Unsupervised Learning From Examples

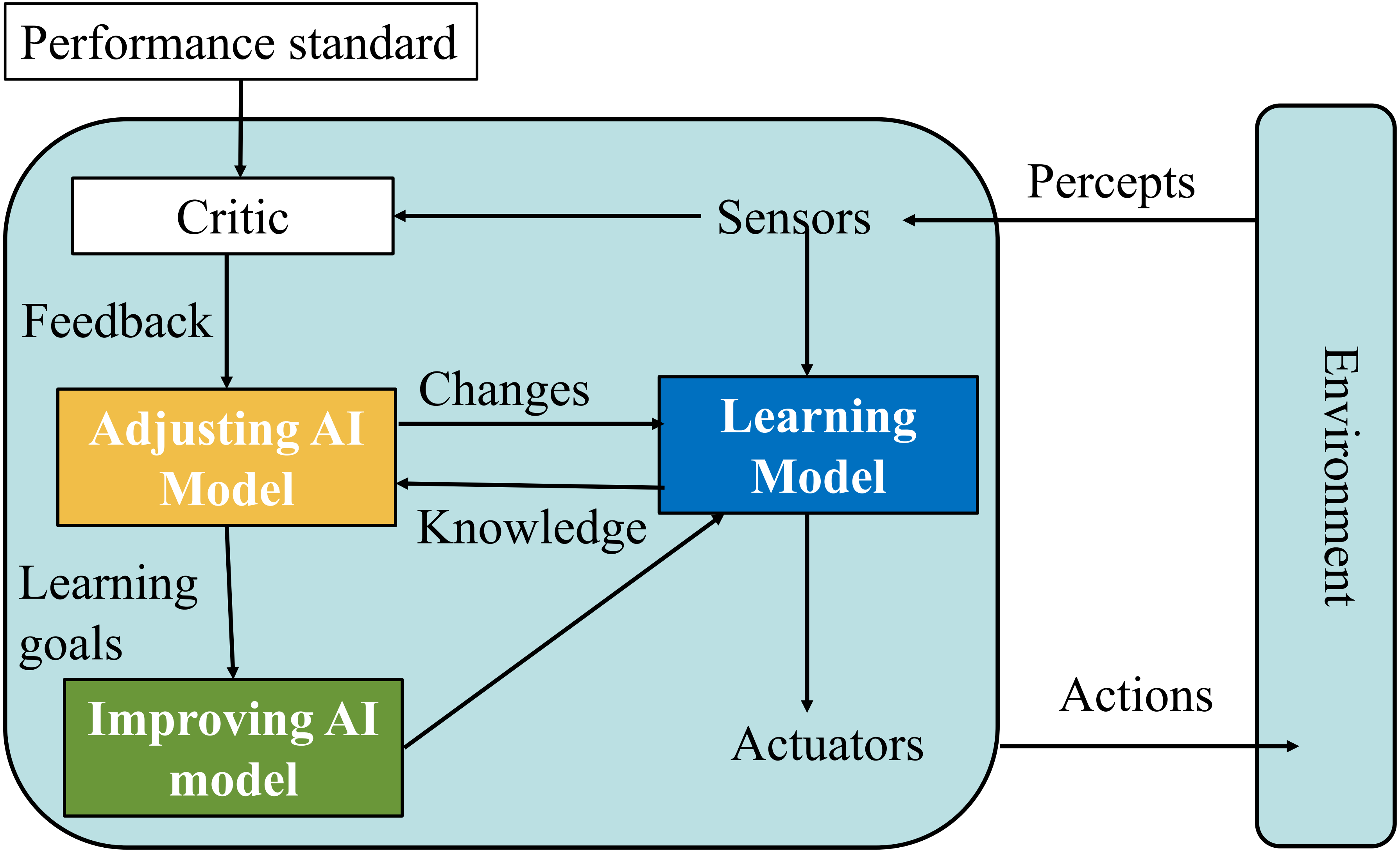
Dr. Lotfi ben Othmane
University of North Texas

Administrative

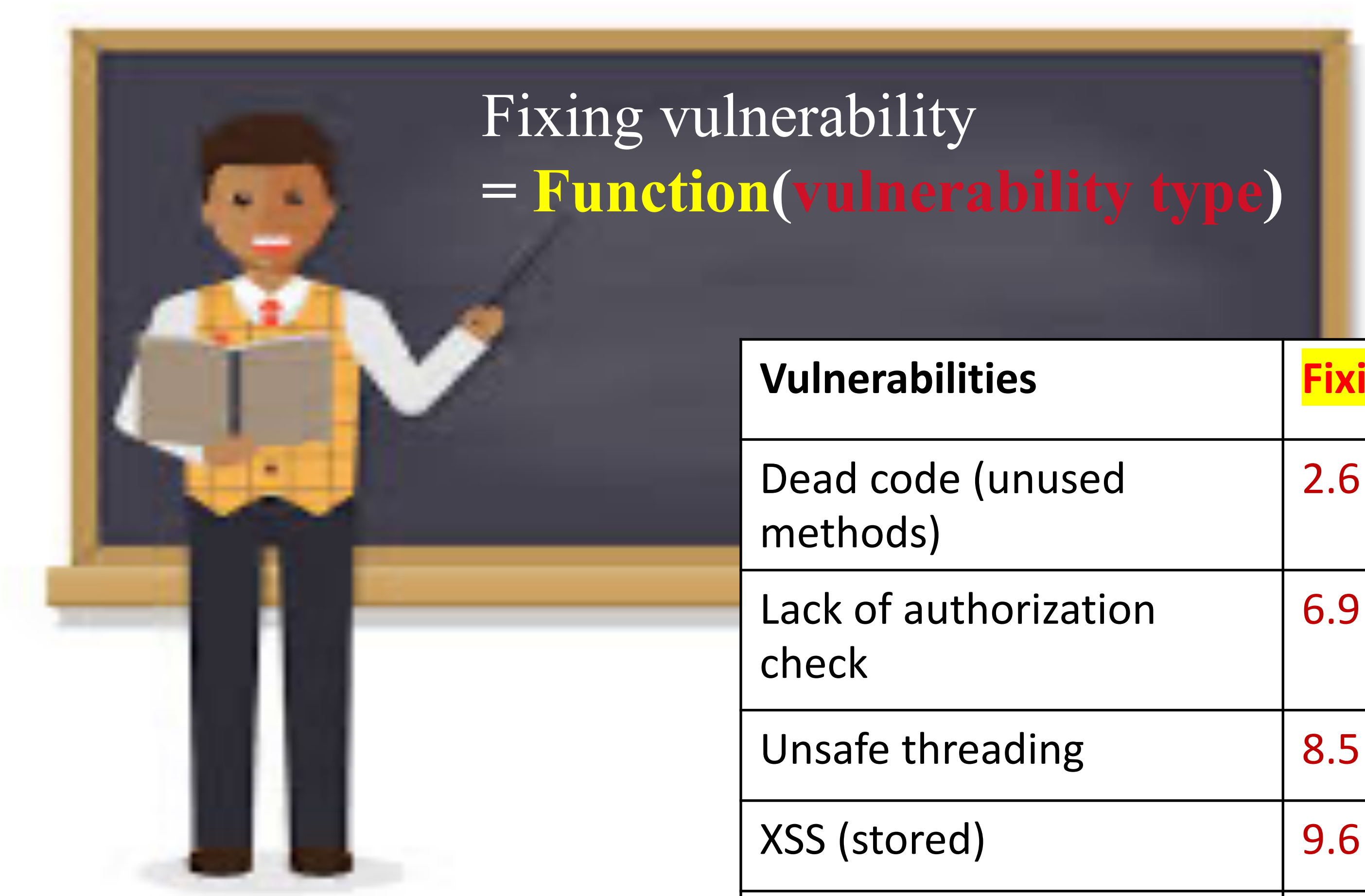
- Assignment 1 is due on Feb 20, 2024
- Project phase 1 is due on Feb 22, 2024 – Select a project from the provided list

- Quiz 2 is Feb 22, 2024

Learning Agents



Associate Percepts to Outputs/Labels



Vulnerabilities	Fixing time
Dead code (unused methods)	2.6
Lack of authorization check	6.9
Unsafe threading	8.5
XSS (stored)	9.6
SQL injection	12.3

Supervised vs Unsupervised ML

- Supervised machine learning is about discovering patterns relating data attributes with data labels.
  - The uses are: regression, classification, and forecasting.

- Unsupervised machine learning is about analyzing and clustering datasets.
  - The uses are clustering, dimensionality reduction, and association.

Case Study – Architecture Recovery

- Prescriptive architecture describes the expected architecture of software (often designed one)
- Descriptive architecture is the as-implemented architecture of software

- Descriptive architecture and prescriptive architecture are often different

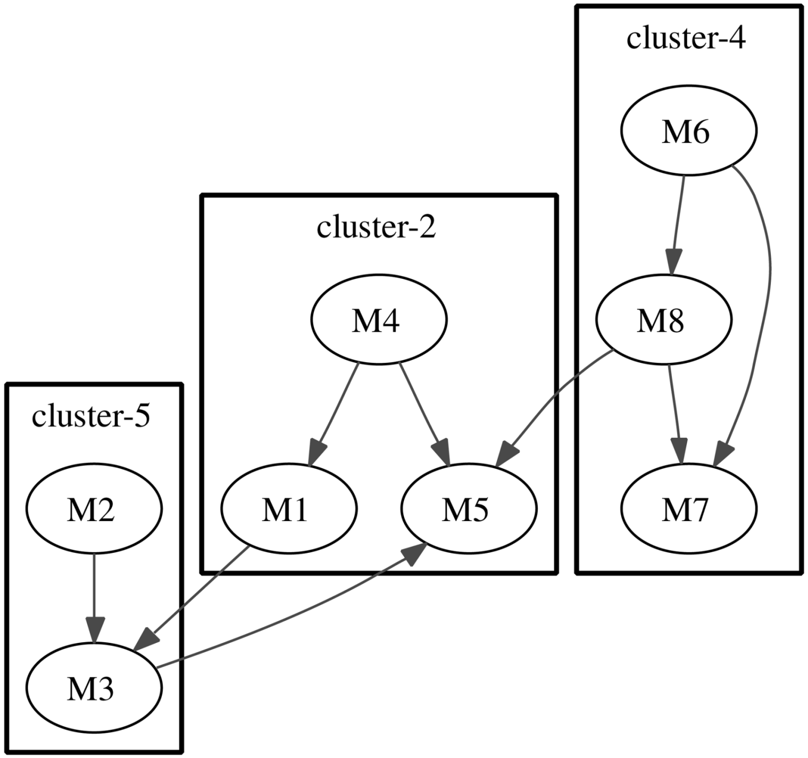
Case Study – Architecture Recovery– Cont.

- Architecture recovery is the extraction and analysis of a software architecture
- Current tools cluster the software code into packages
- Recovery techniques often are based on the call graph of the software

Graph Clustering

Architecture recovery becomes a clustering problem

- Each method defines its own clustering feature



Case Study – Architecture Recovery – Cont.

- Call graph represents the calling relationships between nodes. Each node represents a function (or module) and each edge *(f, g)* represents calls of function (or module) *f* to function (or module) *g*.

- Use code analysis tools to extract the CFG.

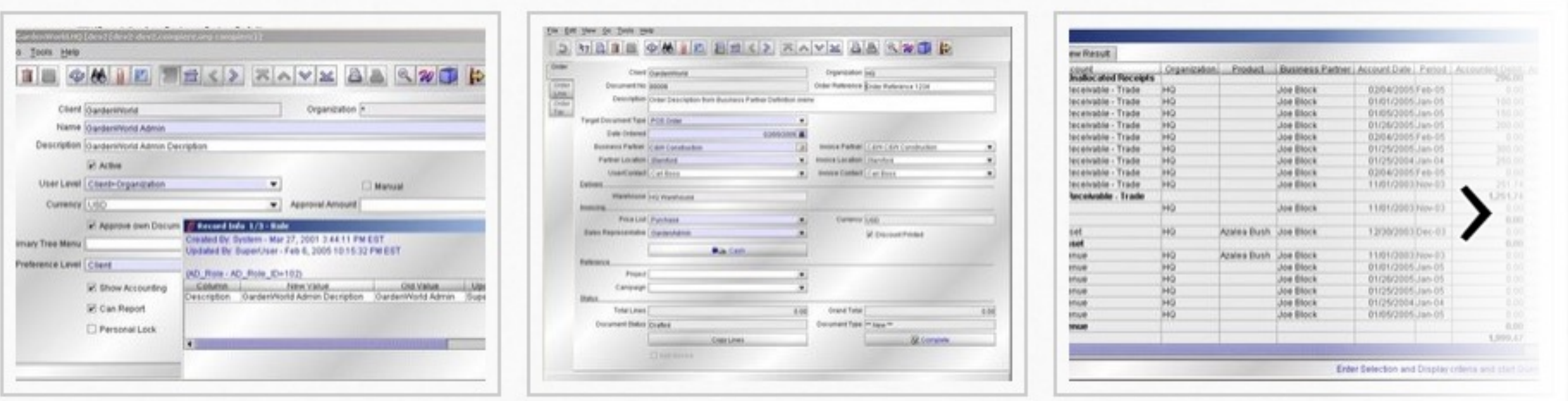
Application- Compiere

Overview

Compiere ERP+CRM is the leading open source ERP solution for...

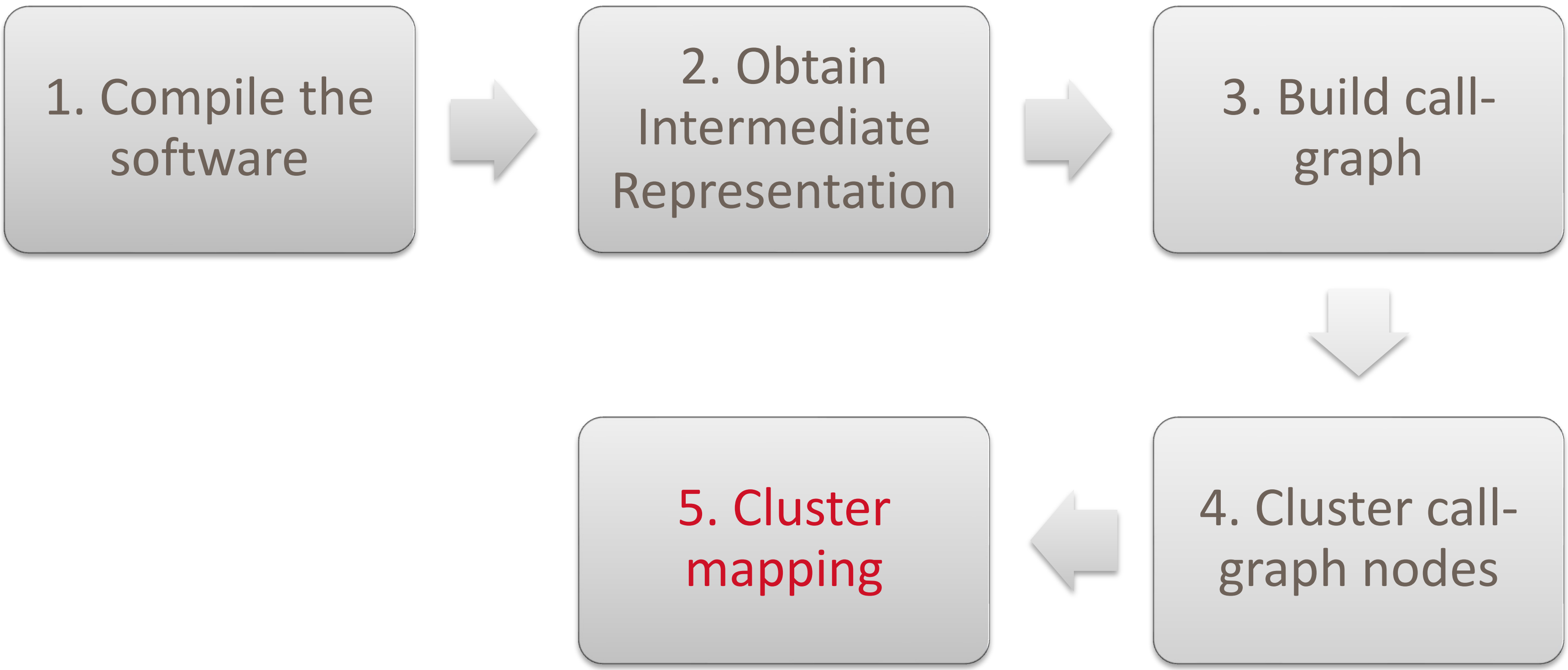
Read More

Compiere ERP + CRM Business Solution Web Site »

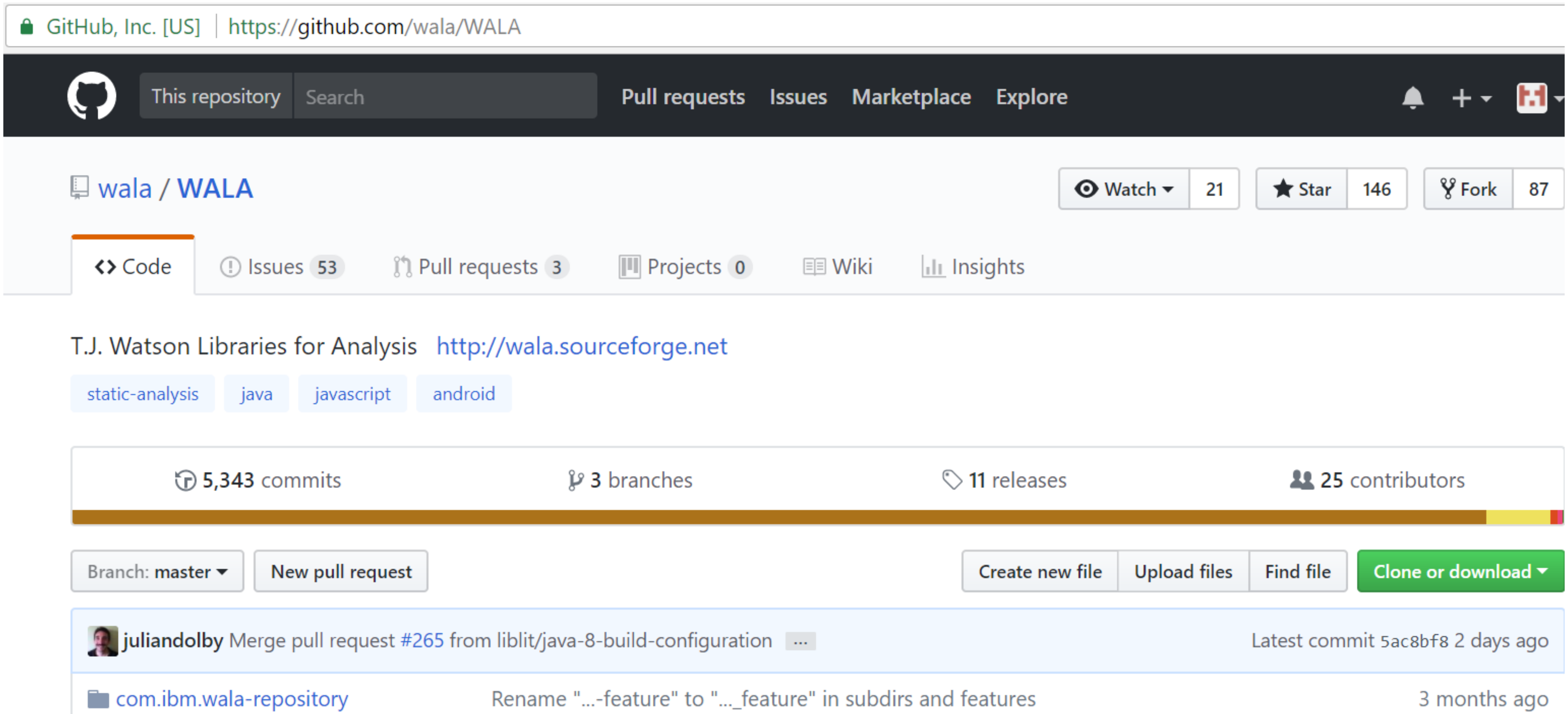


https://sourceforge.net/projects/compiere/

Case Study – Architecture Recovery – Cont.



Control Flow Graph Generated by WALA



Data Cleaning Challenges

1. Code analysis tools add fake nodes.
2. Code includes dependency – e.g., Java-based code includes JRE java methods.
3. Methods may have similar names but in different modules – may confuse code analysis.
4. The number of clusters is huge, e.g., 2000

Concepts for Software Architecture

- Assumes: well-designed software systems are organized into cohesive subsystems that are loosely interconnected.

- Interconnectivity - dependencies between the modules of two distinct subsystems
- Intra-connectivity - dependencies between the modules of the same subsystem
- Modularization Quality – trade-off between Interconnectivity and Intra-connectivity

Bunch Metric

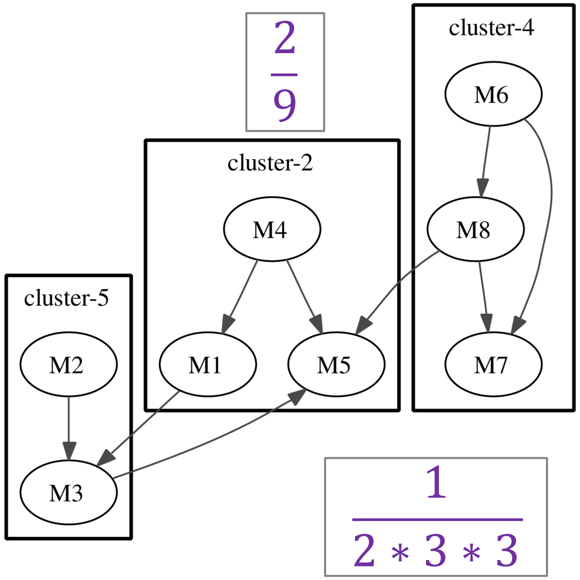
- Intra-connectivity - Coefficient of number of edges in the cluster to potential number of edges in the cluster

$$A_i = \frac{\mu_i}{N_i^2}$$

- Interconnectivity - Coefficient of number of edges between cluster *i* and cluster *j* to double the number of nodes of cluster *i* multiply by the number of nodes of cluster *j*. (0 if in the same cluster.)

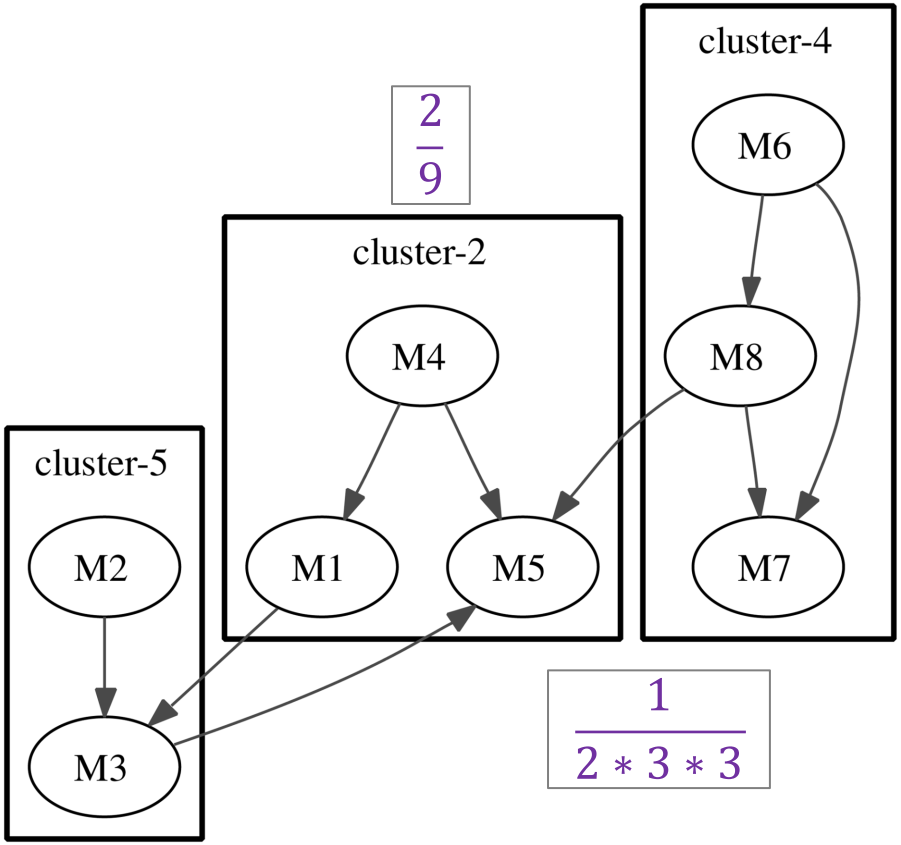
$$E_{i,j} = \frac{\varepsilon_{i,j}}{2\ X\ N_i\ X\ N_j}$$

- Modularization Quality - $\begin{cases} \frac{\sum_{i=1}^{k} A_i}{k} - \frac{\sum_{i,j=1}^{k} E_{i,j}}{\frac{k(k-1)}{2}} \\ A_1 \qquad (k = 1) \end{cases}$

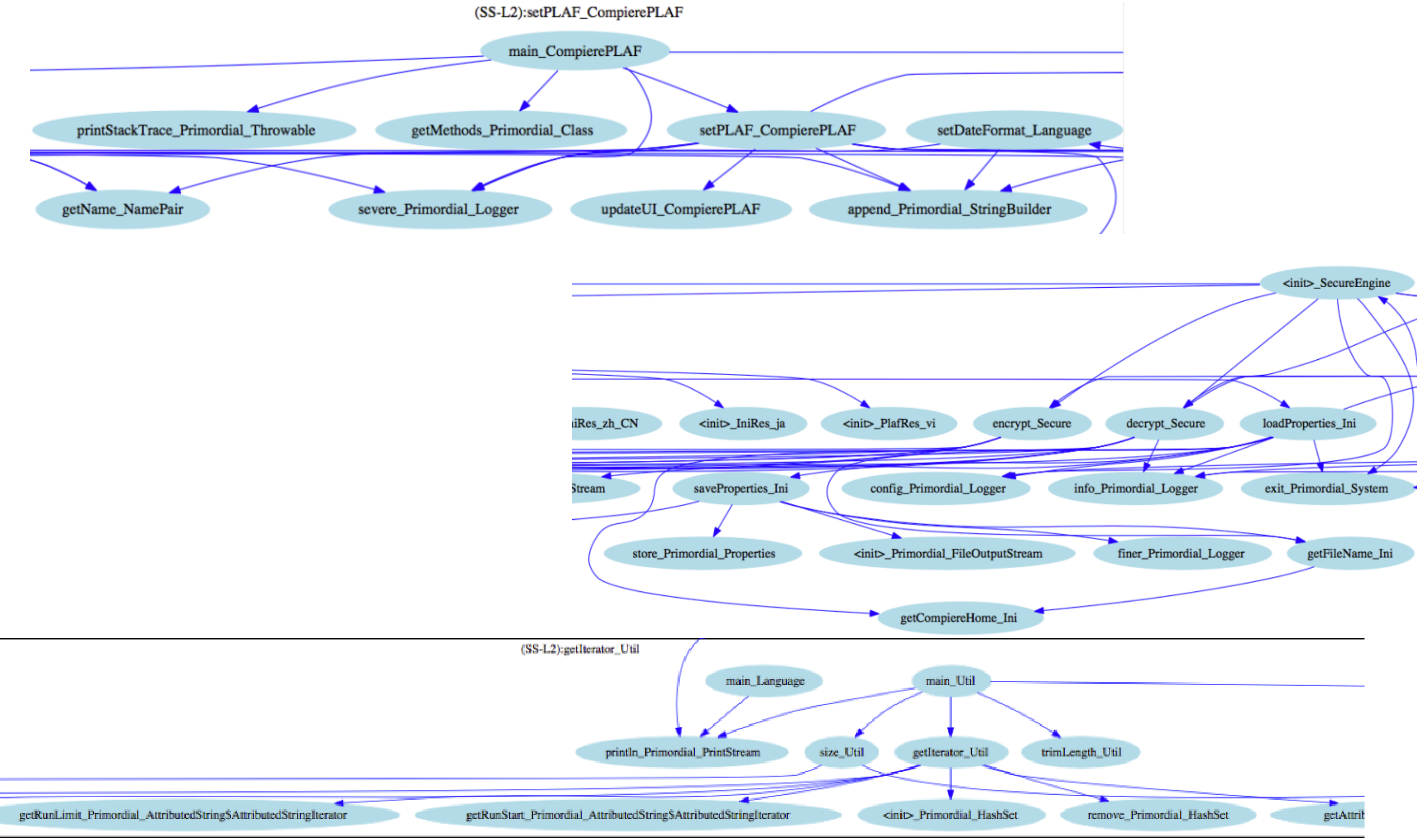


Bunch Metric

It uses hill-climbing and genetic algorithms to solve the optimization problem



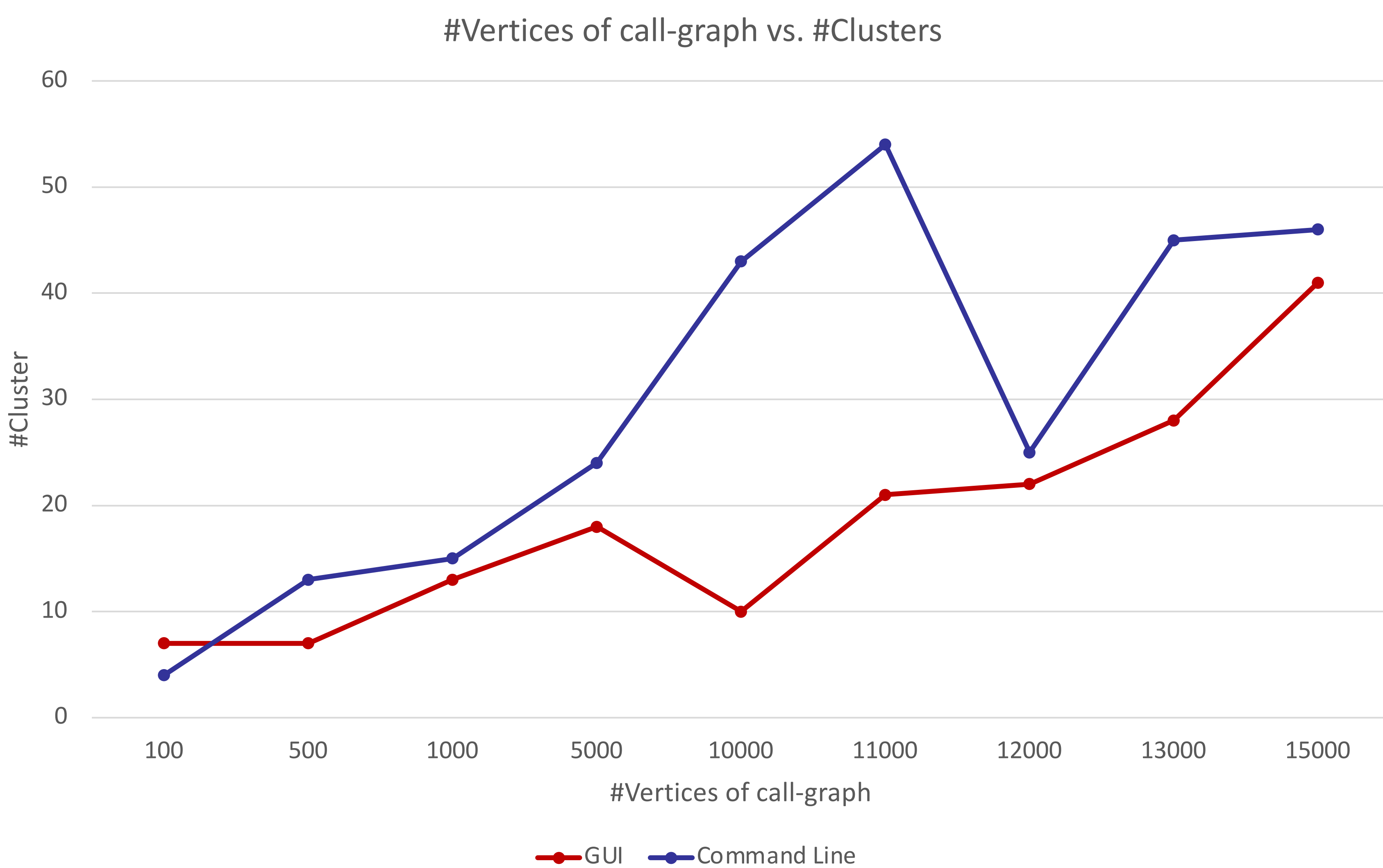
Case Study – Compiere - Clusters



Case Study – Compiere - Clusters

1. Util - main class for utilities
2. New Instance - when user logs in, creates new instance with respective look & feel and language
3. Logger - generation and storage of logs
4. Secure Engine - responsible for implementing security policies within the application and initializing security
5. NameValuePair - stores/retrieves/modifies user related data
6. List Resources - lists user resources on login
7. Main_CompierePLAF - provides look & feel
8. GetLanguage - retrieves the language for user
9. Encrypt_SecureEngine - provides data encryption capabilities
10. Hashing - stores hashmap of user data

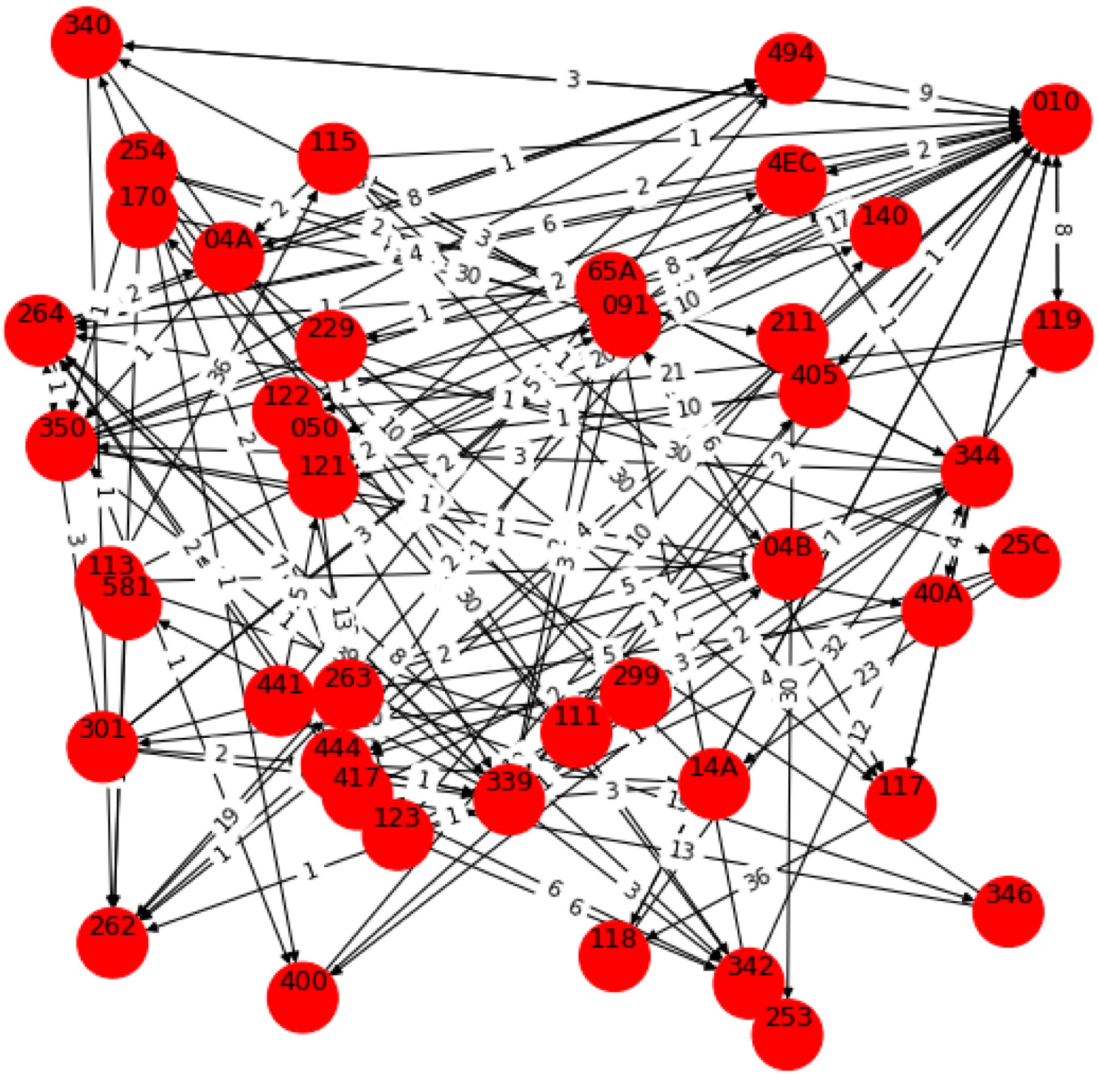
Clustering Capabilities



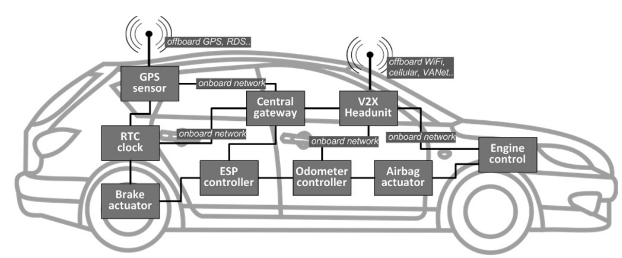
So

- Clustering is about
    - Partitioning
    - ….
- Metric: Use a distance metric
    - Minimize intra-distance metric
    - Maximize inter-distance metric
- The quality depends on the algorithm and distance metric

Clustering – How the Components of the Car Collaborate

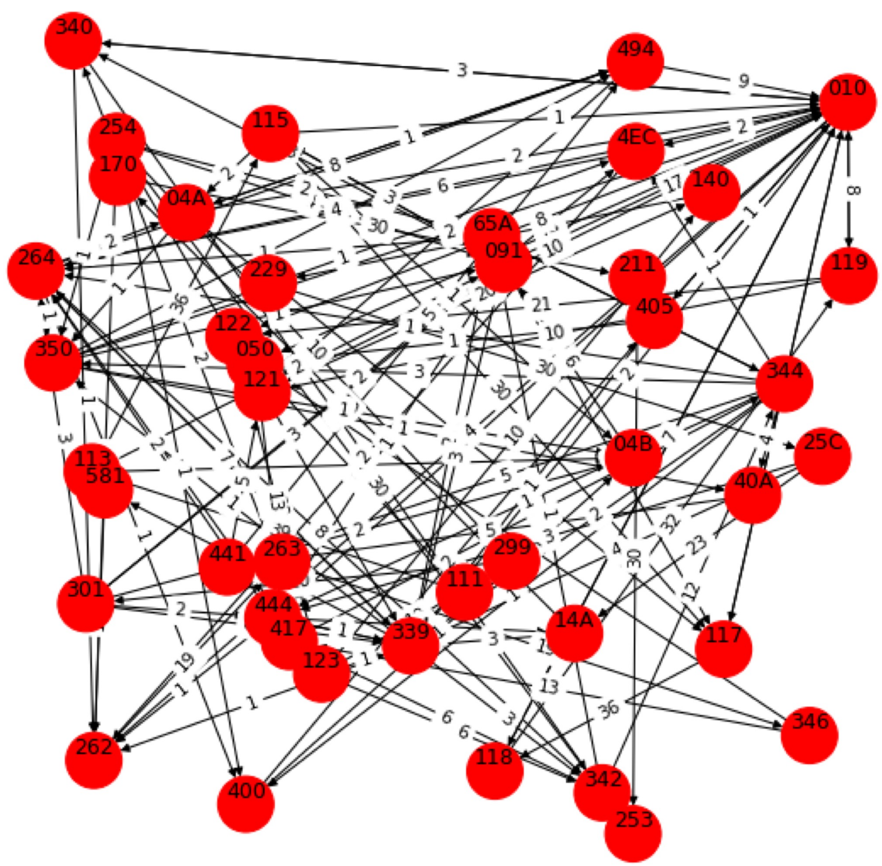


Collaboration is measured by the number of exchanged messages



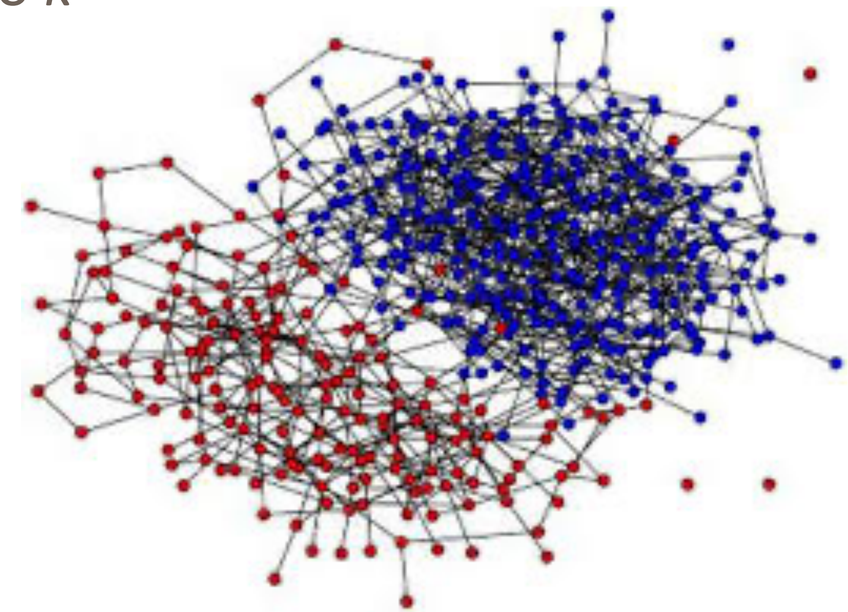
Clustering – Cluster the ECUs based on functionalities

Which components control the speed increase?



Clustering Using K-means

- *K-means* partition the dataset into *k* clusters



- Each cluster has cluster center called centroid
- *K* is an input to the algorithm
- Aims to minimize the sum of the distances between the within-cluster data points and associated centroids.

$$\sum_{i=1}^{k} \sum_{x \epsilon S_i} \| x - y_i \|^2$$

Clustering Using K-means

Algorithm

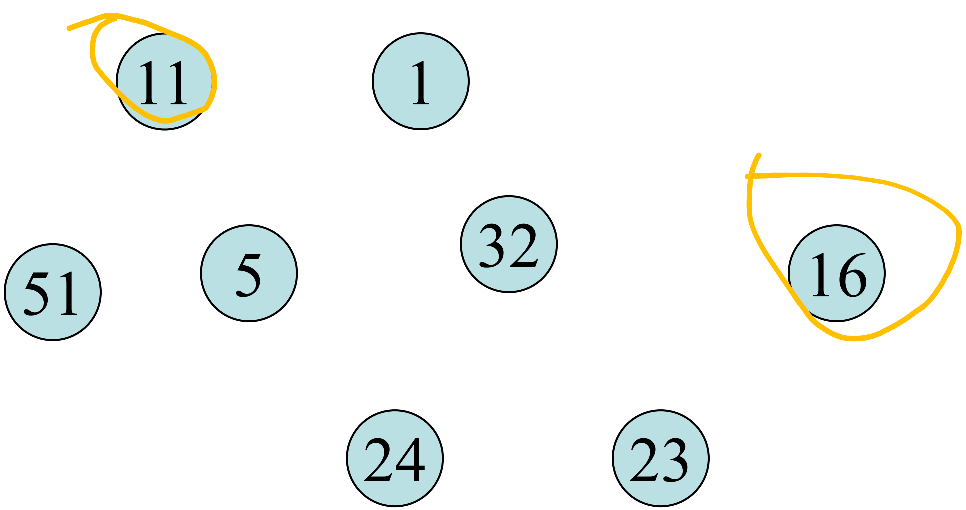
1. Select initial *k* data points $m_1^1\ m_2^1 \dots \dots . m_k^1$
2. For each round *t* assign each data point to the nearest cluster *i*
   $C_i^t = \{x_p : \| x_p - m_i^t \|^2 \leq \| x_p - m_j^t \|^2\ \forall\ 1 \leq j \leq k\}$
3. Recompute the means of the clusters
   $m_i^{t+1} = \frac{1}{|C_i^t|} \sum_{x_j \epsilon C_i^t} x_j$
4. Stop when the assignment no longer change

$$\sum_{i=1}^{k} \sum_{x \epsilon S_i} \| x - y_i \|^2$$

Exercise: Clustering Using K-means

Cluster the students by age into 2 groups

Select centers 11 and 16 as starting points



Algorithm

1. Select initial k data points $m_1^1\ m_2^1 \ldots \ldots .\, m_k^1$
2. For each round t assign each data point to the nearest cluster i $C_i^t = \{x_p : \| x_p - m_i^t \|^2 \leq \| x_p - m_j^t \|^2\ \forall\ 1 \leq j \leq k\}$
3. Recompute the means of the clusters $m_i^{t+1} = \frac{1}{|C_i^t|} \sum_{x_j \epsilon C_i^t} x_j$
4. Stop when the assignment no longer change

Exercise: Clustering Using K-means

$M_{11}=11$

$M_{12}=16$

$C_{11}=\{1,5,11\}$

$C_{12}=\{16,23,24,32,51\}$

$M_{21} = 5.6 \sim 5$

$M_{22}=29.2 \sim 32$

$C_{21}=\{1,5,11,16\}$

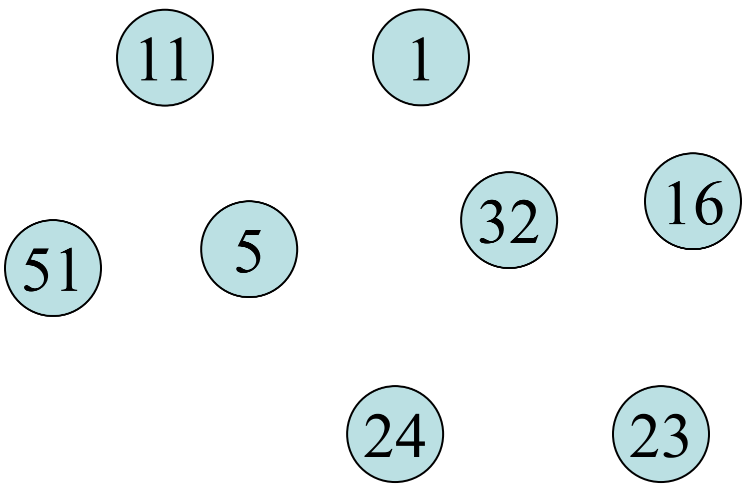
$C_{22}=\{23,24,32,51\}$

$M_{21} = 8.25 \sim 11$

$M_{22}= 32.5 \sim 32$

$C_{21}=\{1,5,11,16\}$

$C_{22}=\{23,24,32,51\}$



Clustering Using K-means

- Frequently use method
- Positive:
  - Easy to understand
  - Efficient
- Negative
  - It is sensitive to outliers
  - You need to specify *k*
  - It kind of favors balanced datasets

Another Method – Hidden Markov Model

Probabilistic Reasoning

Hidden states

Observations

P(Sunny/Umbrella)	P(Rainy/Umbrella)
P(Sunny/No Umbrella)	P(Rainy/No umbrella)

Probabilistic Reasoning

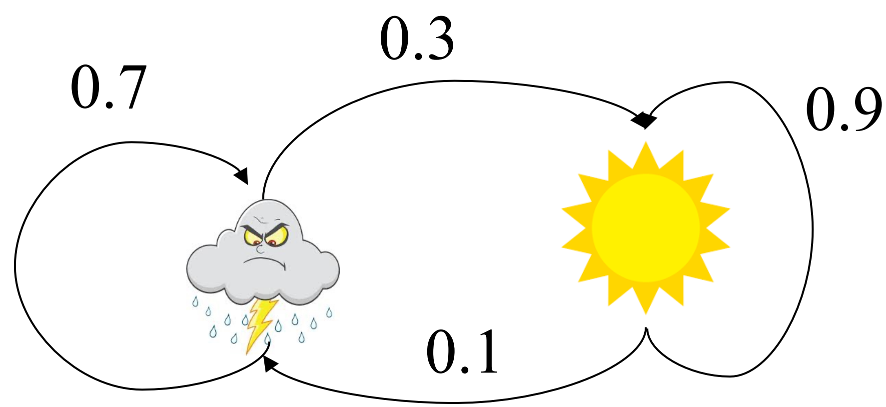
1 2 3. 4. 5

Sat. Sun. Mond. Tues. Wed.

States ={Rain, Sun}

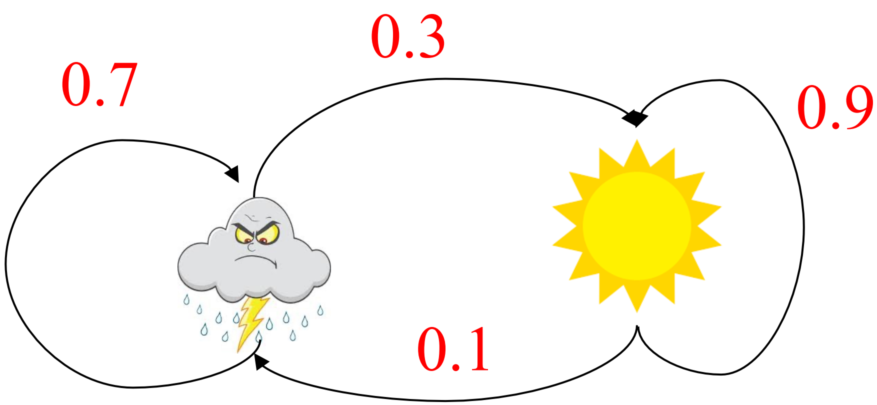
Sunday is sunny with P=1

S_{t-1}	S_t	Prob
Sun	Sun	0.9
Sun	Rain	0.1
Rain	Sun	0.3
Rain	Rain	0.7



Probabilistic Reasoning

What is the probability of Sun on day t?



Conditional Probability

Conditional probability: $P(x, y) = P(x/y) \times P(y)$

Chain rule: $P(x_1, x_2, ..x_n) = P(x_1)P(x_2/x_1)P(x_3/x_2, x_1)....$

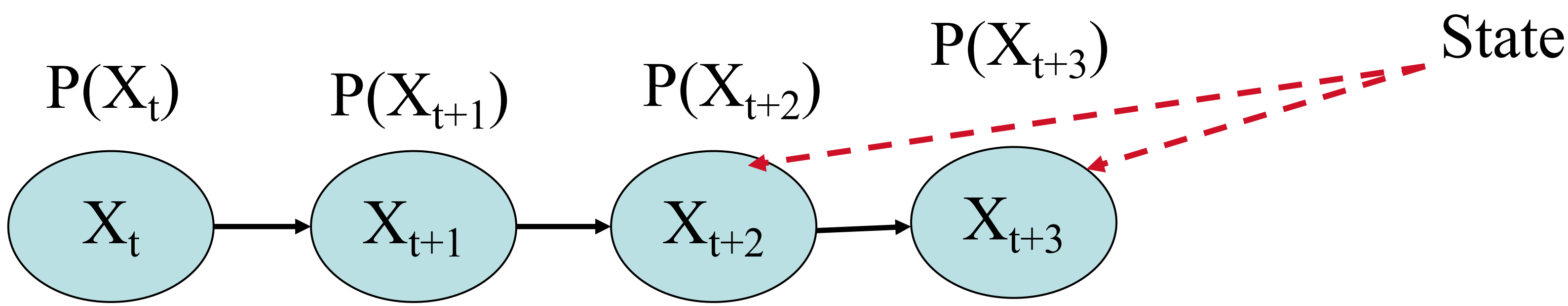
$= \prod_{i-1}^{n} P(x_i | x_1 \ldots x_{i-1})$

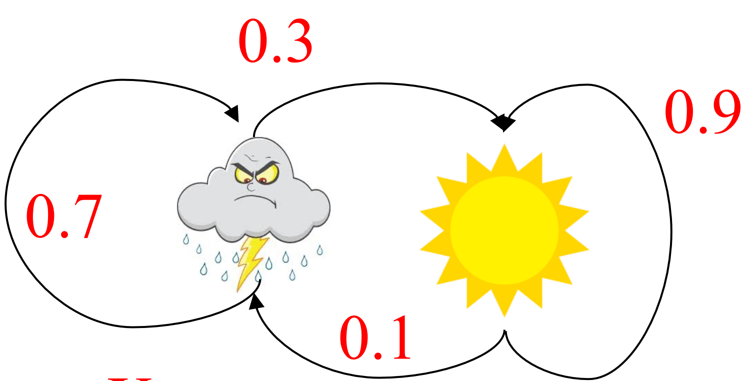
X and Y are independent iff $\forall x \in X, y \in Y, P(x, y) = P(x)P(y)$

Markov Models

Transition probabilities Specify how the state evolves over time.

Stationarity assumption: Transition probability is the same at all times





In Markov models, State X_t depends only on X_{t-1}

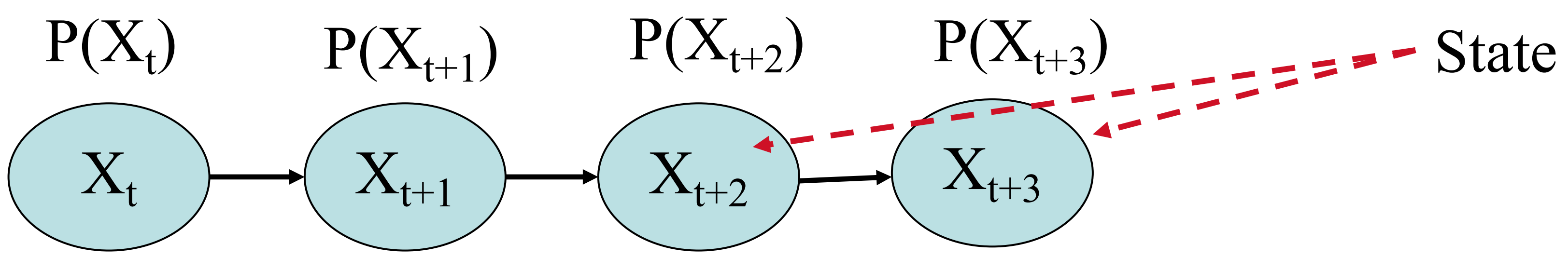
Markov Models

What is the probability of Sun on day *t*?

The probability at time *t* depends on probability at *t-1* and is independent of previous time steps.

$$P(X_t) = \sum P(X_t, X_{t-1})$$
$$= \sum P(X_t/X_{t-1}) * P(X_{t-1})$$

X_t is independent of $X_{t-2}, X_{t-3}, \ldots$



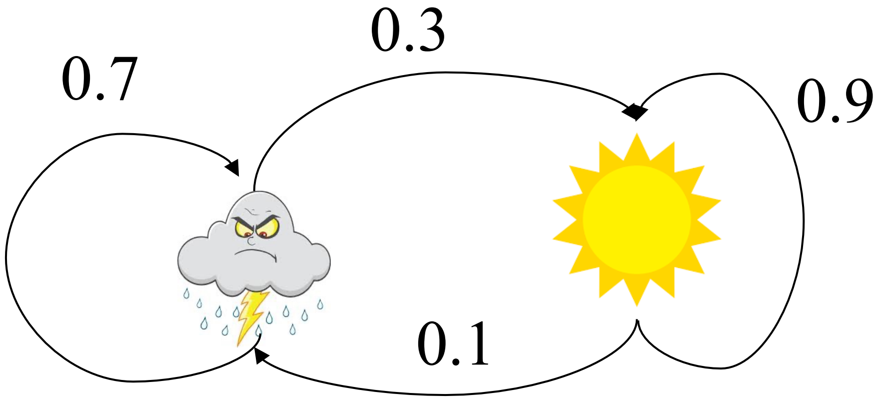
Probabilistic Reasoning

What is the probability of Sun on day 2 given it is sun on day 1?

Sunday is sunny with P=1

S_{t-1}	S_t	Prob
Sun	Sun	0.9
Sun	Rain	0.1
Rain	Sun	0.3
Rain	Rain	0.7

Give it a try



Probabilistic Reasoning

What is the probability of Sun on day 2 given it is sun on day 1?

S_{t-1}	S_t	**Prob**
Sun	Sun	0.9
Sun	Rain	0.1
Rain	Sun	0.3
Rain	Rain	0.7

Sunday is sunny with P=1

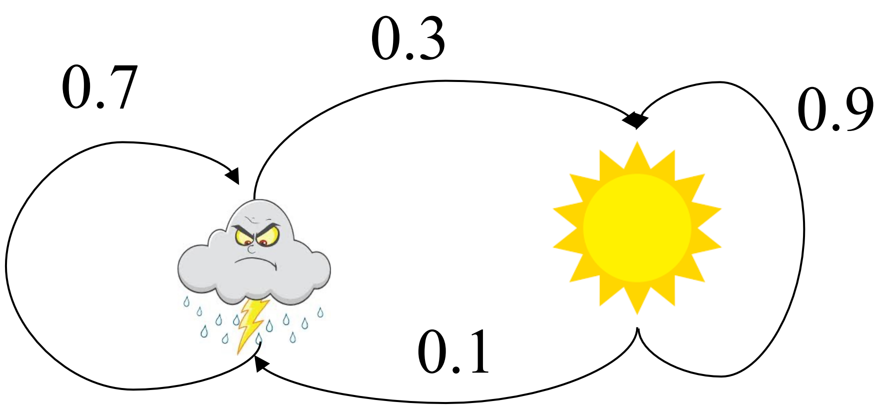
$P(X_2=Sun) =$

$P(X_2=Sun/X_1=Sun)*P(X_1=Sun)$

$+P(X_2=Sun/X_1=Rain)*P(X_1=Rain)$

$= 0.9*1.0+0.3*0.0$

$=0.9$



Markov Models

$$P(X_t) = \sum P(X_t/X_{t-1}) * P(X_{t-1})$$

- $P(X_1=\text{Sun}) = 1$

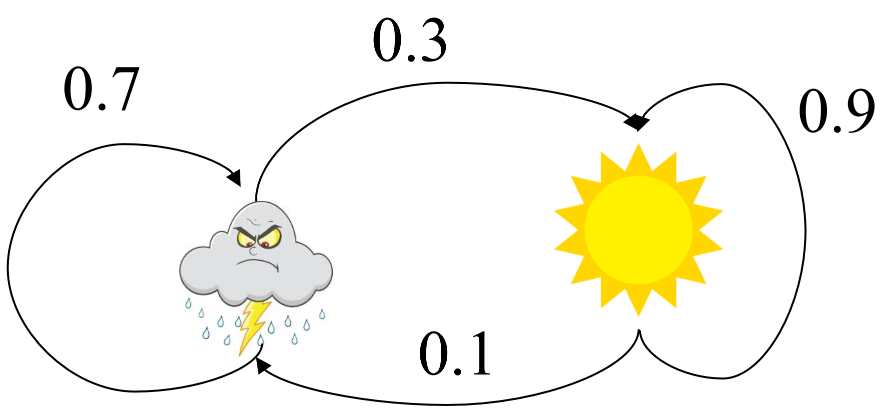
$$\begin{pmatrix} \color{red}{Sun} \\ \color{red}{Rain} \end{pmatrix} \begin{pmatrix} 1.0 \\ 0.0 \end{pmatrix} \begin{pmatrix} 0.9 \\ 0.1 \end{pmatrix} \begin{pmatrix} 0.84 \\ 0.16 \end{pmatrix} \begin{pmatrix} 0.804 \\ 0.196 \end{pmatrix} \cdots\cdots\cdots\cdots\cdots \begin{pmatrix} 0.75 \\ 0.25 \end{pmatrix}$$

$P(X_1)$. $P(X_2)$. $P(X_3)$. $P(X_4)$ $P(X_\infty)$

- $P(X_1=\text{Rain}) = 1$

$$\begin{pmatrix} 0.0 \\ 1.0 \end{pmatrix} \begin{pmatrix} 0.3 \\ 0.7 \end{pmatrix} \begin{pmatrix} 0.48 \\ 0.52 \end{pmatrix} \begin{pmatrix} 0.588 \\ 0.422 \end{pmatrix} \cdots\cdots\cdots\cdots\cdots \begin{pmatrix} 0.75 \\ 0.25 \end{pmatrix}$$

$P(X_1)$. $P(X_2)$. $P(X_3)$. $P(X_4)$ $P(X_\infty)$



Markov Models

Stationary distribution of P: $P_{\infty+1}(X) = P_{\infty}(X)$

- P(X_∞=sun) = (0.9 * P(X_∞=sun)) + (0.3 * P(X_∞=rain))
- P(X_∞=rain) = (0.1 * P(X_∞=sun)) + (0.7 * P(X_∞=rain))
- P(X_∞=rain) + P(X_∞=sun) = 1

⇒ P(X_∞=sun) = 3 P(X_∞=rain)

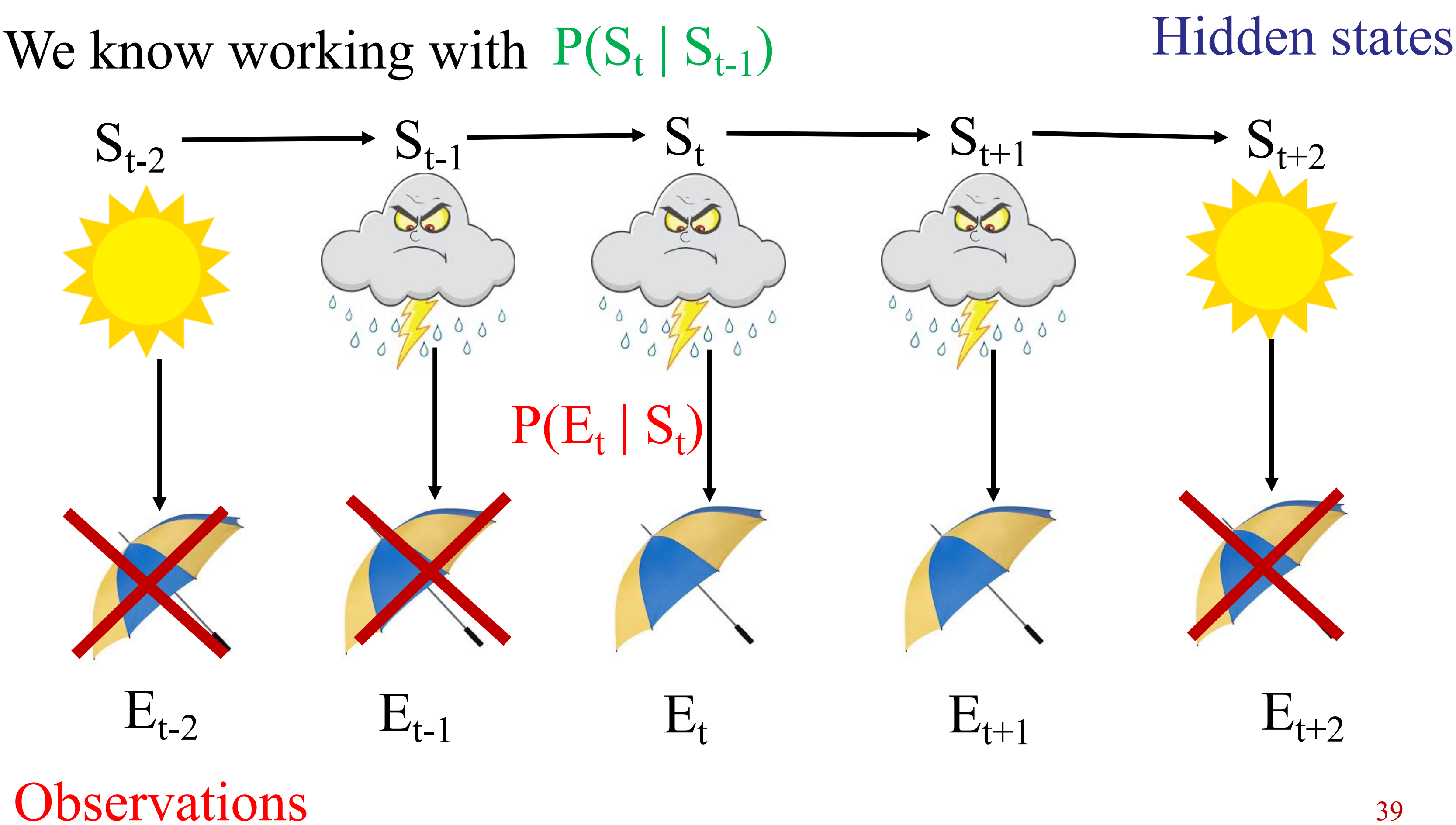
- From other initial distribution P(X_1)

$\begin{pmatrix} p \\ 1-p \end{pmatrix}$ .... $\begin{pmatrix} 0.75 \\ 0.25 \end{pmatrix}$

P(X_1) ....P(X_∞)

Hidden Markov Model

What is the probability of Sun on day t given umbrella = true?



Hidden Markov Model

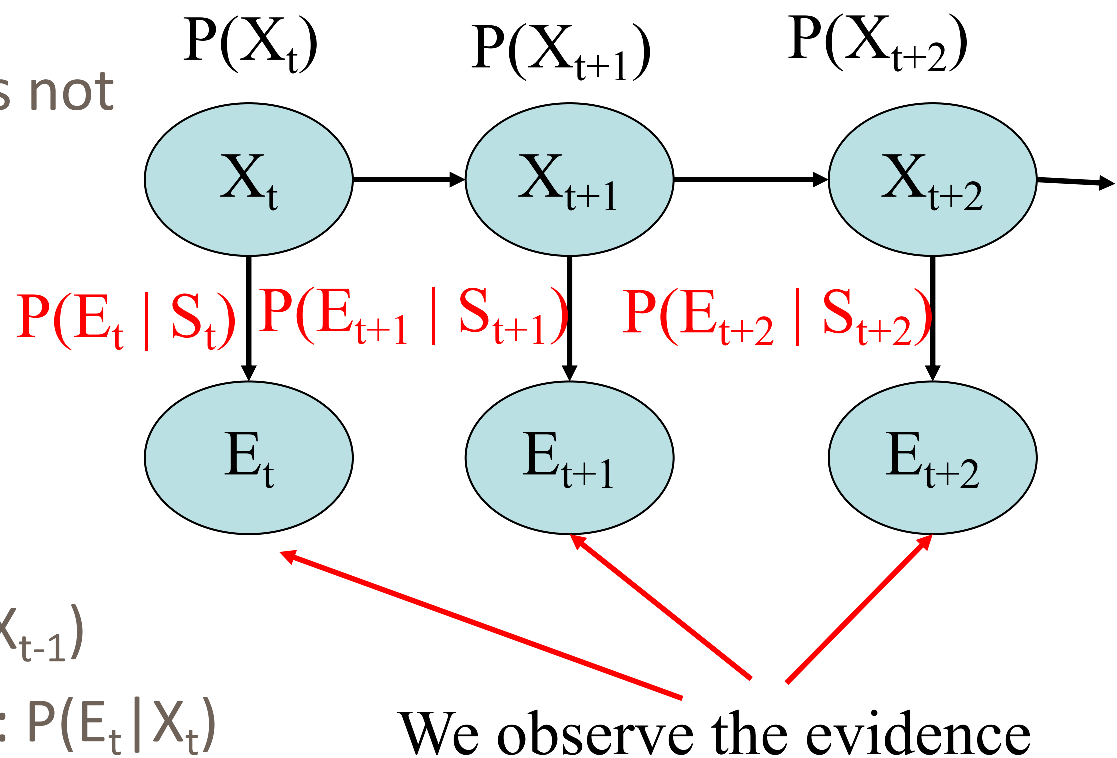
- Markov chain/process is a scholastic model describing a sequence of possible events where the probability of each events depends only on the state of the previous event.

- Hidden Markov Model is a probabilistic model where two coexistent stochastic processes: the process of moving between states and the process of emitting an output sequence, characterized by Markov property and the output independence.

Franzece and Luliano, 2019

Hidden Markov Model (HMM)

- Usually, the true state is not observed directly



HMM is defined by

- Initial distribution $P(X_0)$
- Transition model: $P(X_t | X_{t-1})$
- Emission/Sensor model: $P(E_t | X_t)$

Hidden Markov Model (HMM)

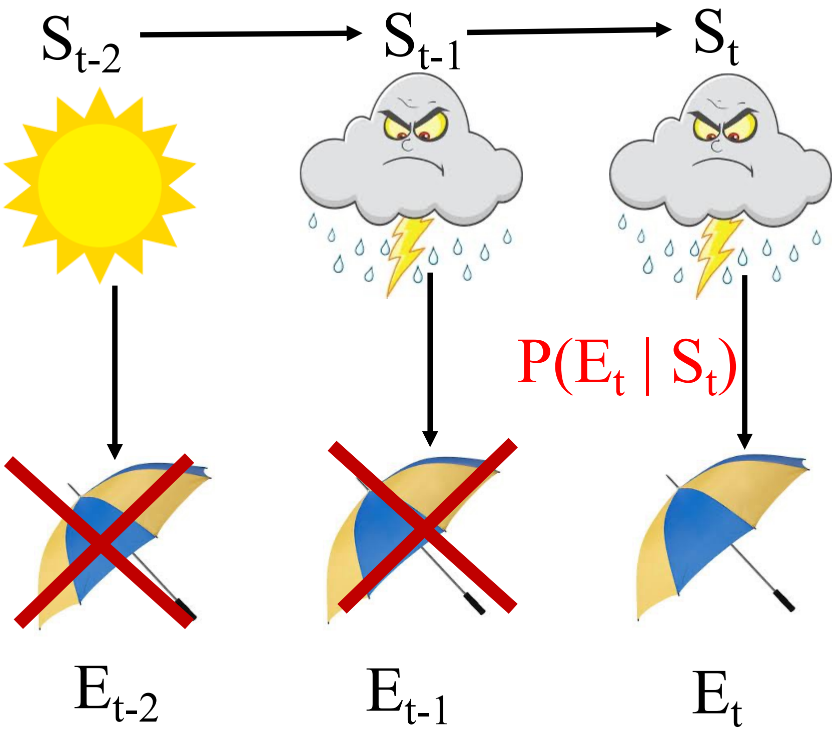
What the weather is like at time 1?

S_{t-1}	$P(S_t \mid S_{t-1})$	
	sun	rain
sun	0.9	0.1
rain	0.3	0.7

States {rain, sun}

E for Evidence (Our indicator/ signal of the weather)

S_t	$P(E_t \mid S_t)$	
	true	false
sun	0.2	0.8
rain	0.9	0.1



Hidden Markov Model (HMM)

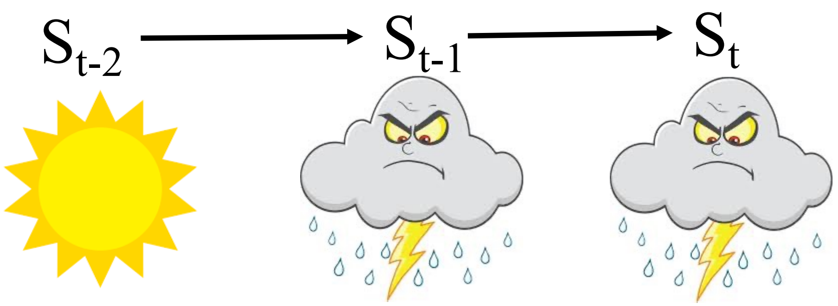
- Time 0 $P(S_0)=<0.5,0.5>$
- $P(X_t) = \sum P(X_t/X_{t-1}) * P(X_{t-1})$
- The weather like at time 1

  $P(X_1) = <0.9, 0.1> * 0.5$

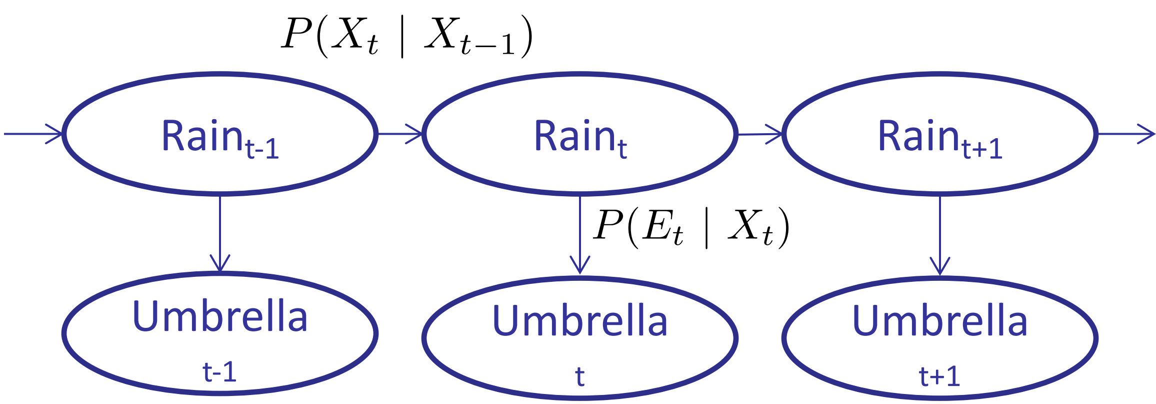
  $+ <0.3, 0.7> * 0.5$

  $P(X_1) = <0.6,0.4>$

S_{t-1}	$P(S_t \mid S_{t-1})$	
	sun	rain
sun	0.9	0.1
rain	0.3	0.7



Weather Example



An HMM is defined by:

- Initial distribution: $P(X_1)$
- Transitions: $P(X_t \mid X_{t-1})$
- Emissions: $P(E_t \mid X_t)$

The probability are different form the one on slide 41

X_t	X_{t+1}	$P(X_{t+1} \mid x_t)$
+r	+r	0.7
+r	-r	0.3
-r	+r	0.3
-r	-r	0.7

X_t	E_t	$P(E_t \mid X_t)$
+r	+u	0.9
+r	-u	0.1
-r	+u	0.2
-r	-u	0.8

Belief Updates

- Let B(X_t)= $P(x_t, e_{1:t})$ be believe at t

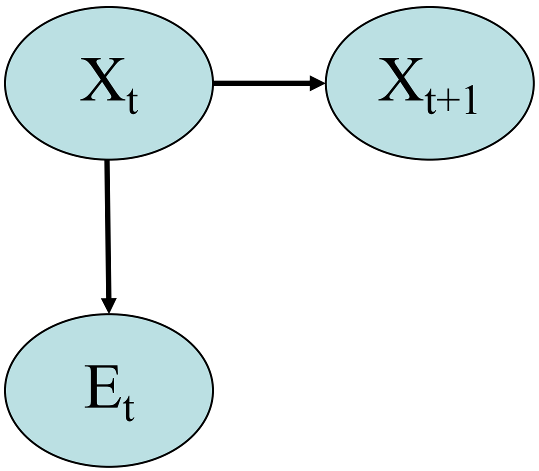
- $P(x_t, e_{1:t}) = \sum_{x_{t-1}} P(X_{t-1}, x_t , e_{1:t})$ Consider previous states

  = $\sum_{x_{t-1}} P(X_{t-1}, e_{1:t-1}) P(x_t|x_{t-1})$P($e_t|x_t$)

  =P($e_t|x_t$)$\sum_{x_{t-1}} P(x_t|x_{t-1})$ P($x_{t-1}, e_{1:t-1}$)

$\Rightarrow$ B'(X_t) = $\sum_{x_{t-1}} P(X_t|x_{t-1})$ B(X_{t-1})

$\Rightarrow$ B(X_t)= P($e_t|x_t$) B'(X_t)



Weather Example

$P(S_0)$ =<0.5 0.5> - We do not know

$P(s_1|e_1= +u)=\alpha\, P(E_1|S_1)P(S_0)$

$= \alpha<0.9,0.2> * <0.5, 0.5>$

$= \alpha<0.45, 0.1>$

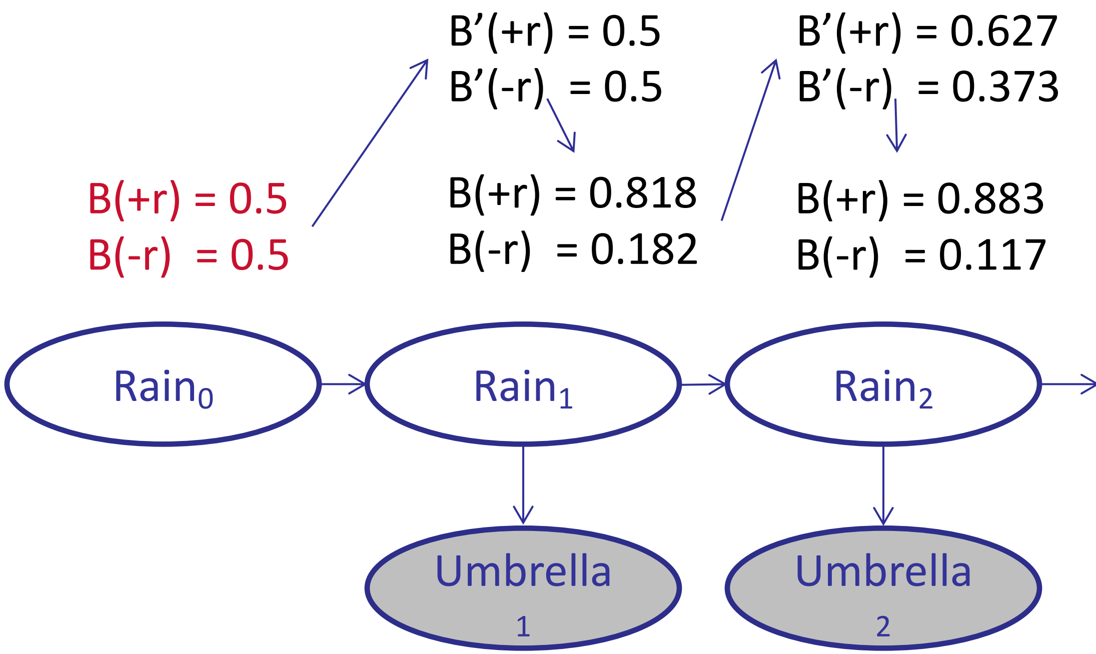
$=<0.818, 0.182>$

α is for normalization = sum of the probabilities is 1

X_t	E_t	$P(E_t\|X_t)$
+r	+u	0.9
+r	-u	0.1
-r	+u	0.2
-r	-u	0.8

X_t	X_{t+1}	$P(X_{t+1}\|X_t)$
+r	+r	0.7
+r	-r	0.3
-r	+r	0.3
-r	-r	0.7

Weather Example



X_t	X_{t+1}	$P(X_{t+1}\|X_t)$
+r	+r	0.7
+r	-r	0.3
-r	+r	0.3
-r	-r	0.7

X_t	E_t	$P(E_t\|X_t)$
+r	+u	0.9
+r	-u	0.1
-r	+u	0.2
-r	-u	0.8

$\Rightarrow B'(X_t) = \sum_{x_t} P(X_t|x_{t-1})\, B(X_{t-1})$

$\Rightarrow B(X_t) = P(e_t|x_t)\, B'(X_t)$

Weather Example

B'(+r) = 0.5
B'(-r) = 0.5

B'(+r) = 0.627
B'(-r) = 0.373

0.818 * <0.7, 0.3> = 0.572+0.054 = 0.627

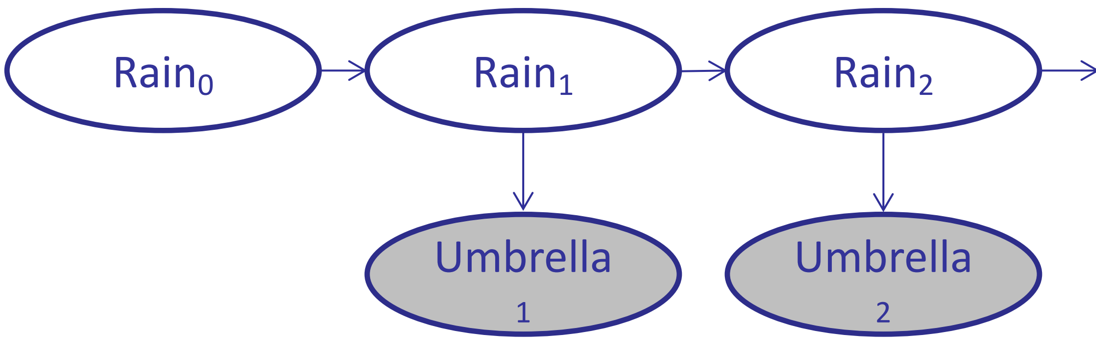
0.182 * <0.3, 0.7> = 0.245+0.127 = 0.373

B(+r) = 0.5
B(-r) = 0.5

B(+r) = 0.818
B(-r) = 0.182

B(+r) = 0.883
B(-r) = 0.117

0.9 * 0.627 = 0.564 α 0.883
0.2 * 0.373 = 0.074 α 0.117



X_t	X_{t+1}	$P(X_{t+1}\|X_t)$
+r	+r	0.7
+r	-r	0.3
-r	+r	0.3
-r	-r	0.7

X_t	E_t	$P(E_t\|X_t)$
+r	+u	0.9
+r	-u	0.1
-r	+u	0.2
-r	-u	0.8

$$\Rightarrow B'(X_t) = \sum_{x_t} P(X_{t+1}|x_t)\, B(X_{t-1})$$

$$\Rightarrow B(X_t) = P(e_t|x_t)\, B'(X_t)$$

Clustering HMM

- The state-observations probability matrices would be of *k* clusters, say 2 HMM $\lambda_1 \, and \, \lambda_2$
- $\lambda_i = (u^i, A^i, B^i)$ // u is for initial state, *A* is the state probability matrix and *B* is for state-observation probability matrix

Distance $\quad d(\lambda_1, \lambda_2) \triangleq \left\| B^{(1)} - B^{(2)} \right\|$

$$\triangleq \left\{ \frac{1}{MN} \sum_{j=1}^{N} \sum_{k=1}^{M} \left[b_{jk}^{(1)} - b_{p(j)k}^{(2)} \right]^2 \right\}^{1/2}$$

Levinson et a1. 1983

Clustering HMM

A classic method to measure dissimilarity of probability distributions is Kullbak-Leiber divergence:

$$D_{KL}(P||Q) = \sum_{x \in X} P(x) log \left(\frac{P(x)}{Q(x)} \right)$$

The points could be clustered using portioning algorithms that minimize the average dissimilarity inside the clusters.

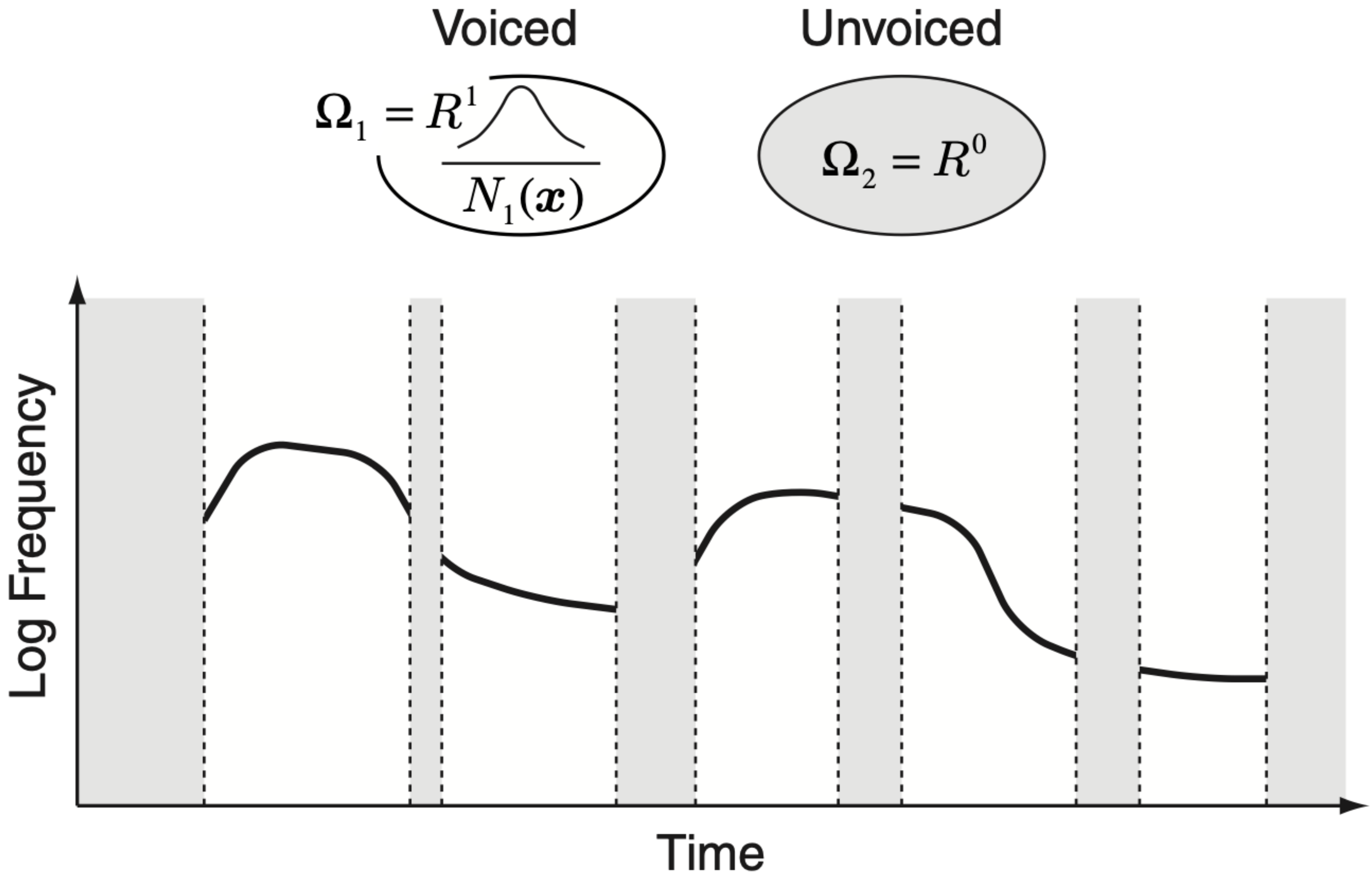
HMM Clustering in Python

```
import hmmlearn.hmm as hmm

gHmm = hmm.GaussianHMM(n_components, n_iter)
model = gHmm.fit(Data)
hidden_states = model.predict(Data)
```

model.transmat_ : Matrix of transition probabilities between states.

HMM Clustering for Voice Analysis



https://wiki.inf.ed.ac.uk/pub/CSTR/Trajectory
Modelling/HTS-Introduction.pdf

Hand-writing Recognition with HMM

Figure 1: Example observation from data set: actual word is “commanding” with the first letter removed

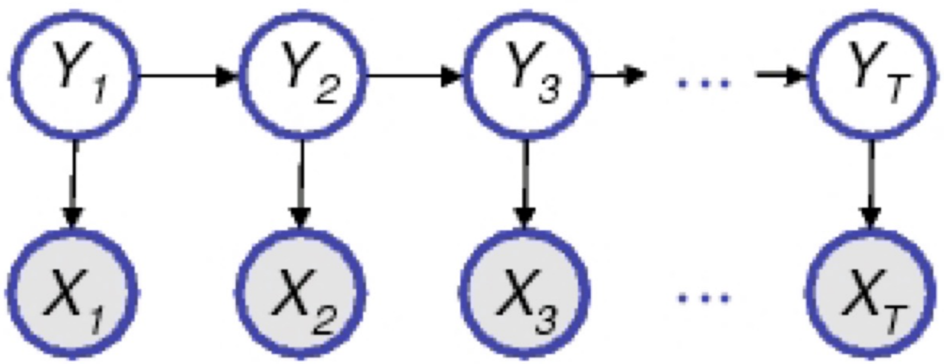


Figure 2: In our Markov model, the hidden variables *Yt* are the 26 letters of the English alphabet and the observed variables are the bitmap images

Conclusions

- Unsupervised machine learning is about analyzing and clustering datasets.
    - The uses are clustering, dimensionality reduction, and association.

- Clustering algorithms use distance metrics to partition the data such that similar items are grouped together.

Thank you

Any Question?