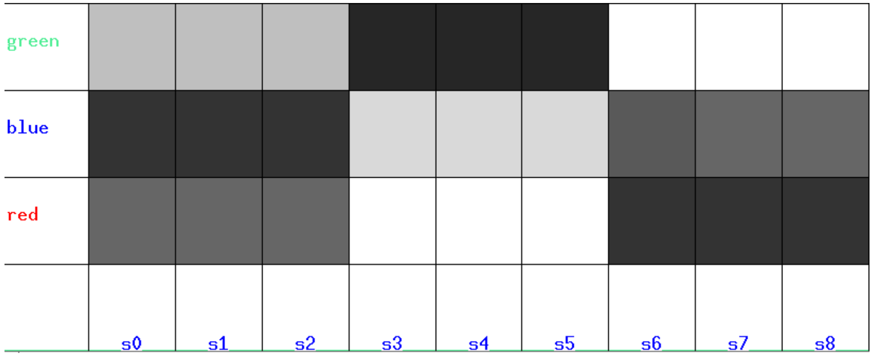


θ^5:

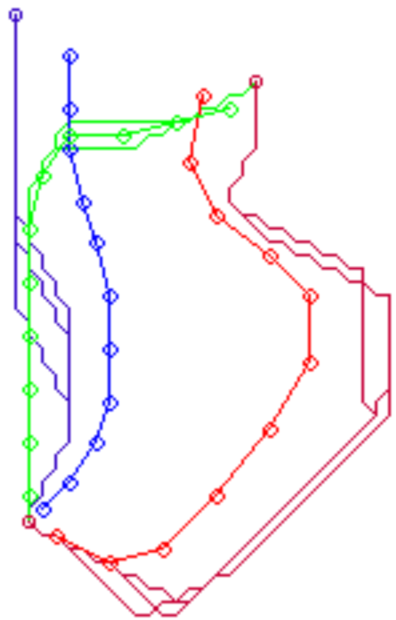


EM: Example (step 6)

C_{im}:

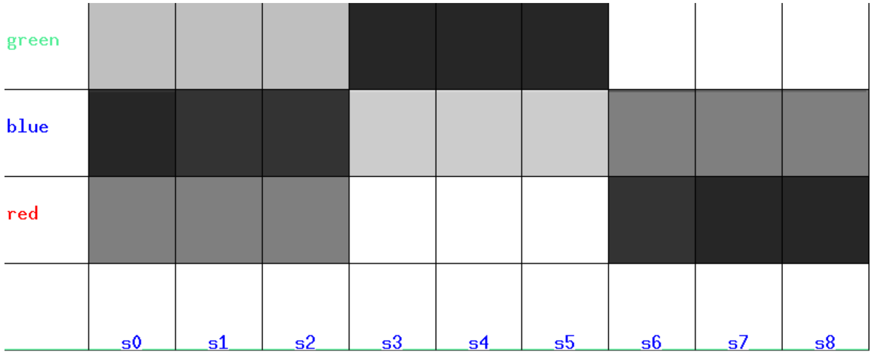


θ^6:

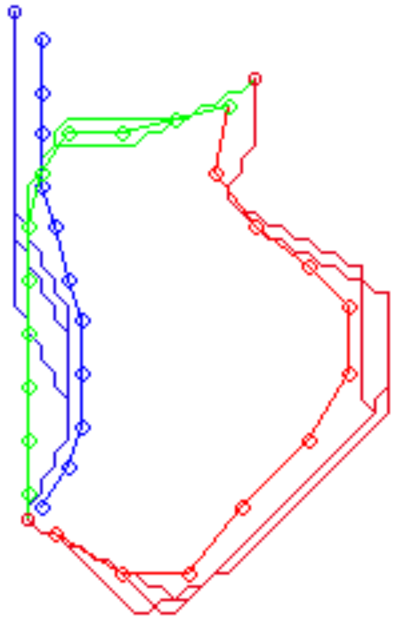


EM: Example (step 7)

C_{im}:

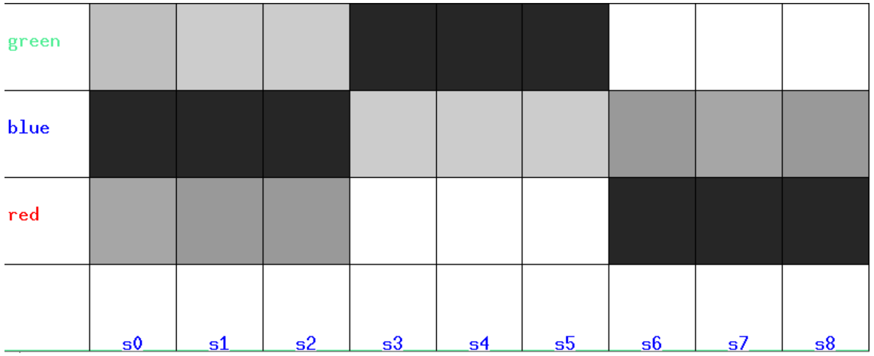


θ^7:

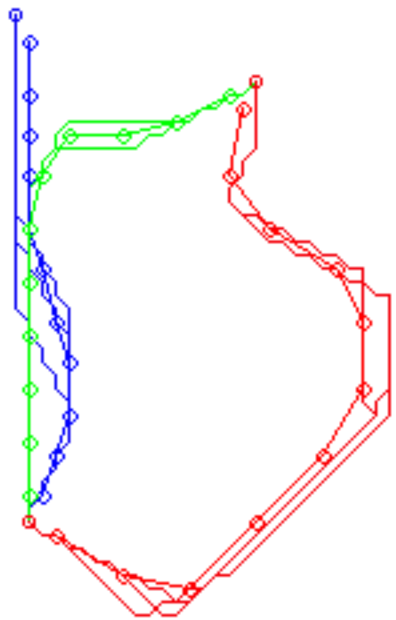


EM: Example (step 8)

C_{im}:

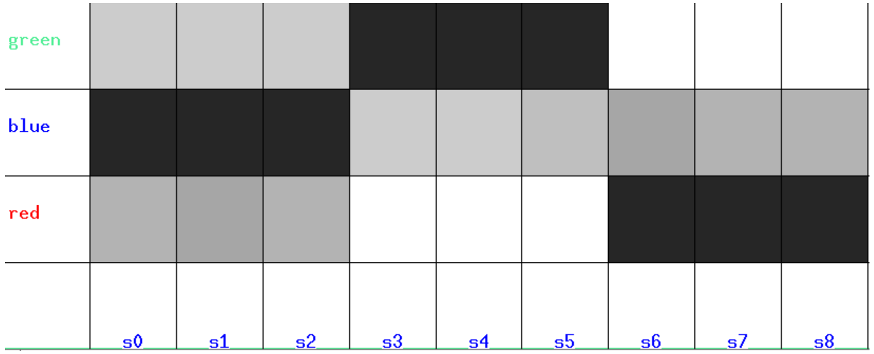


θ^8:

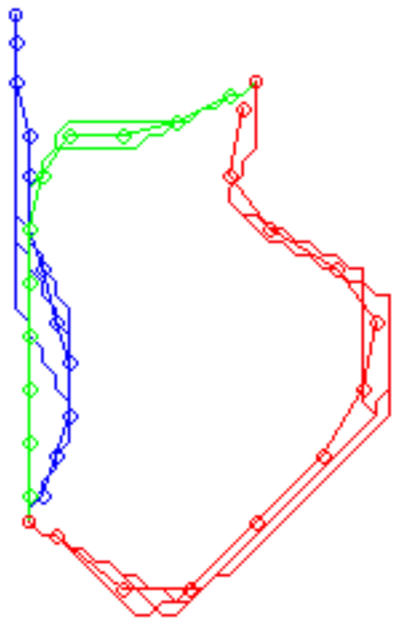


EM: Example (step 9)

C_{im}:



θ^9:



Estimating the Number of Model Components Greedily

After convergence of the EM check whether the model can be improved

- by introducing a new model component for the trajectory which has the lowest likelihood or
- by eliminating the model component which has the lowest utility.

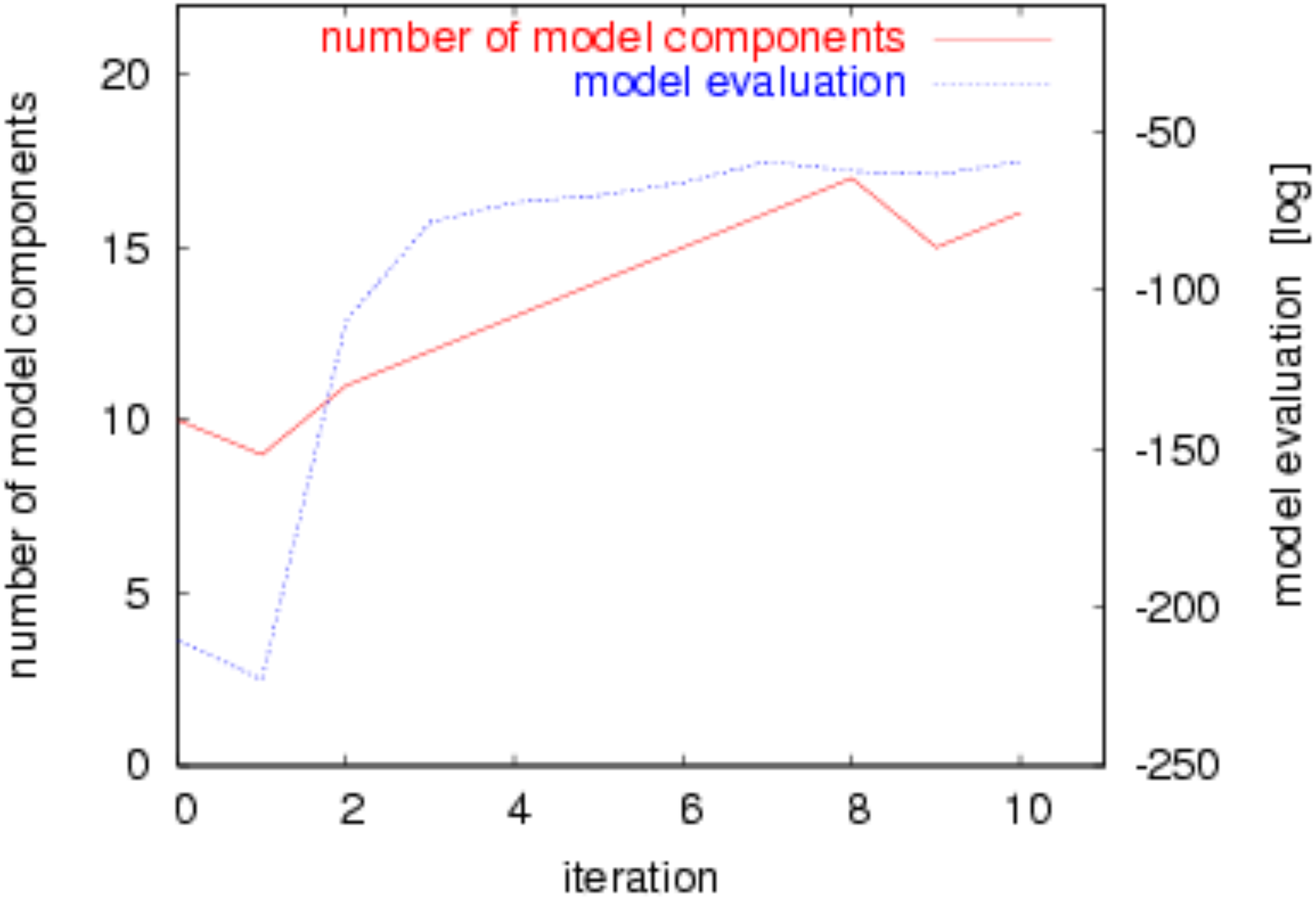
Select model θ which has the **highest evaluation**

$$E_c[\log P(d, c \mid \theta) \mid \theta, d] - \frac{M}{2} \log I$$

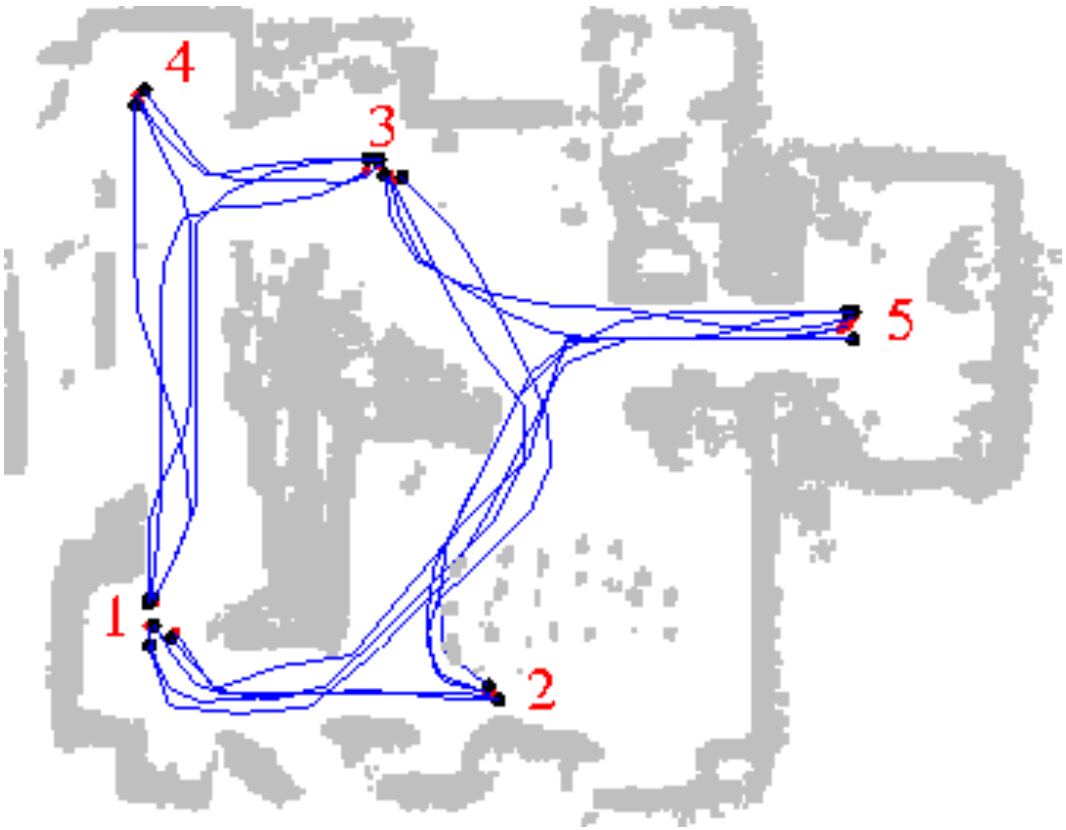
where M = #model components, I = #trajectories

Bayesian Information Criterion [Schwarz,`78]

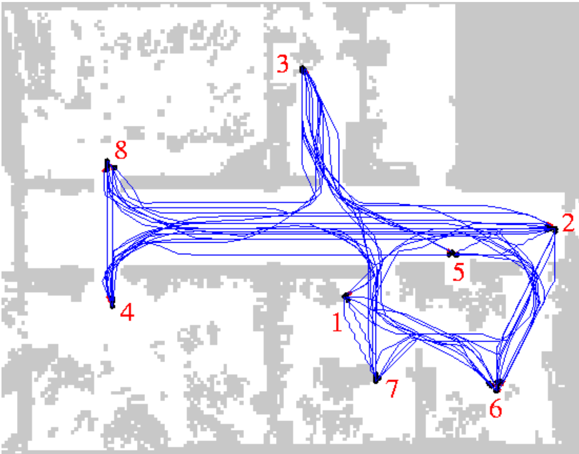
Application Example



Learned Motion Patterns

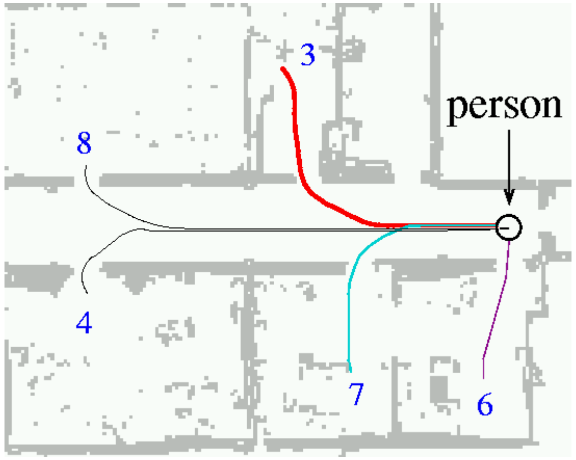


Prediction of Human Motion

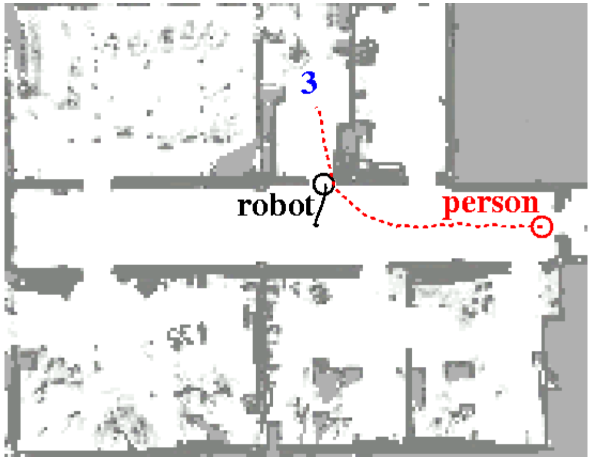


learned motion patterns

situation

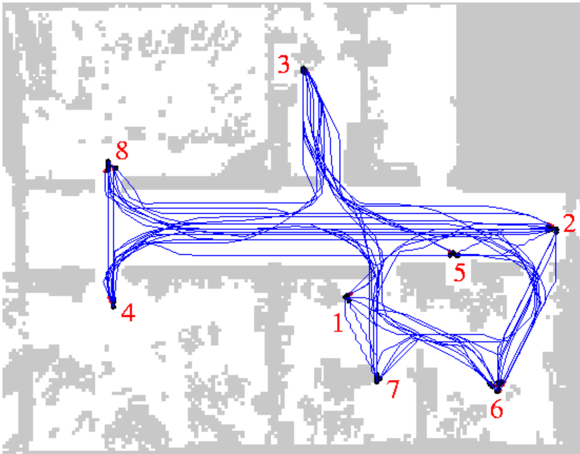


motion prediction

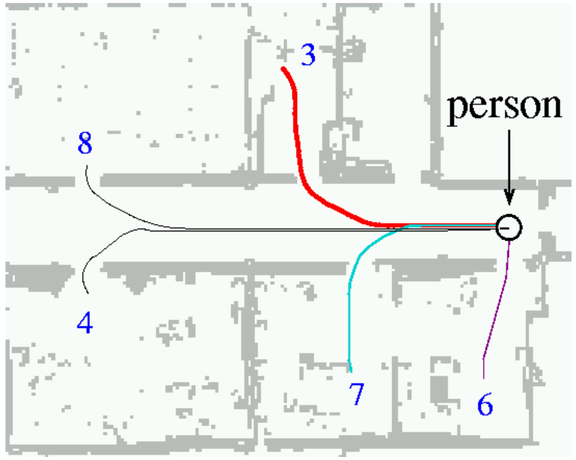


anticipation

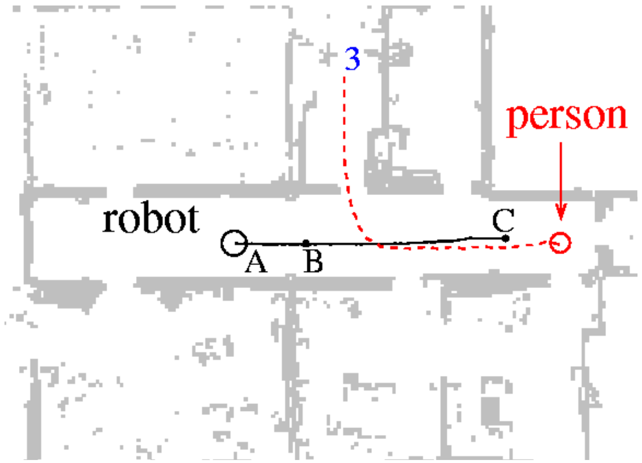
Prediction of Human Motion



learned motion patterns



motion prediction

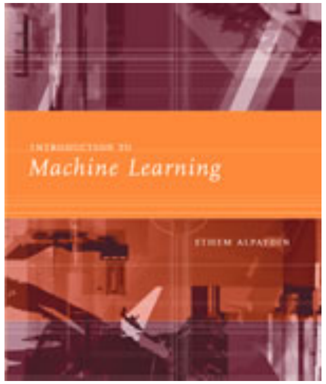


behavior adaption

Summary

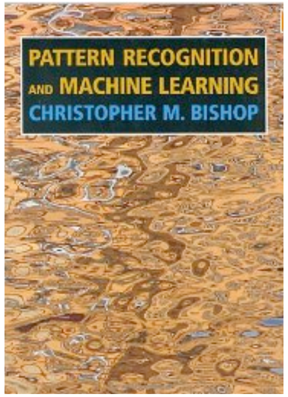
- K-means is the most popular clustering algorithm
- It is efficient and easy to implement
- Converges to a local optimum
- A variant of hard k-means exists allowing soft assignments
- Soft k-means corresponds to the EM algorithm which is a general optimization procedure

Further Reading



E. Alpaydin

Introduction to Machine Learning



C.M. Bishop

Pattern Recognition and Machine Learning

J. A. Bilmes

A Gentle Tutorial of the EM algorithm and its Applications to Parameter Estimation (Technical report)