WISSEN SCHAFFT ZUKUNFT

Carl-Cranz-Gesellschaft e.V.

Gesellschaft für technisch-wissenschaftliche Weiterbildung

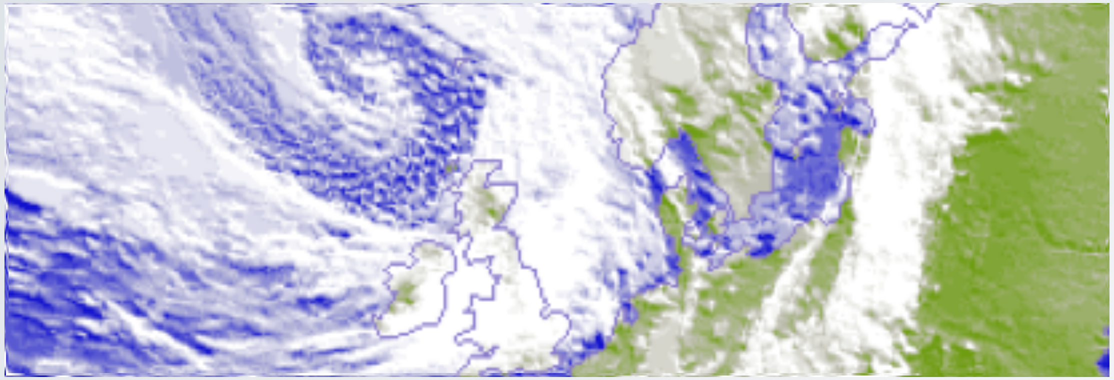
Information Fusion over Space and Time

Prof. Bernd Neumann, Ph.D.
Hamburger Informatik Technologie-Center (HITeC)

Space and Time: Examples (1)

WISSEN SCHAFFT ZUKUNFT

Global Weather



Airport Activities

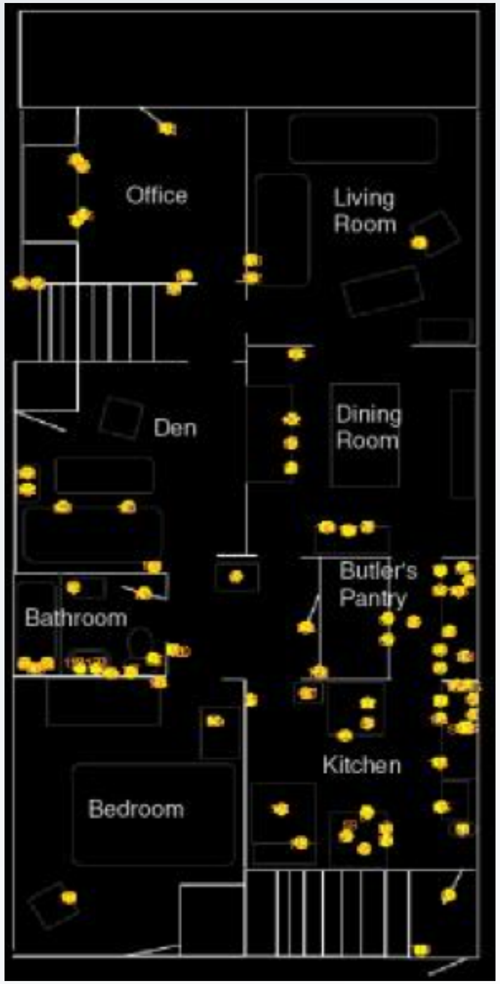
Space and Time: Examples (2)

WISSEN SCHAFFT ZUKUNFT

Recognising Smart Home activities

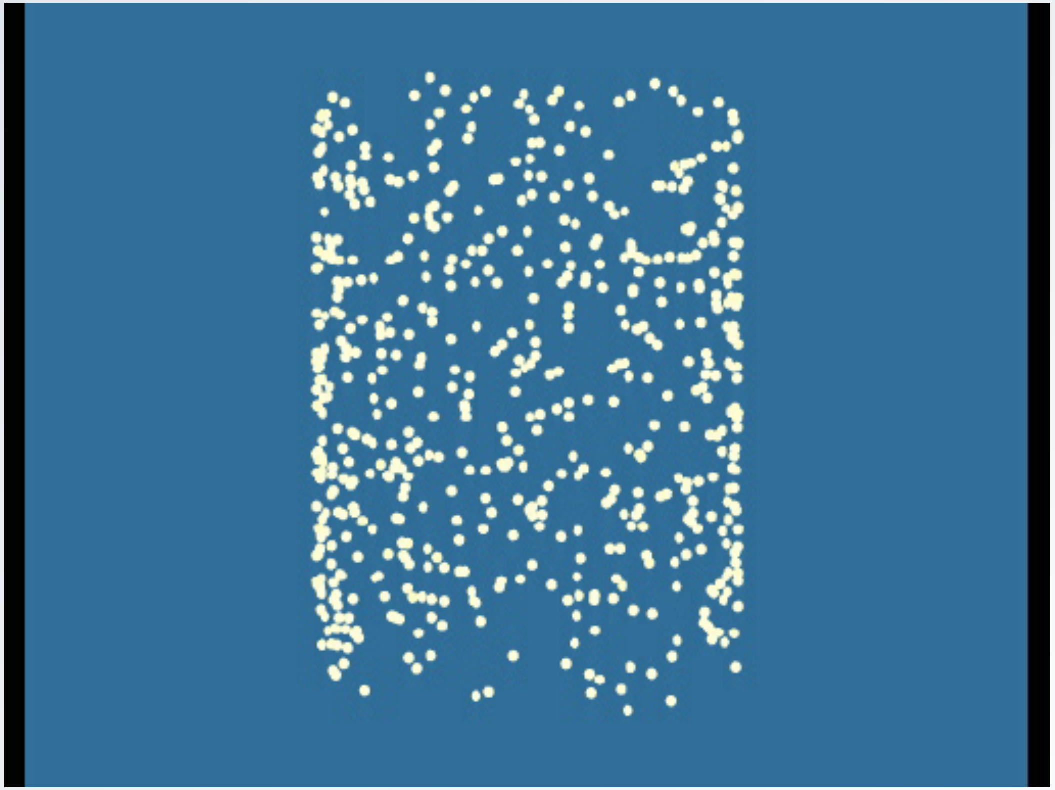
Activities

Preparing dinner
Preparing lunch
Preparing breakfast
Preparing a snack
Preparing a beverage
Taking medication
Washing dishes
Listening to music
Watching TV
Bathing
Dressing
Grooming
Toileting
Doing laundry
Cleaning
Going out



Space and Time: Examples (3)

"Shape from Motion"
in human vision



Key Dimensions Space and Time

WISSEN SCHAFFT ZUKUNFT

Space and time are key dimensions for describing

- **physical phenomena**
- **human activities**
- **spatiotemporal technical systems**

Descriptions use a large repertoire of terms:

- **quantitative measurements**
- **qualitative expressions**
- **abstractions**

➡ **feature vectors**

➡ **spatiotemporal aggregates**

Goals of this Lecture

WISSEN SCHAFFT ZUKUNFT

Survey of techniques and methods for the

- **representation of spatiotemporal phenomena**
- **learning and discovery**
- **recognition**

Agenda

WISSEN SCHAFFT ZUKUNFT

- **Space and time in feature space**
  Distance measures
  Learning
  Clustering
  Probabilistic models
- **Spatiotemporal aggregates**
  Logic-based representations
  Bayesian Compositional Hierarchies

Part 1

WISSEN SCHAFFT ZUKUNFT

Space and Time in Feature Space

Feature Space

WISSEN SCHAFFT ZUKUNFT

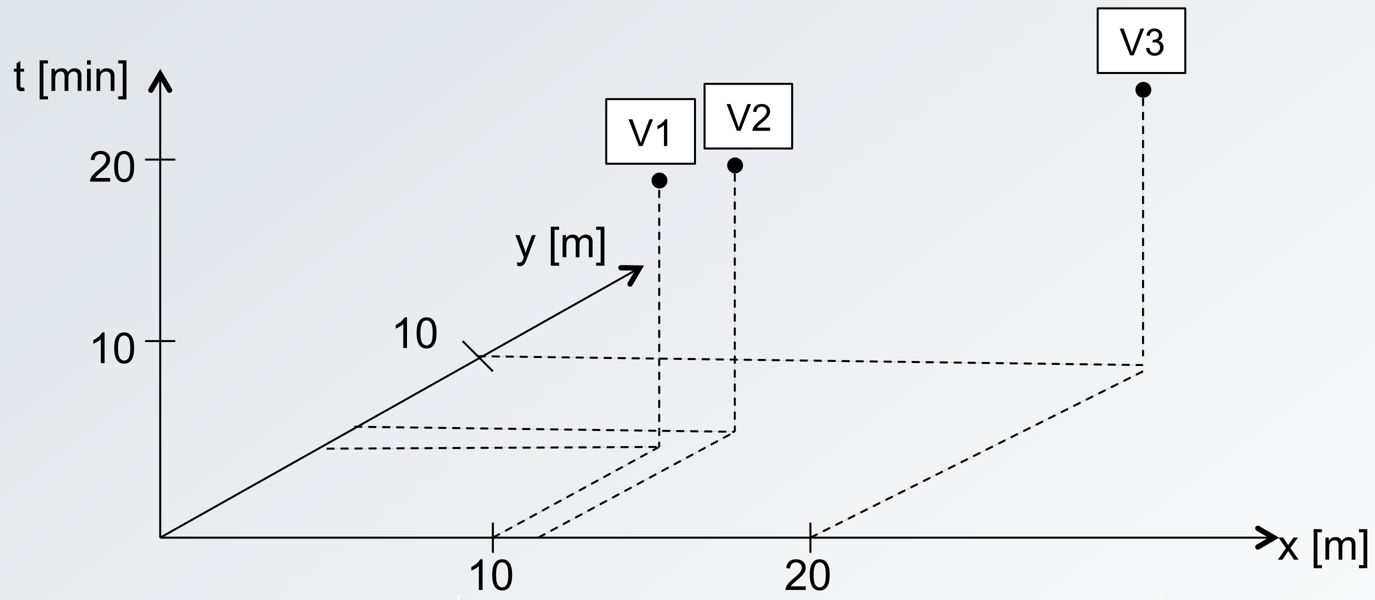
Objects with numerical features can be represented as points in N-dimensional feature space.

Example: Vehicle arrival times and locations during aircraft servicing

V1	
Arrival	13
Location	[10, 5]

V2	
Arrival	14
Location	[11, 6]

V3	
Arrival	14
Location	[20, 10]



Learning Feature-based Descriptions

WISSEN SCHAFFT ZUKUNFT

If objects can be described as feature vectors, a large repertoire of learning procedures exists to establish object classes.

Unsupervised learning:

Discover "useful" object classes in large data sets

Supervised learning:

Determine discriminating functions for examples with known class membership

Discovering Clusters

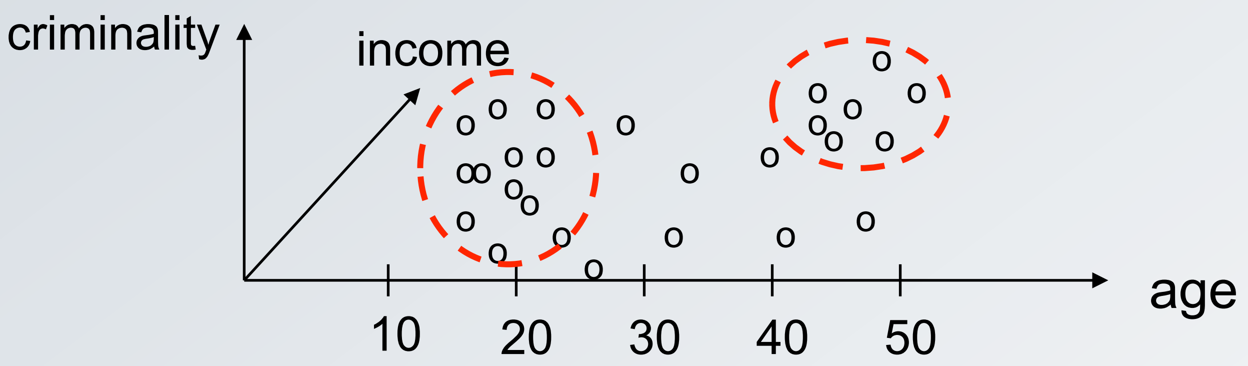
"Understanding our world requires conceptualising the similarities and differences between the entities that compose it."

Tyron & Bailey 1970

"Clusters" of data objects are hypothetical classes based on similarities and distances. Data objects should be as similar as possible within clusters and as distinct as possible between clusters.

Defining similarity and distance is the crucial part of clustering!

Role of Distance in Clustering



Cluster 1: age 15 - 25, low income, high criminality ("youth criminality")
Cluster 2: age 45 - 55, high income, high criminality ("white-collar criminality")

- **Data objects are viewed as points in multi-dimensional feature space.**
- **Similarity of data is judged by one of several distance measures.**

Challenges of Clustering

- **Determine useful features of data objects**
  E.g. is daytime a useful feature for activity clustering?
- **Collect representative data**
  Clusters from statistically biased samples may be misleading
- **Determine similarity measure**
  What is the "distance" between different combinations of timepoints and locations?
- **Determine granularity or cluster number**
- **Determine clusters**

Distance Measures

A valid distance measure between two data objects x and y must satisfy

- **d(x, y) ≥ 0**
- **d(x, y) = d(y, x)**
- **x = y => d(x, y) = 0**

A distance measure is a metric if

- **d(x, z) ≤ d(x, y) + d(y, z)**
- **d(x, y) = 0 => x = y**

Distance measures depend on the data types of the features which must be compared:

- Continuous-valued
- Discrete-valued
- Binary-valued
- Symbol-valued

Distance for Numerical Features (1)

<u>x</u> and <u>y</u> are <u>continuous-valued</u> N-dimensional data objects

<u>Example</u>: Space and time

Weighted distance: $d(\underline{x}, \underline{y}) = w_1 |x_1-y_1|^g + w_2 |x_2-y_2|^g + \dots w_N |x_N-y_N|^g$

For $w_1 \dots w_N = 1$ we have

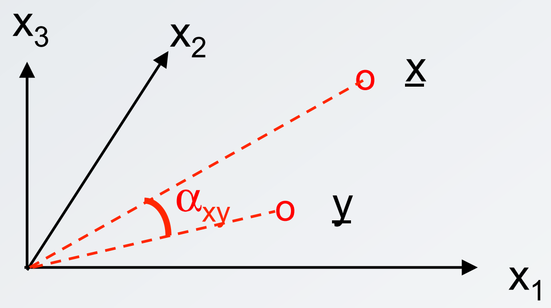
- $g = 1$: Manhattan metric
- $g = 2$: Euclidean metric
- $g \Rightarrow \infty$: Chebychev metric $d(\underline{x}, \underline{y}) = \max(|x_i-y_i|)$ (emphasizes the dimension with largest distance)

Distance for Numerical Features (2)

$\underline{x}$ and $\underline{y}$ are continuous-valued N-dimensional data objects, viewed as vectors

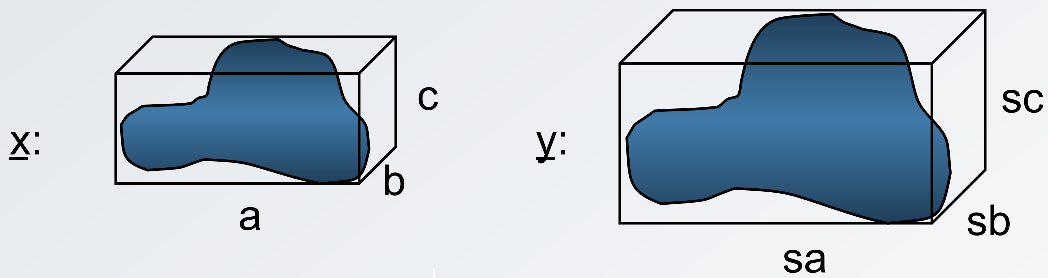
If the angle between $\underline{x}$ and $\underline{y}$ is significant for the distance rather than the magnitude of each vector, the cosine distance is appropriate:

$$d(\underline{x},\underline{y}) = 1 - \frac{\underline{x}^T \underline{y}}{\|\underline{x}\|\|\underline{y}\|} = 1 - \cos\alpha_{xy}$$



Example:

Scale invariant description of shape by bounding box [x_1, x_2, x_3]



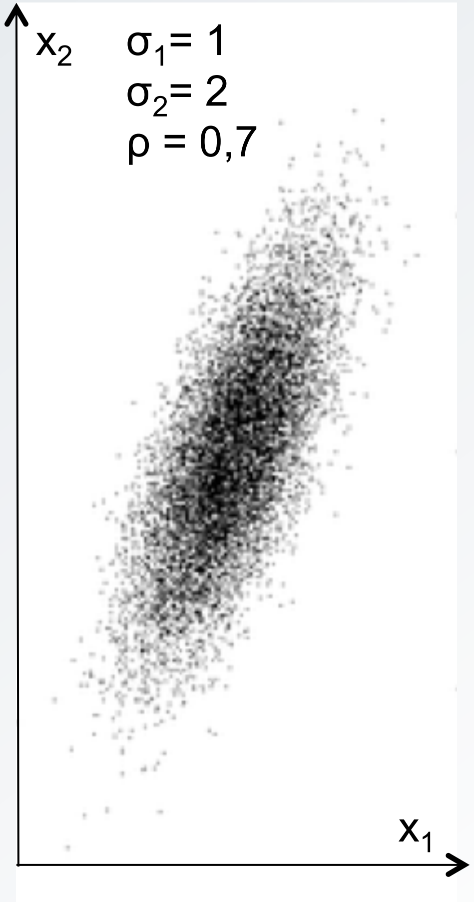
⇒ $d(\underline{x}, \underline{y}) = 0$

Probabilistic Distance Measure

WISSEN SCHAFFT ZUKUNFT

Given that a probability distribution of <u>continuous-valued</u> N-dimensional data points is available.

Example: Multivariate Gaussian distribution



Mahalanobis distance:

$$d(\underline{x},\underline{y}) = \sqrt{(\underline{x}-\underline{y})^T \Sigma^{-1} (\underline{x}-\underline{y})}$$

Mahalanobis distance scales down

- **high-variance dimensions**
- **influence of correlations**

Distance for Discrete-valued Features

WISSEN SCHAFFT ZUKUNFT

$\underline{x}$ and $\underline{y}$ are <u>discrete-valued</u> N-dimensional data objects

<u>Example</u>: number-of-parts {0, 1, 2, ... }

All distance measures for continuous-valued features can be applied in principle .

Sometimes distances for large values are less important than for small values:

$$d(x_k, y_k) = \frac{|x_k - y_k|}{x_k + y_k} \qquad \text{for } x_k + y_k > 0$$

Distance for Binary Features

WISSEN SCHAFFT ZUKUNFT

x and y are binary-valued N-dimensional data objects.

Distance is determined by counting equal and unequal 0 and 1 features.

For symmetric binary features, 0 and 1 are equally valued:

$$d(\underline{x}, \underline{y}) = \frac{|\text{unequal features}|}{N}$$

Hamming distance

For asymmetric binary features, 0 is often considered less valued, and features are ignored where both objects are 0 :

q = |equal 0,0 features| r = |equal 1,1 features|

s = |unequal 0,1 features| t = |unequal 1,0 features|

$$d(\underline{x}, \underline{y}) = \frac{s + t}{r + s + t}$$

Distance for Nominal Values

WISSEN SCHAFFT ZUKUNFT

$\underline{x}$ and $\underline{y}$ are symbol-valued N-dimensional data objects

Examples: colour $\in$ {red, green, blue, black, white}

sex $\in$ {male, female}

1. Matching distance $d(\underline{x},\underline{y}) = \dfrac{N-m}{N}$ **m = number of matches**

2. Transformation into binary features

A k-valued nominal feature is transformed into k binary features.

Example: colour $\in$ {red, green, blue, black, white}

→ red $\in$ {T, F}, green $\in$ {T, F}, blue $\in$ {T, F}, ...

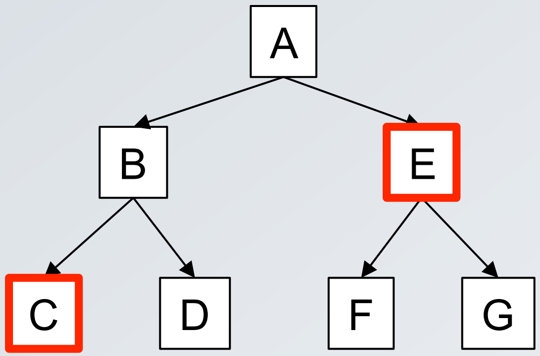
After transformation, distance measures for binary features can be applied.

Distance in Taxonomies (1)

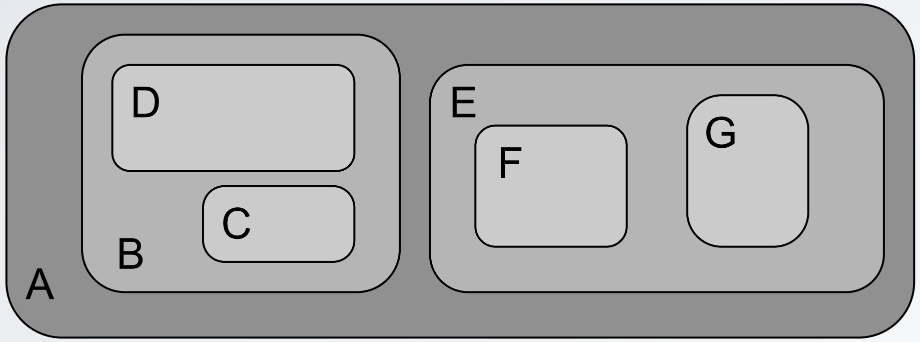
WISSEN SCHAFFT ZUKUNFT

Distance computations can be refined by the taxonomical structure of nominal distance values.

taxonomical structure



taxonomy of zones

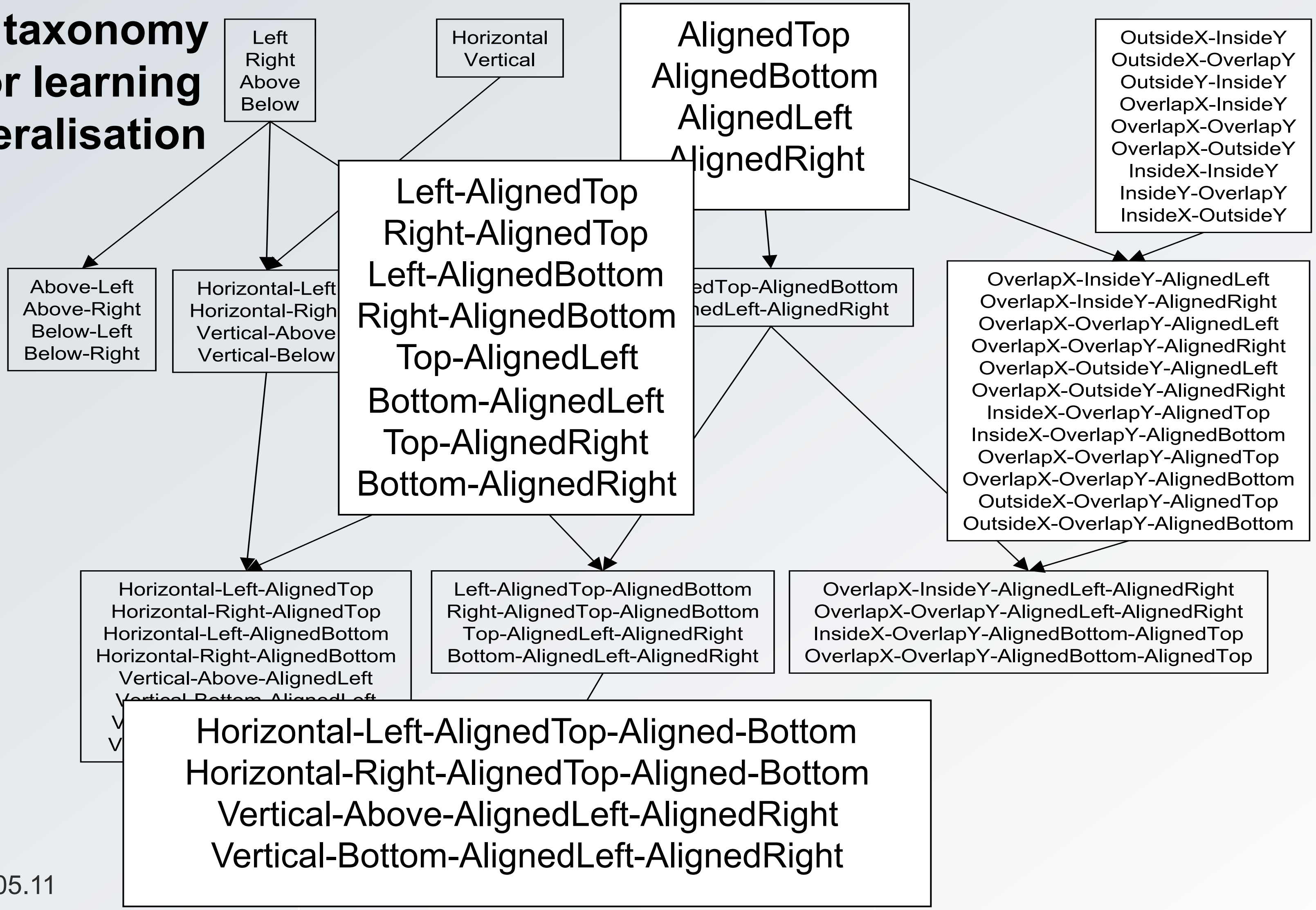


What is the distance between C and E?

The length of the path between two nominal values is an obvious distance measure, but it does not always reflect intuitions about a domain.

Distance in Taxonomies (2)

Spatial taxonomy used for learning by generalisation



Distance for Mixed Types

WISSEN SCHAFFT ZUKUNFT

Distances between features of different types can be combined by first normalising the typed distance measures to the range [0 .. 1] and then using a distance measure for numerical values.

Normalisation of continuous-valued feature distance:

$$d(x_k, y_k) = \left(\frac{|x_k - y_k|}{\max x_k - \min x_k} \right)^g$$

The problem of combining "apples with pears" cannot be solved satisfactorily, not only for different data types but generally for features belonging to different semantic categories.

K-Means Clustering Algorithm (1)

WISSEN SCHAFFT ZUKUNFT

- Most popular clustering algorithm
- Searches for local minimum of sum of Euclidean sample distances to cluster centers
- Guaranteed convergence in a finite number of steps
- Requires initialisation of fixed number of clusters K
- May converge to local minimum

K-Means Clustering Algorithm (2)

WISSEN SCHAFFT ZUKUNFT

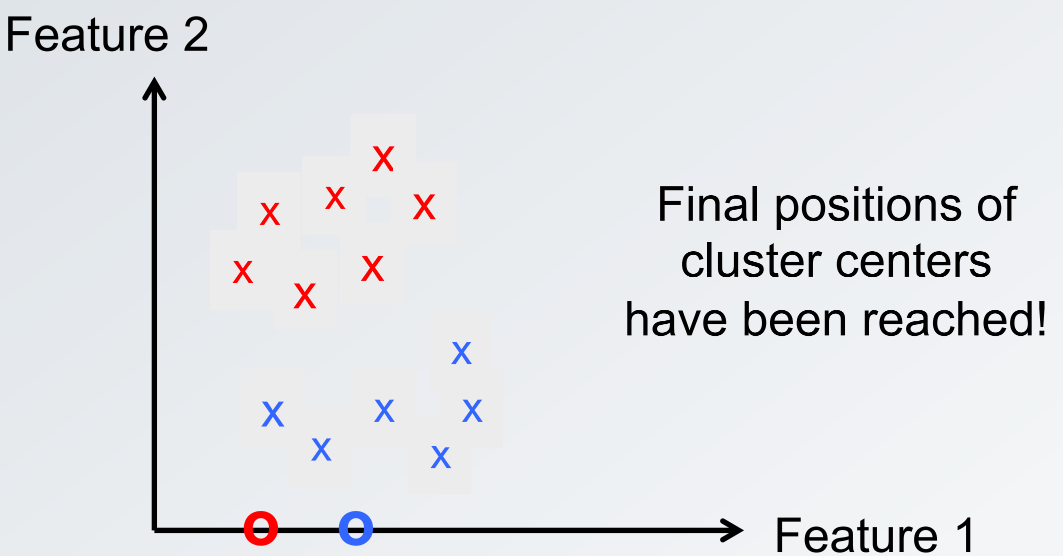
Represent exemplars as points in N-dimensional feature space

A **Choose arbitrary initial cluster centers**

B **Determine cluster assignments using distance measure**

C **Move cluster centers to center-of-gravity of assigned exemplars**

D **Repeat B and C until no further changes occur**



Expectation Maximization (EM)

K-means clustering is a special case of the Expectation-Maximization (EM) algorithm.

Basic idea of EM-algorithm:

Initially one has

- **data with missing values**
  e.g. unassigned cluster memberships
- **distribution models**
  e.g. rule to assign to nearest-distance cluster center

Iterate the two steps:

E-Step: Compute expected distribution parameters based on data (initially by random choice)

M-Step: Maximise likelihood of missing values based on distribution parameters

Example for EM (1)

Problem: Fit 3 straight lines to data points, not knowing which point belongs to which line.

Algorithm:

A Select 3 random lines initially

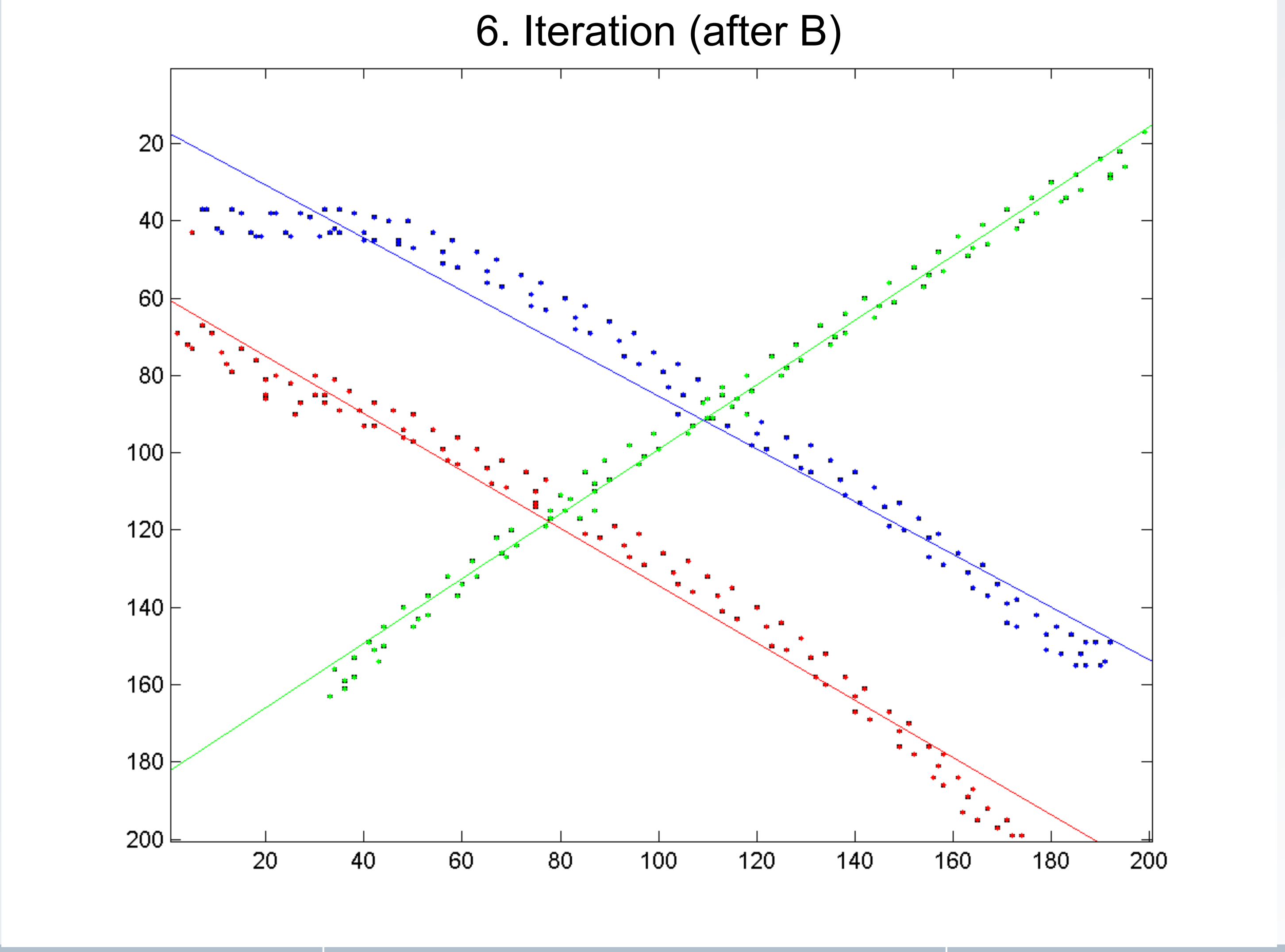
B Assign data points to each line by minimum distance criterion

C Determine best-fitting straight line for assigned data points

D Repeat B and C until no further changes occur

(Example by Anna Ergorova, FU Berlin)

Example for Expectation Maximization (1)



Agglomerative Clustering

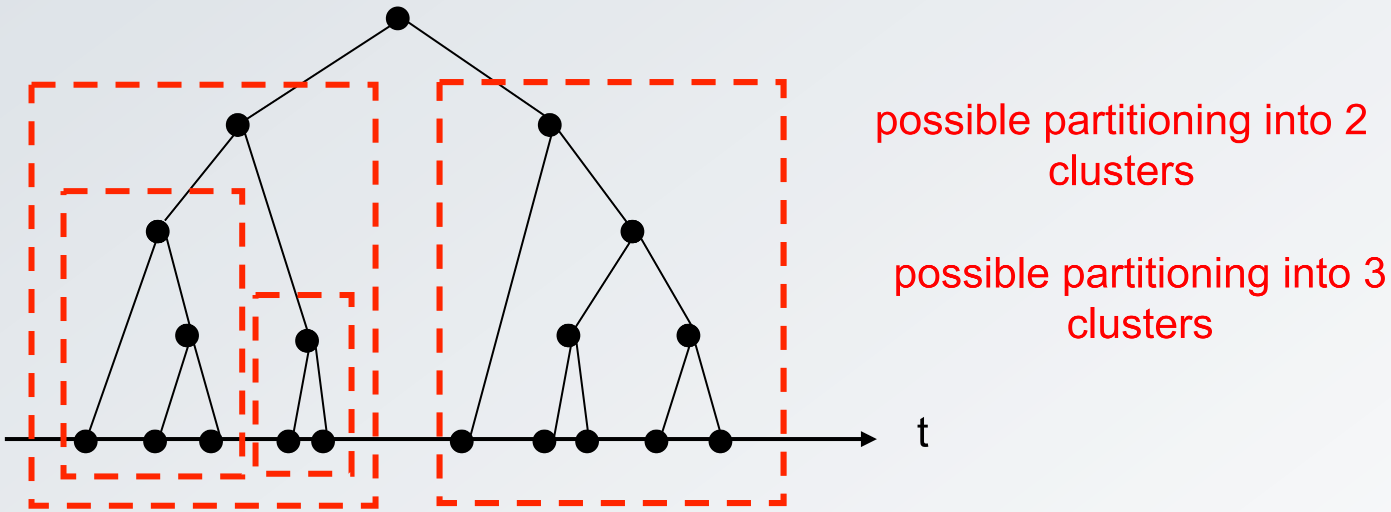
WISSEN SCHAFFT ZUKUNFT

- **Data are incrementally combined to clusters**
- **A cluster tree is generated**
- **Final partitioning into clusters is left to the user**

A Initially, all data objects are distinct clusters
B Merge cluster pair with nearest distance
C Enter new cluster into cluster tree
D Repeat steps B and C until all clusters are merged

Example for Agglomerative Clustering

Example:
Clustering of 1-dimensional data objects



Distance Measures for Clusters

WISSEN SCHAFFT ZUKUNFT

Intra-cluster distance can be measured by

- the average distance
- the maximum distance

between cluster members and cluster center.

(see distance measures introduced earlier)

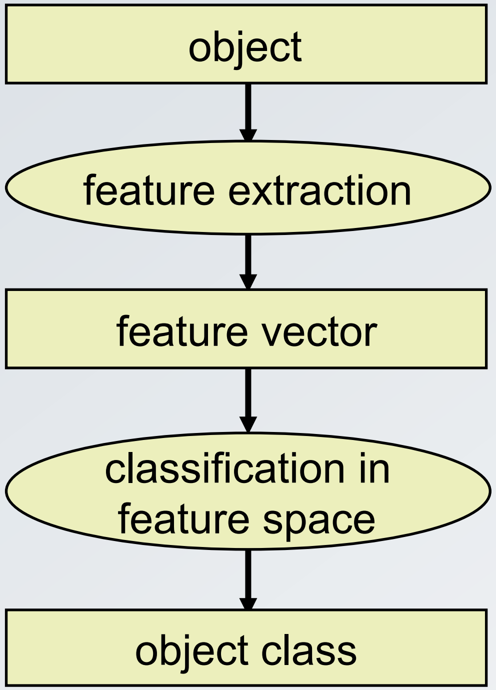
Inter-cluster distance can be measured by

- the smallest distance between elements of two distinct clusters ("single-link clustering")
- the largest distance between elements of two distinct clusters ("complete-link clustering")
- the average distance between elements of two distinct clusters ("average-link clustering")

Supervised Learning of Feature Space Boundaries

Given a large set of examples with known class membership, how can one determine discriminating functions for unknown examples?

➡ **Pattern Recognition paradigm**



K classes $\omega_1 \dots \omega_K$

N dimension of feature space

$\underline{x}^T = [x_1\ x_2 \dots\ x_N]$ feature vector

$\underline{y}^T = [y_1\ y_2 \dots\ y_N]$ prototype

(feature vector with known class membership)

$\underline{y}_i^{(k)}$ i-th prototyp of class k

$g_k(\underline{x})$ discriminant function for class k

<u>Problem</u>:

Determine $g_k(\underline{x})$ such that

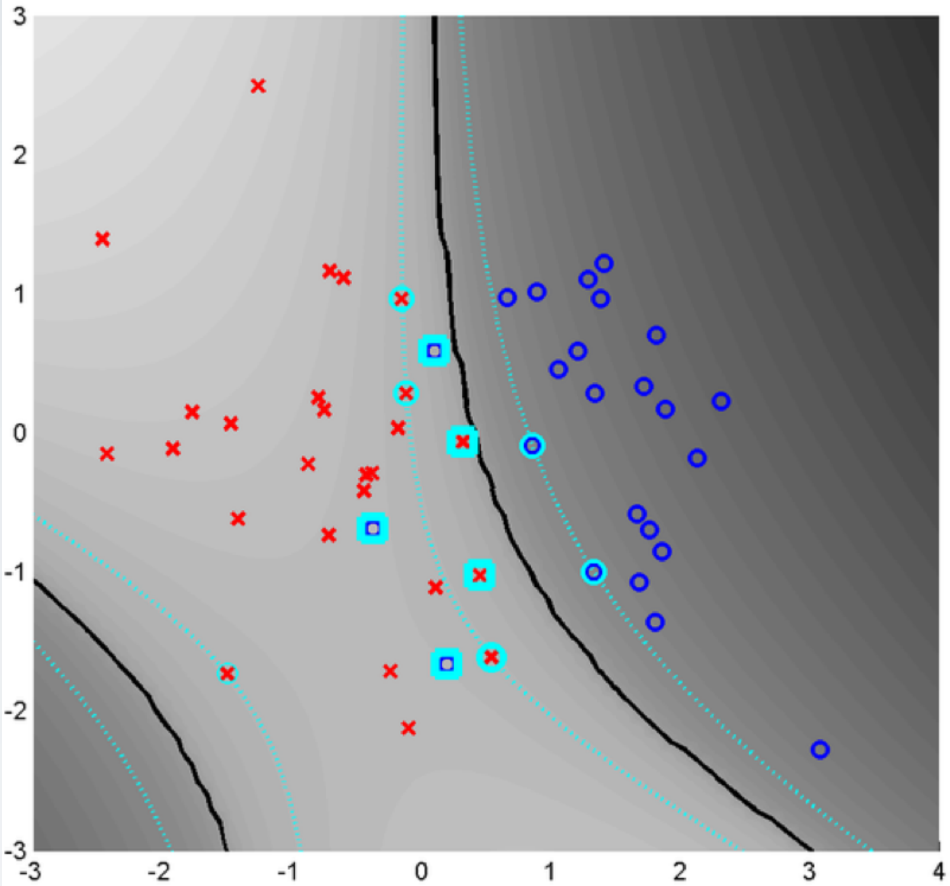
$$g_k(\underline{x}) > g_j(\underline{x}) \quad \forall \underline{x} \in \omega_k \quad \forall k \neq j$$

Support Vector Machines (SVMs)

WISSEN SCHAFFT ZUKUNFT

State-of-the art method for learning discriminating functions

- **problem reduction to critical ("supporting") prototypes**
- **non-linear transformation with kernel function to achieve separability**
- **separating hyperplanes in transformed feature space**
- **best possible margin**



Probabilistic Models

WISSEN SCHAFFT ZUKUNFT

The joint probability distribution $P(\underline{x}\ \omega_k)$ of feature vectors $\underline{x}$ and classes ω_k provides a powerful basis for classification.

Bayes decision rule:

Classify given $\underline{x}$ as class ω' such that ω' minimizes the probability of error $P(\omega \neq \omega' | \underline{x})$

➡ Choose ω' which maximizes the posterior probability $P(\omega | \underline{x})$

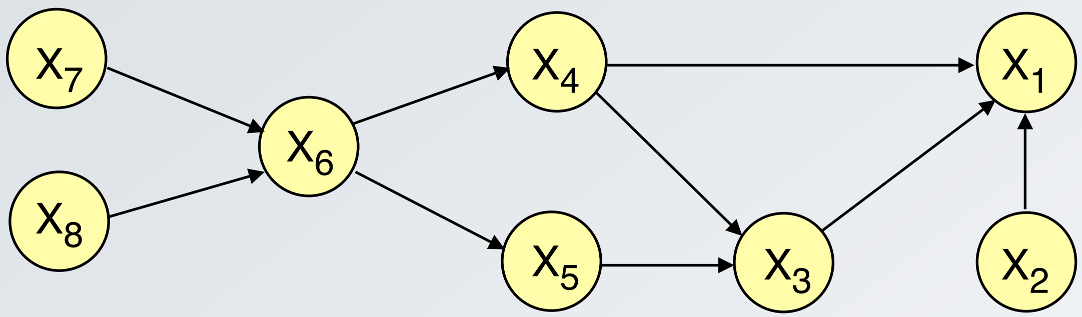
Decisions with partial information and estimation of missing values

Example: $x_1=a_1 \ldots x_n=a_n$):

$$P(\omega_k | x_1=a_1 \ldots x_n=a_n) = \frac{\sum_{x_{n+1}\ldots x_N} P(\omega_k x_1=a_1 \ldots x_n=a_n\ x_{n+1} \ldots x_N)}{\sum_{\omega_k x_{n+1}\ldots x_N} P(\omega_k x_1=a_1 \ldots x_n=a_n\ x_{n+1} \ldots x_N)}$$

Bayesian Networks

Conditional dependencies (causality relations) of random variables allow compact representation as "Graphical Model":



➡ $P(X_1 \ldots X_N) = \prod P(X_i \mid Parents(X_i))$

Tabular representations of Bayesian Networks with large variable domains (e.g. for locations and time) may still be intractable because of the table sizes.

Parametric probability distributions are preferable (e.g. Multivariant Gaussians) .

Part 2

WISSEN SCHAFFT ZUKUNFT

Spatiotemporal Aggregates

Requirements for Aggregate Representations

Aggregate descriptions must be able to specify richly structured objects with diverse properties and with parts which may be also aggregates.

Conceptual description of a spatiotemporal aggregate must specify

- **name**
- **taxonomical relations**
- **aggregate properties**
- **parts**
- **spatiotemporal relations between parts**

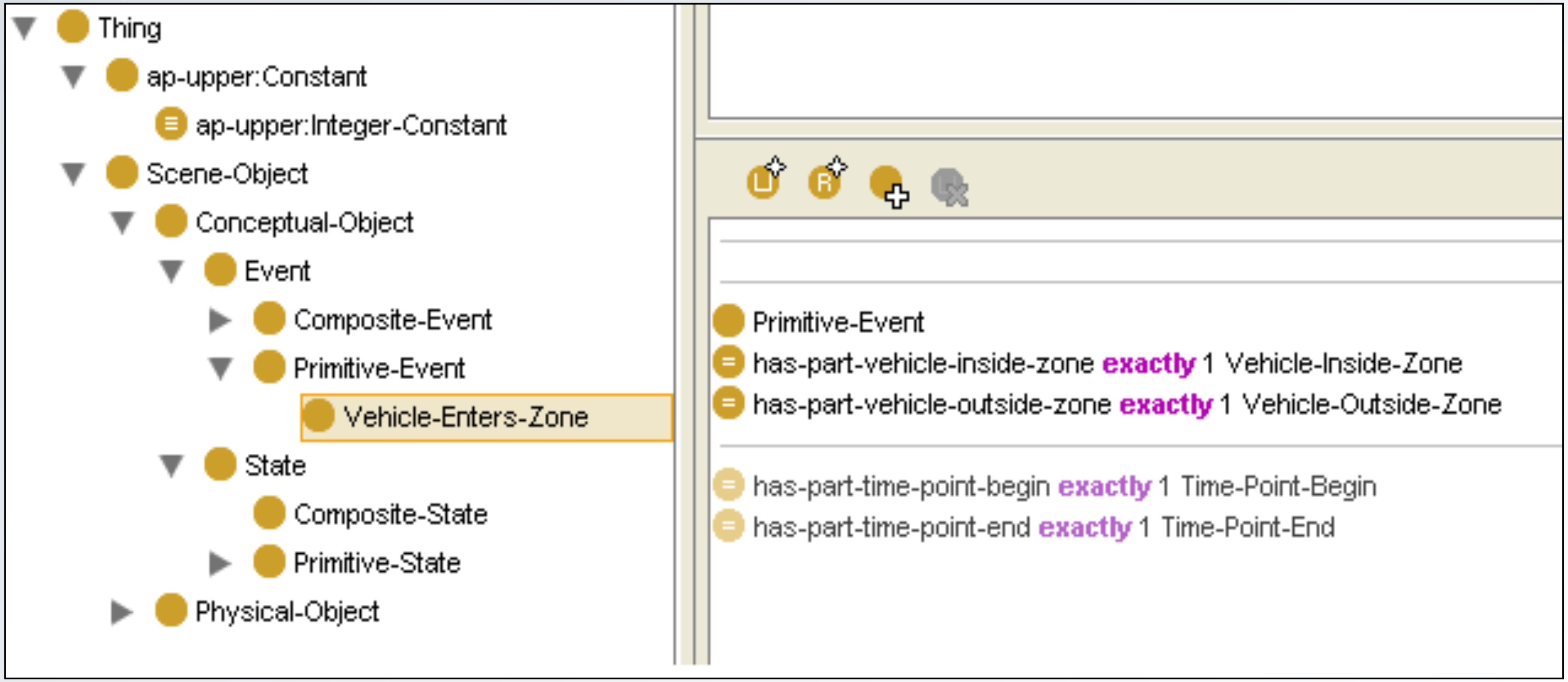
➡ **taxonomical hierarchies**

➡ **partonomical (compositional) hierarchies**

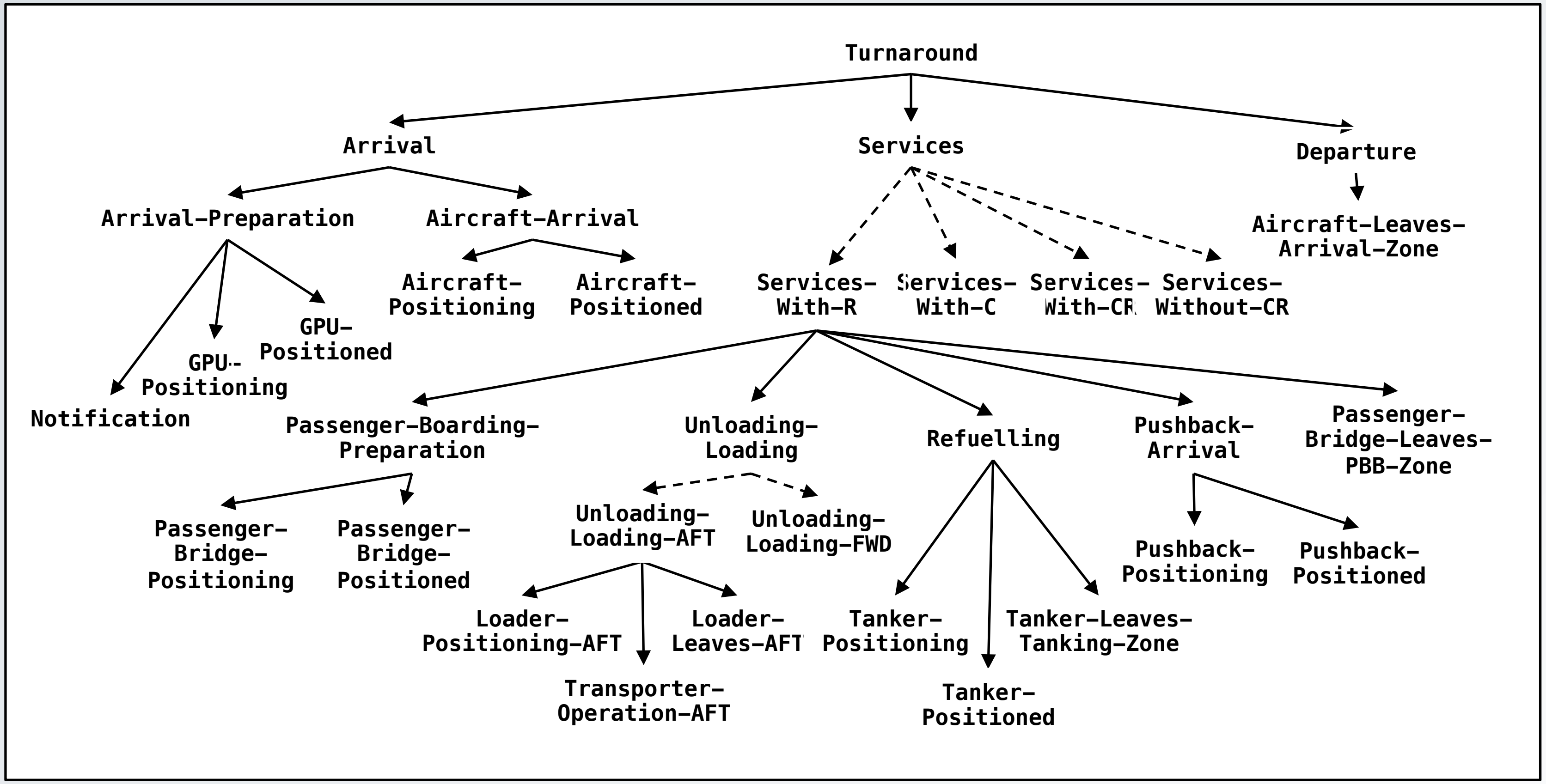
Taxonomical Hierarchy of Airport Service Activities

WISSEN SCHAFFT ZUKUNFT

Taxonomical hierarchy and conceptual activity definition using OWL and the Protégé editor



Partonomical Hierarchy of Airport Service Activities



has-part

has-specialisation

Spatial Constraints for Aircraft Servicing

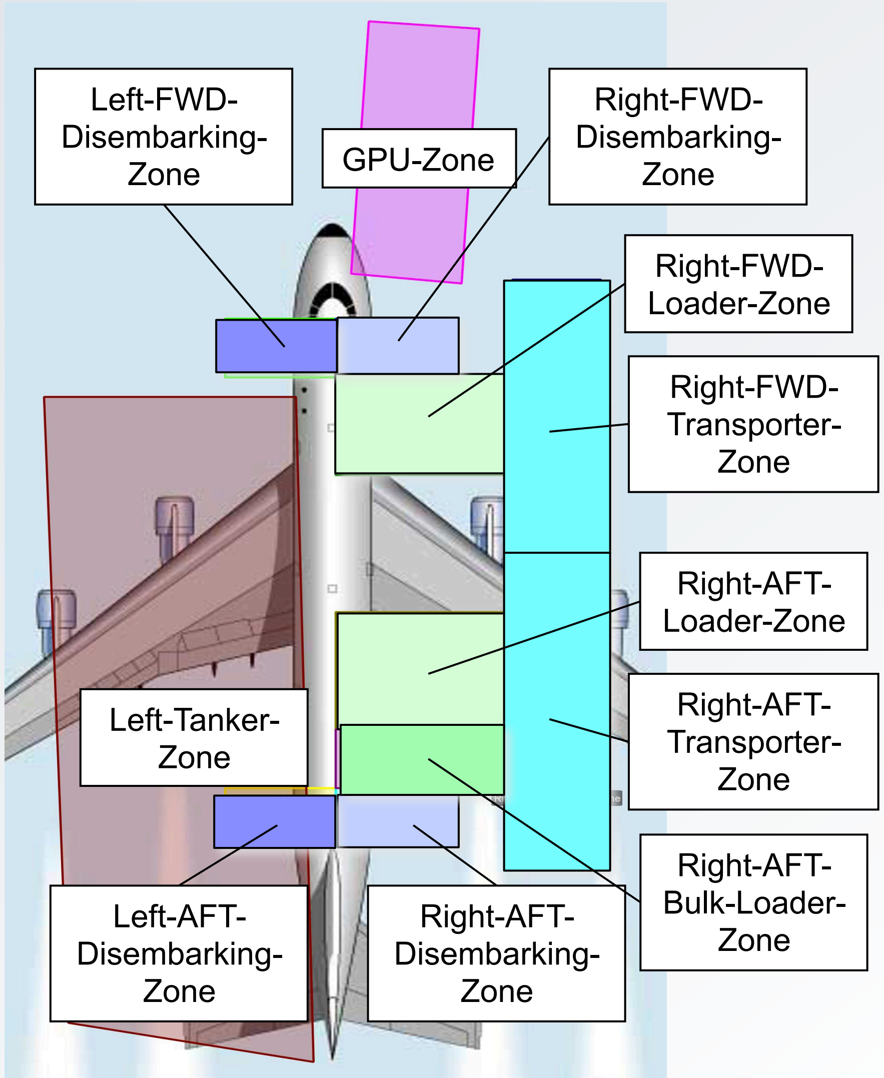
WISSEN SCHAFFT ZUKUNFT

Spatial constraints for service activities in terms of zones and predicates

Vehicle-Enters-Zone

GPU-Inside-GPU-Zone

➡ **Qualitative symbolic spatial representation**



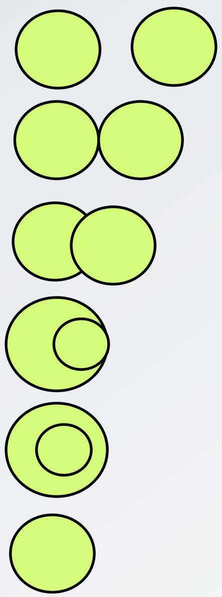
Region Connection Calculus (RCC)

WISSEN SCHAFFT ZUKUNFT

Systematic representation of topological region relations supports reasoning.

<u>Elementary relations of RCC8</u>:

- **disconnected**
- **externally connected**
- **partial overlap**
- **tangential proper part**
- **non-tangential proper part**
- **equal**

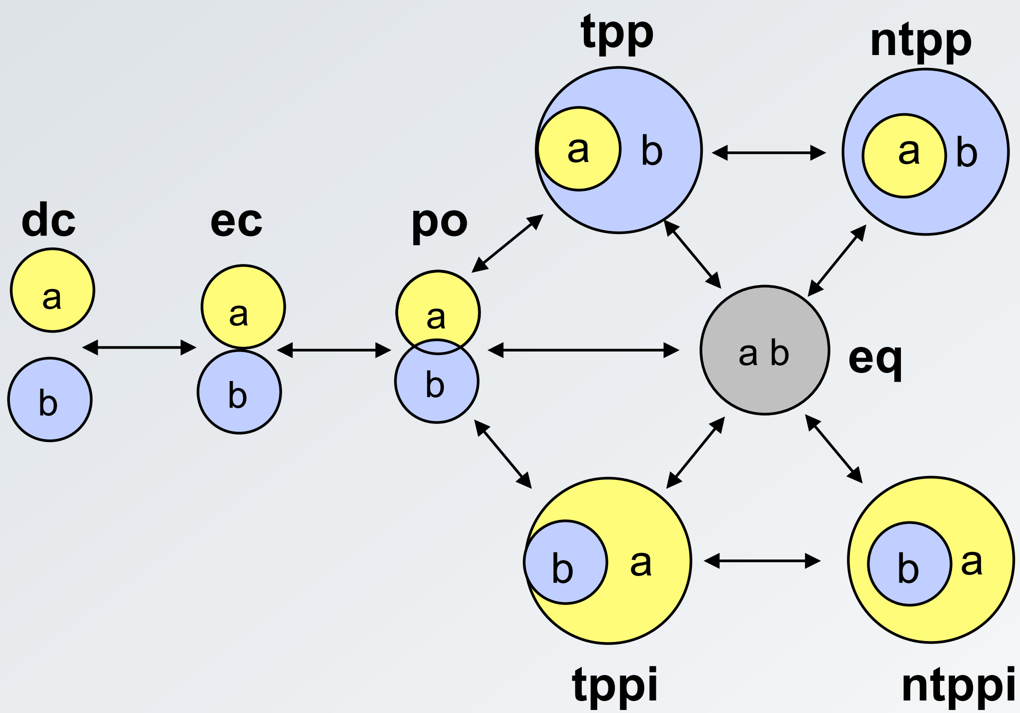


Subsets (RCC5, RCC3) may also be useful.

Combining RCC8 with Time

CCG
WISSEN SCHAFFT ZUKUNFT

Temporal transitions define "conceptual neighborhoods":



Conceptual neighborhoods support basic inferences, e.g. about intermediate stages of time-varying topological region relations.

Temporal Constraints for Aircraft Servicing

Monitoring service activities requires <u>quantitative</u> temporal constraints.

Passenger-Stairs must be positioned not later than 5 minutes after aicraft arrival.

A GPU will stop not later than 1 minute after entering the GPU zone.

In OWL, quantitative constraints can only be represented using the rule extension SWRL or – in OWL 2 – using OWL-RL.

SWRL rules have disadvantages:

- **Not elegantly connected to OWL classes**
- **Reasoning with SWRL is undecidable (in general)**

Example of Temporal SWRL Rule

WISSEN SCHAFFT ZUKUNFT

OWL class definition of a vehicle visiting a zone

Visit ⊑ Composite-Event ⊓
has-part1 exactly 1 Vehicle-Enters-Zone ⊓
has-part2 exactly 1 Vehicle-Stopped-Inside-Zone

SWRL rule premise establishes variable names

Visit(?vis)
∧ has-part1(?vis, ?veh-enters)
∧ has-part2(?vis, ?veh-stopped)
∧ has-start-time(?vis, ?vis-st)
∧ has-finish-time(?vis, ?vis-ft)
∧ has-time-point(?veh-enters, ?veh-enters-tp)
∧ has-agent(?veh-enters, ?veh-enters-ag)
∧ has-zone(?veh-enters, ?veh-enters-zn)
∧ has-start-time(?veh-stopped, ?veh-stopped-st)
∧ has-agent(?veh-stopped, ?veh-stopped-ag)
∧ has-zone(?veh-stopped, ?veh-stopped-zn)
→

SWRL rule consequence specifies identity constraints and temporal constraints

?vis-st = ?veh-enters-tp
∧ ?vis-ft = ?veh-stopped-ft
∧ ?veh-enters-ag = ?veh-stopped-ag
∧ ?veh-enters-zn = ?veh-stopped-zn
∧ ?veh-enters-tp ≤ ?veh-stopped-st

Logic-based Aggregate Representations

Spatiotemporal information fusion with OWL?

- **OWL is a standardised ontology language**
  - definition of properties, aggregate taxonomies and partonomies
  - awkward definition of quantitative constraints
- **Powerful Description Logic reasoners support OWL**
  - no support for stepwise recognition
  - no support for constraint solving
- **Crisp relations**
  - fuzzy or probabilistic information cannot be represented

Spatiotemporal Representations in SCENIOR

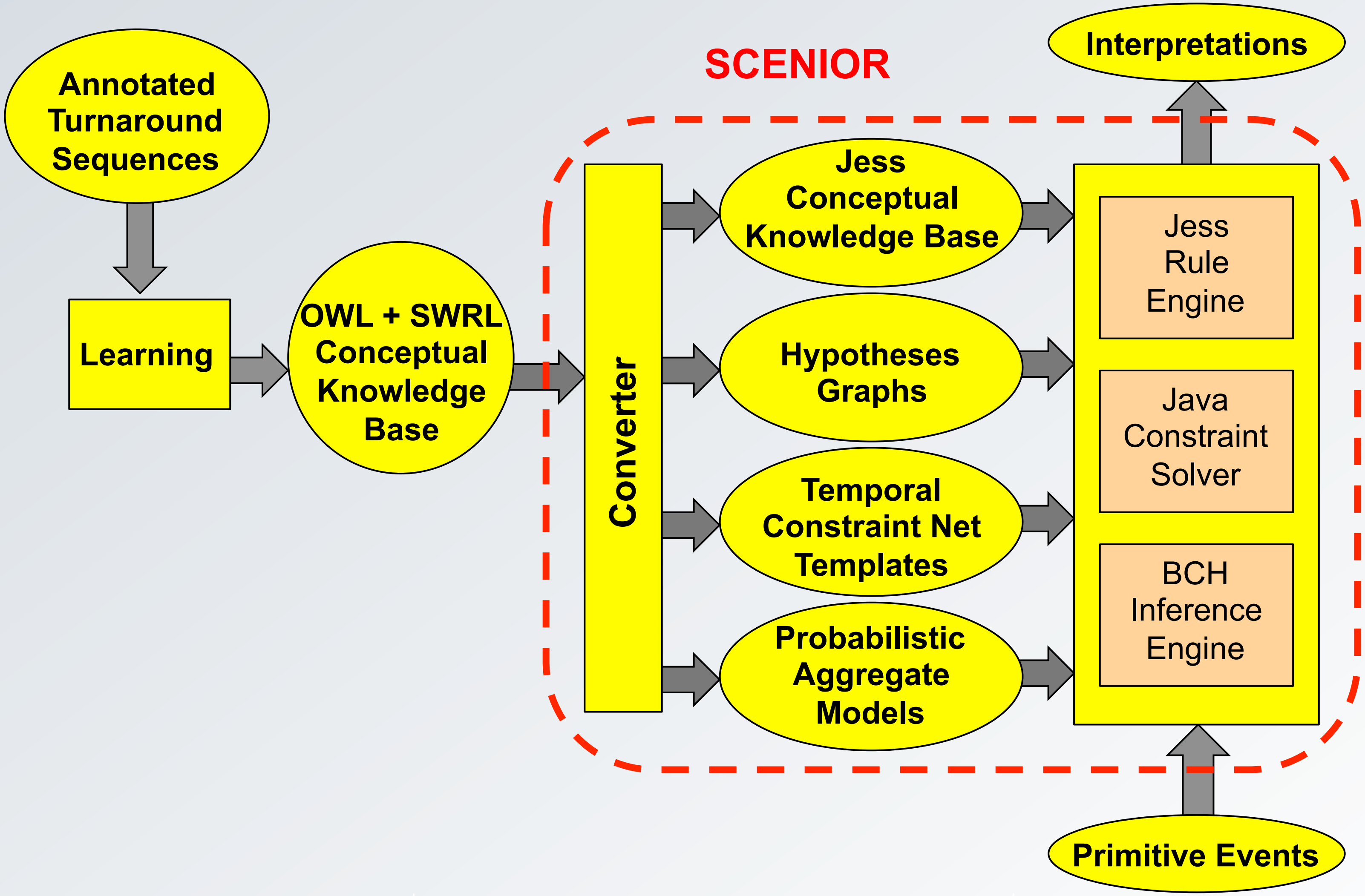
WISSEN SCHAFFT ZUKUNFT

SCENIOR (SCENe Interpretation with Ontology-based Rules) is an operational system for interpreting aircraft service activities developed by HITeC and the Cognitive Systems Laboratory at Hamburg University.

Locations	➡	**fixed zones**
Time points	➡	**quantitative values**

- **in a temporal constraint net**
- **in Bayesian Probabilistic Hierarchiesy (BCHs)**

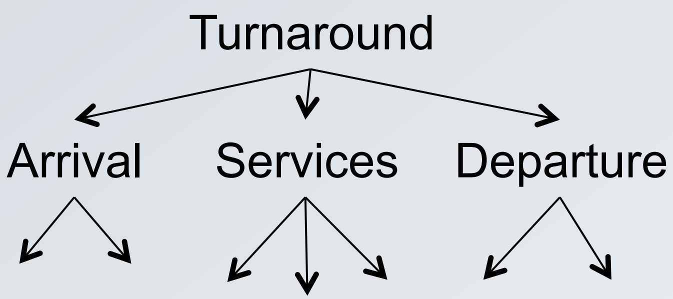
SCENIOR System Structure



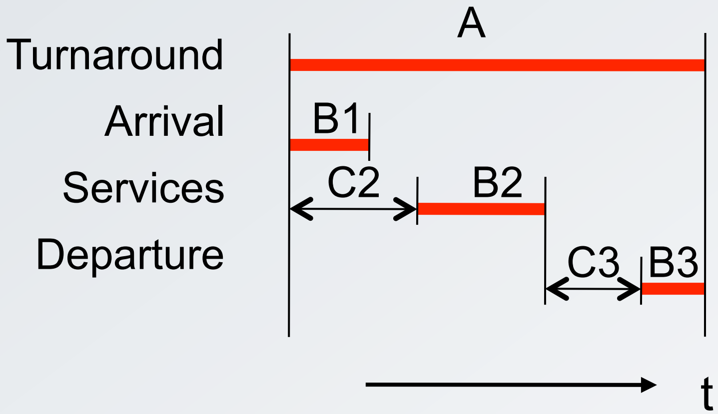
Bayesian Compositional Hierarchies

WISSEN SCHAFFT ZUKUNFT

Aggregate partonomy



Temporal aggregate structure



Aggregate JPD: $P_{Turnaround}(A\ B_1\ C_2\ B_2\ C_3\ B_3)$

➡ $P'_{Turnaround}(B_1\ C_2\ B_2\ C_3\ B_3 \mid A)$

Scene JPD for turnaround model m is product of aggregate JPDs (as in Bayesian Networks)

$$P^m_{Scene} = p_m\ P'_{Turnaround}\ P'_{Arrival} \cdots P'_{Refuelling} \cdots P'_{Pushback}\ P_{clutter}$$

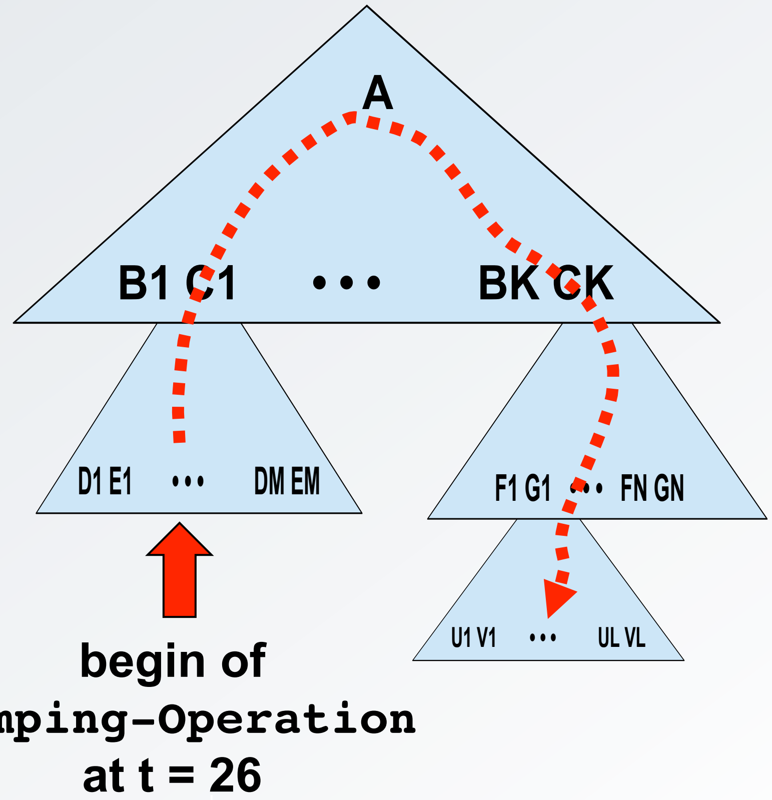
Probability Propagation

WISSEN SCHAFFT ZUKUNFT

Representation of durations and offsets by Gaussians allows efficient probability update

- **Enter begin or end of events**
- **Propagate change throughout BCH**
- **Estimate non-instantiated temporal variables**

➡ **obtain dynamic priors (context-dependent)**



Summary

- **Spatial and temporal information can be mapped into feature space**
- **A rich feature space technology is available**
  - Distance measures
  - Clustering methods
  - Supervised learning methods
  - Probabilistic models
- **Spatiotemporal aggregate models are useful for highly structured domains**
  - Representation in standardised ontology language OWL
  - Reasoning systems
  - Probabilistic support by BCHs