Machine Learning and Its applications
(and beyond)

Hisashi Kashima
Department of Mathematical Informatics
The University of Tokyo

東京大学
THE UNIVERSITY OF TOKYO

What is Machine Learning?

Machine learning as a buzzword:
Increasing popularity of machine learning

- After the IT revolution, “how to exploit data” is increasingly more important than “how to store data”
  - Companies are trying to position data analytics as foundations of competitiveness
- “Machine learning” as a buzzword
  - The rise of “big data” (another buzzword)
  - Data scientist is “the sexiest job in the 21st century”

What is machine learning?：
Machine learning is a data analytics

- Originally a branch of artificial intelligence
  - Computer programs that “learns” from experience
  - Based on logical inference
- Rise of “statistical” machine learning
  - Successes in bioinformatics, natural language processing, and other business areas
  - Victory of IBM’s Watson QA system

What can machine learning do?: Prediction and discovery

- Two categories of the use of machine learning:

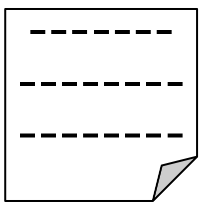
1. Prediction (supervised learning)
   - "What will happen in future data?"
   - Given past data, predict about future data
2. Discovery (unsupervised learning)
   - "What is happening in data in hand?"
   - Given past data, find insights in them

Formulations of machine learning problems: Supervised and unsupervised

- Learning system as a function f: $\boldsymbol{x} \rightarrow y$
  - $\boldsymbol{x} \in \Re^D$: input as real vector
  - $y \in \{1, 2, \ldots, C\}$: discrete output
- We want f $\Rightarrow$ Learn from data
- Two machine learning problems:
  1. Supervised learning: have access to input and output pairs
     - $\{(\boldsymbol{x}^{(1)}, y^{(1)}), (\boldsymbol{x}^{(2)}, y^{(2)}), \ldots, (\boldsymbol{x}^{(N)}, y^{(N)})\}$: N in/output pairs
  2. Unsupervised learning: have access only to input data
     - $\{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(N)}\}$: N inputs

An application of supervised learning:
Sentiment analysis

- Classify the polarity of a given text
  - $y \in \{+, -\}$:Whether or not a blog post $\boldsymbol{x}$ favors a product
- $\boldsymbol{x}$ is defined by using words appearing in the text



$$\boldsymbol{x} := \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{pmatrix}$$

of "good"

of "simple"

⋮

of "slow"

bag-of-words representation

- Note: design of the feature vector is left to uses

Various applications of machine learning:
From online shopping to system monitoring

- Marketing
  - Product recommendation
  - Sentiment analysis on Web
  - Advertisement optimization
- Finance
  - Credit risk management
  - Fraud detection
- Bio/healthcare
  - Medical diagnosis
  - Gene recognition
- Web
  - Web search
  - Spam detection
  - SNS
- Multimedia
  - Voice recognition
  - Face/object recognition
- System monitoring
  - fault diagnosis
  - Intrusion detection

Recommender Systems

Recommender systems:
Personalized information filter

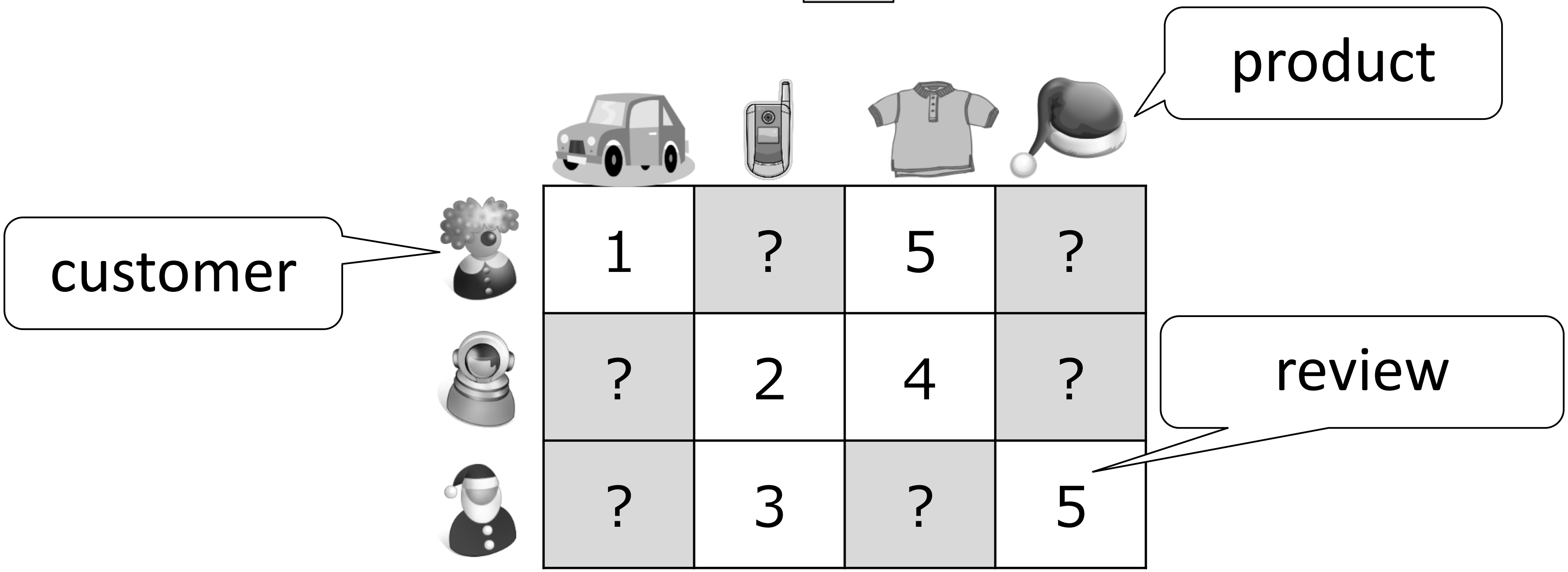
- Amazon offers a list of products I am likely to buy (based on my purchase history)

Ubiquitous recommender systems:
Recommender systems are present everywhere

- A major battlefield of machine learning algorithms
  - Netflix challenge (with $100 million prize)
- Recommender systems are present everywhere:
  - Product recommendation in online shopping stores
  - Friend recommendation on SNSs
  - Information recommendation (news, music, …)
  - …

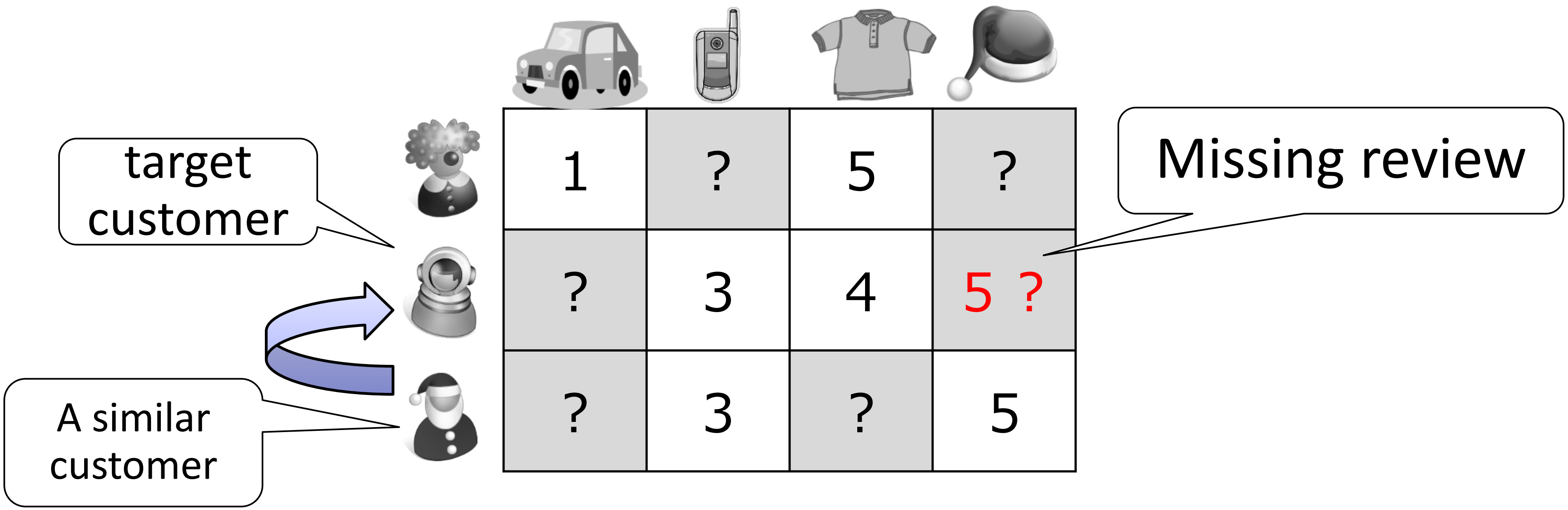
A formulation of recommendation problem:
Matrix completion

- A matrix with rows (customers) and columns (products)
  - Each element = review score
- Given observed parts of the matrix, predict the unknown parts (?)



Basic idea of recommendation algorithms:
“Find people like you”

- GroupLens: an earliest algorithm (for news recommendation)
  - Inherited by MovieLens (for Movie recommendation)
- Find people similar to the target customer, and predict missing reviews with theirs

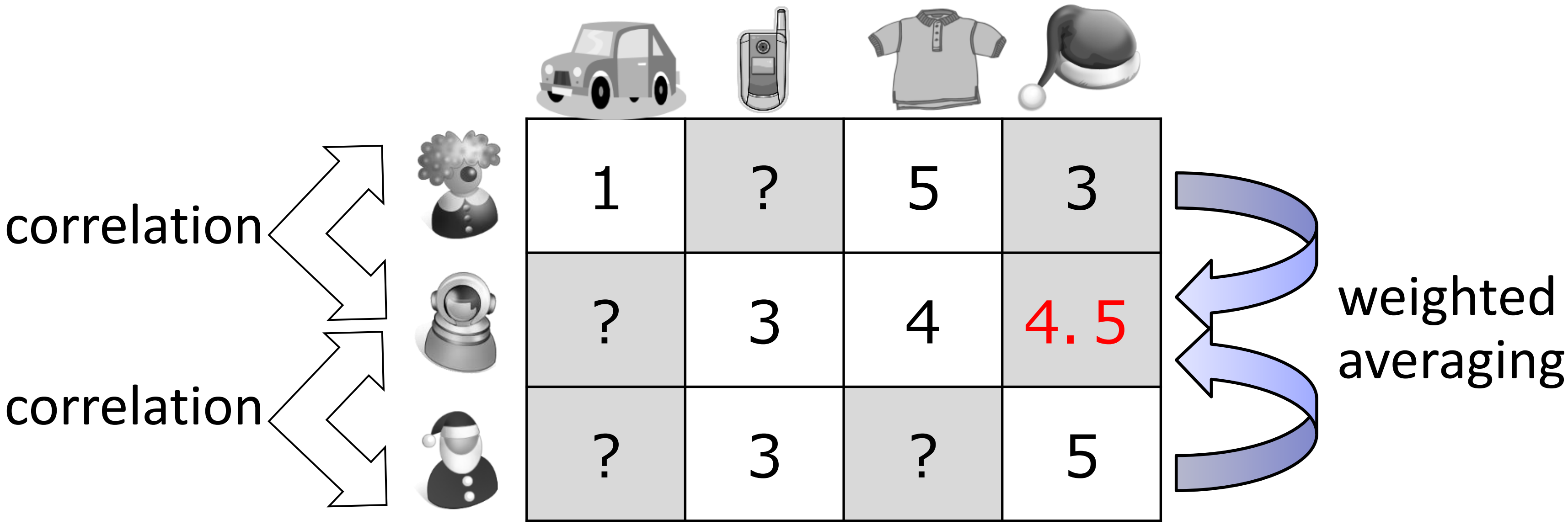


GroupLens:
Weighted prediction with correlations among customers

- Define customer similarity by correlation (of observed parts)
- Make prediction by weighted averaging with correlations :

$$y_{i,j} = y_i + \Sigma_{k\neq i}\, \rho_{i,k} (y_{k,j} - y_k) \,/\, \Sigma_{k\neq i}\, \rho_{i,k}$$

correlation



1	?	5	3
?	3	4	4. 5
?	3	?	5

Low-rank assumption for matrix completion:
GroupLens implicitly assumes low-rank matrices

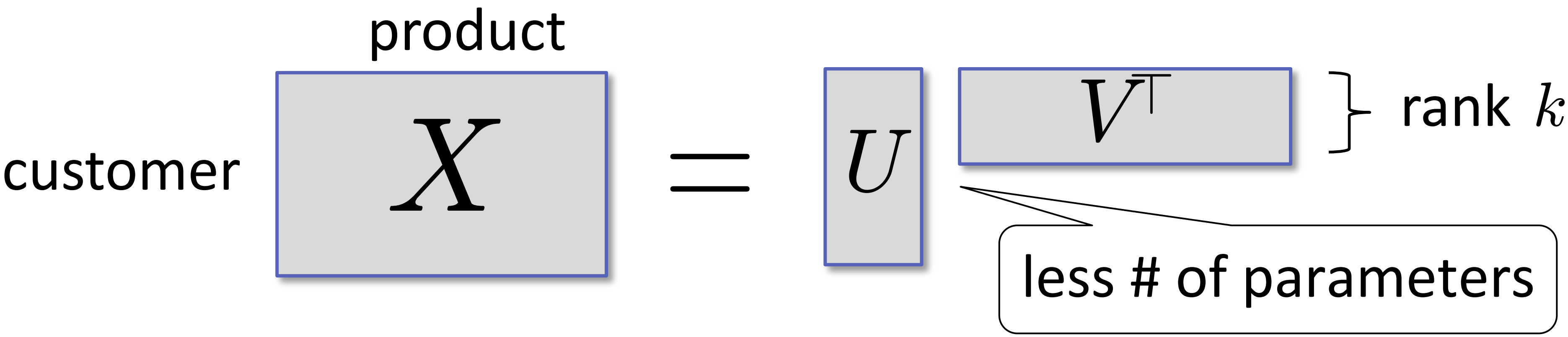
- Assumption of GroupLens algorithm:
  Each row is represented by a linear combination of the other rows (i.e. linearly dependent)

⇒ The matrix is not full-rank （≒ low-rank）

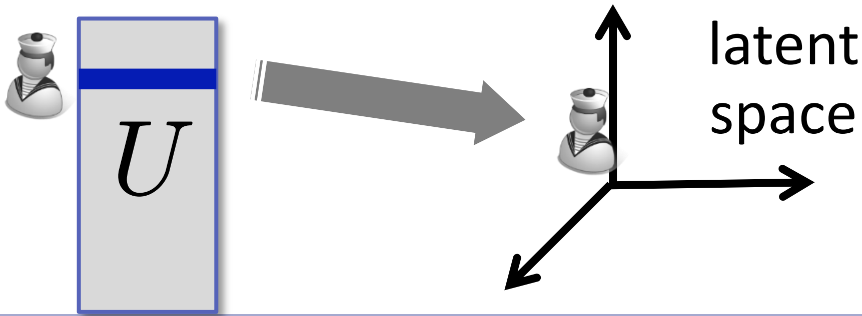
- Low-rank assumption helps matrix completion

Low-rank matrix factorization:
Projection onto low-dimensional latent space

- Low-rank matrix: product of two (thin) matrices



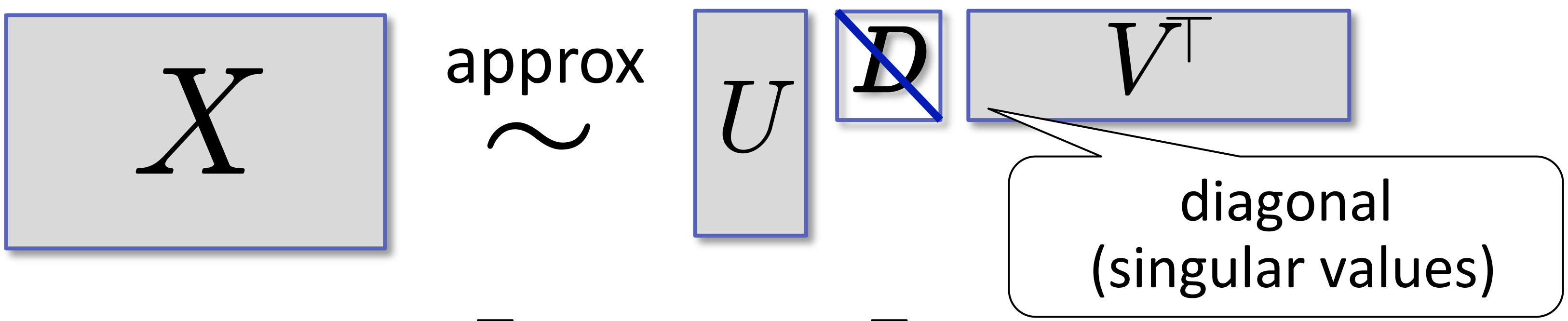
- Each row of U and V is an embedding of each customer (or product) onto low-dimensional latent space



Example of low-rank matrix decomposition:
Singular value decomposition

- Find a best low-rank approximation of a given matrix

$$\underset{\boldsymbol{Y}}{\text{minimize}} \ \|\boldsymbol{X} - \boldsymbol{Y}\|_F^2 \ \text{ s.t. } \operatorname{rank}(\boldsymbol{Y}) \leq k$$

- Singular value decomposition (SVD)
  - 

    wrt constraint: $U^\top U = I \ \ V^\top V = I$
  - The largest k eigenvalues of $X^\top X$ best approximate

Strategies for matrices with missing values:
EM algorithm, gradient descent, and trace norm

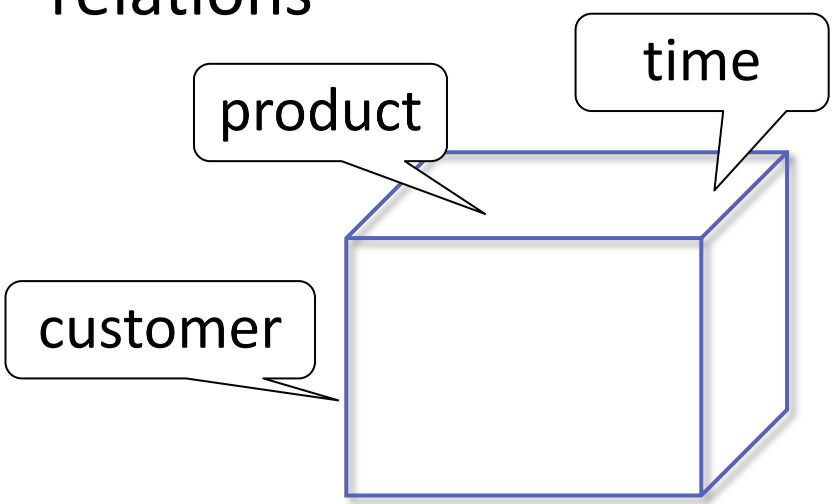
- SVD is not applicable to matrices with missing values
- For completion problem:
  - Direct application of SVD to a (somehow) filled matrix
  - Iterative applications: iterations of completion and decomposition
- For large scale data:
  Gradient descent using only observed parts
- Convex formulation: Trace norm constraint

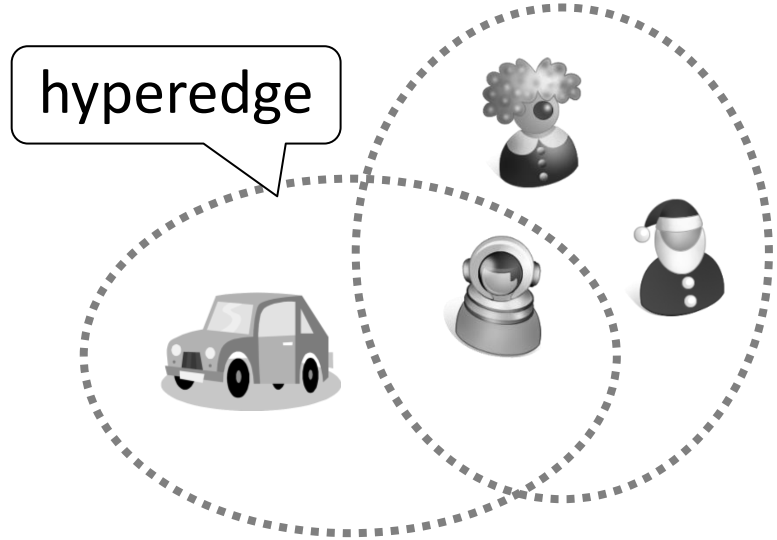
Predicting more complex relations: Multinomial relations

- Matrices can represent only one kind of relations
  - Various kinds of relations (actions):
    Review scores, purchases, browsing product information, …
  - Correlations among actions might help
- Multinomial relations:
  - (customer, product, action)-relation:
    (Alice, iPad, buy) represents “Alice bought an iPad.”
  - (customer, product, time)-relation:
    (John, iPad, July 12^{th}) represents “John bought an iPad on July 12th.”

Multi-dimensional array:
Representation of multinomial relations

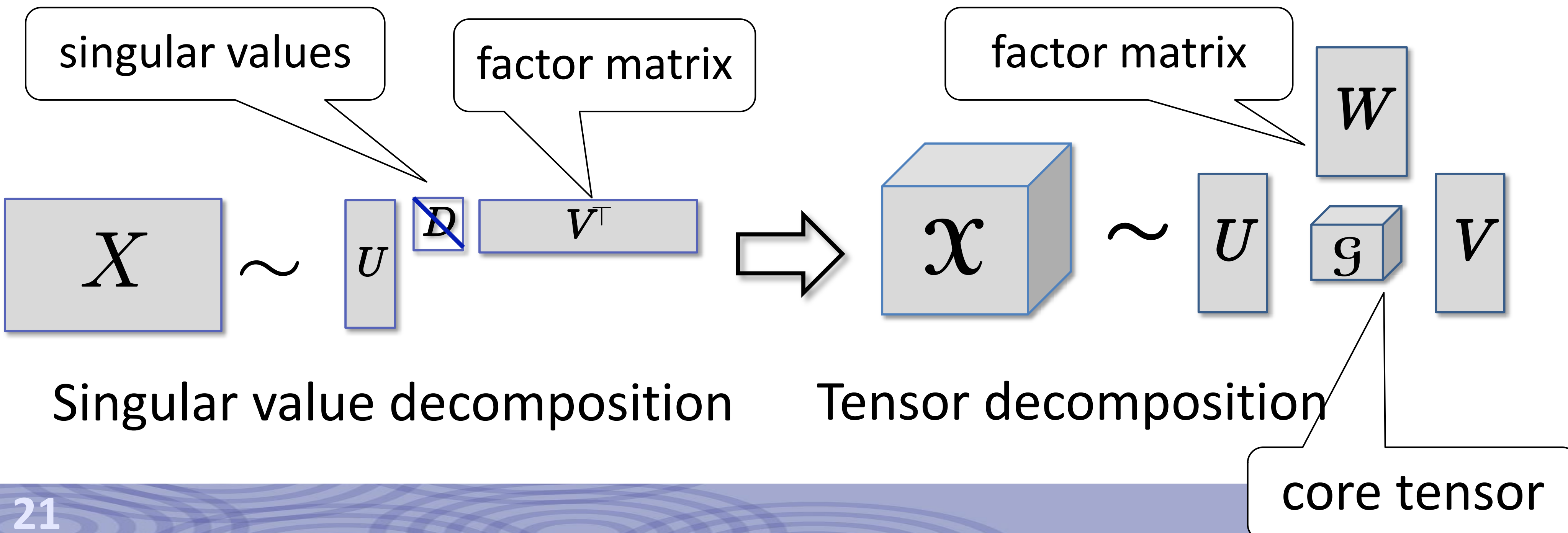
- Multidimensional array: Representation of complex relations among multiple object
  - Types of relations (actions, time, conditions, …)
  - Relations among more than two objects
- Hypergraph: allows variable number of objects involved in relations





Tensor decomposition:
Generalization of low-rank matrix decomposition

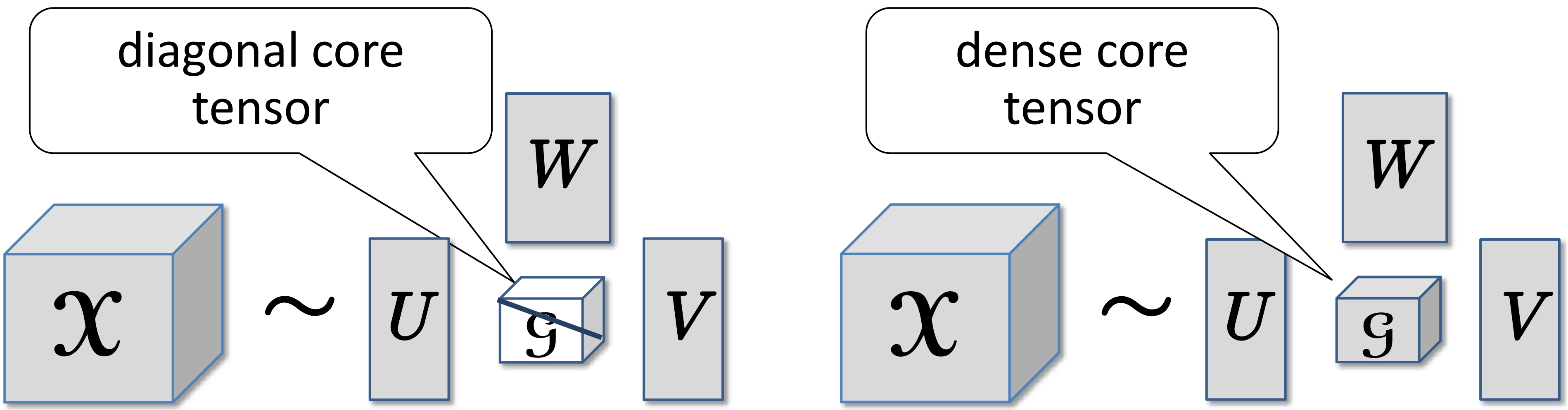
- Generalization of matrix decomposition to multidimensional arrays
  - A small core tensor and multiple factor matrices
- Increasingly popular in machine learning/data mining



Singular value decomposition   Tensor decomposition

Tensor decompositions:
CP decomposition and Tucker decomposition

- CP decomposition: A natural extension of SVD (with a diagonal core)
- Tucker decomposition: A more compact model (with a dense core)



CP decomposition

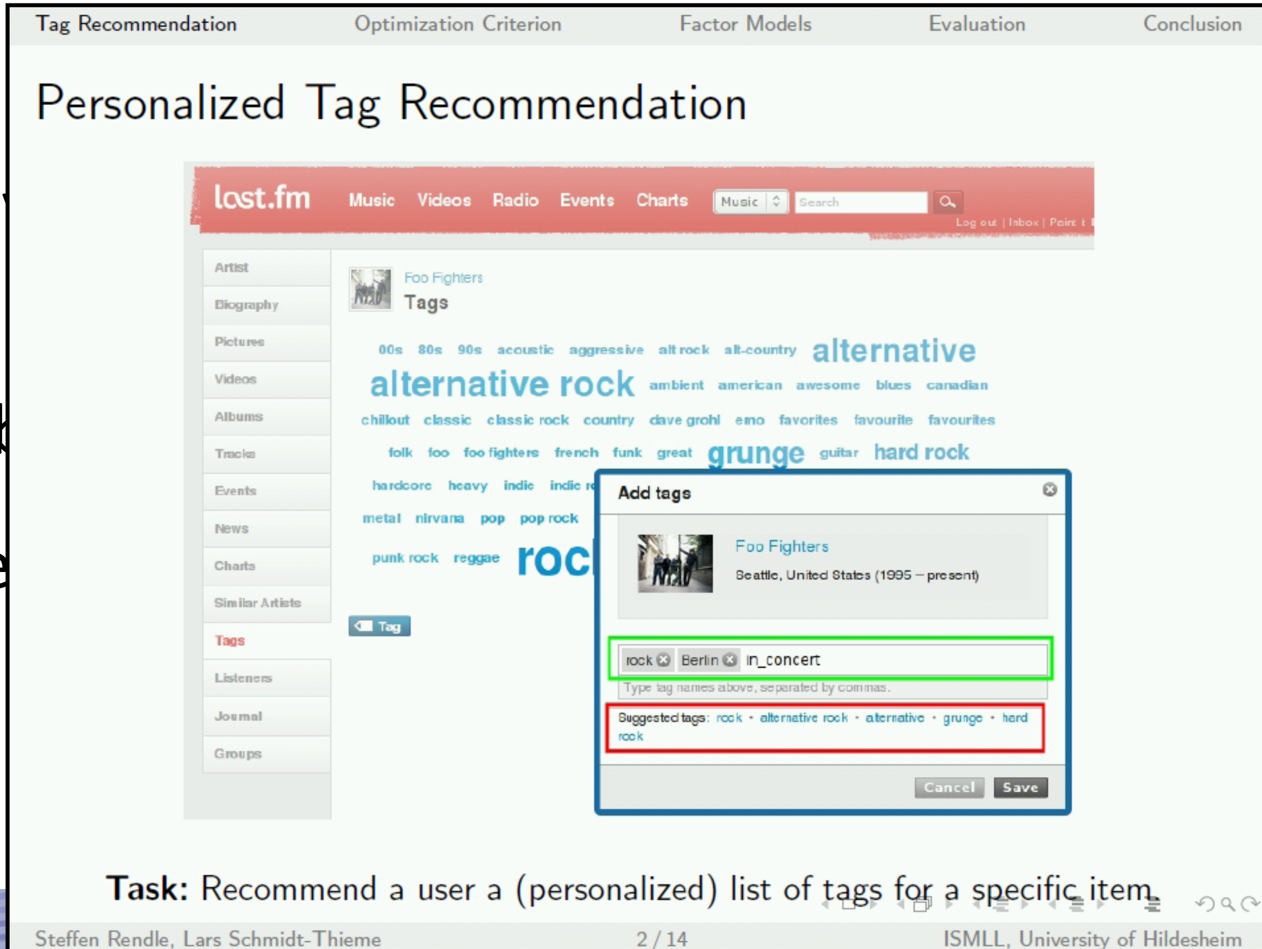
Tucker decomposition

Applications of tensor decomposition:
Tag recommendation, social network analysis, …

- Personalized tag recommendation (user×webpage×tag)
  - predicts tags a user gives a webpage
- Social network analysis (user×user×time)
  - analyzes time-variant relationships
- Web link analysis
  (webpage×webpage×anchor text)
- Image analysis　(image×person×angle×light×…)

Applications of tensor decomposition:
Tag recommendation, social network analysis, …

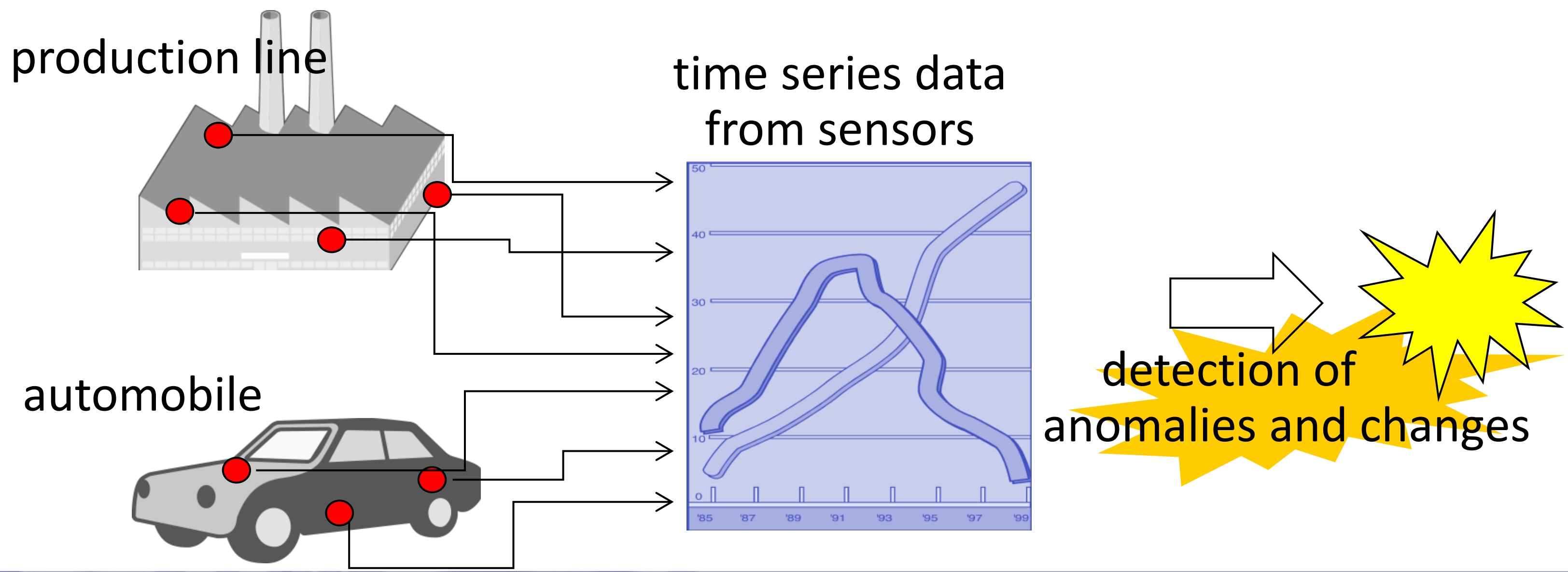
- Personalized tag recommendation (user×webpage×tag)
  - predicts tags a user gives a webpage
- Social
  - anal
- Web l
  (web
- Image …)



Anomaly Detection

Anomaly detection:
Early warning for system failures reduces costs

- A failure of a large system can cause a huge loss
  - production line in factory
  - Infection of computer virus/intrusion to computer systems
- Early detection of failures from data collected from sensors

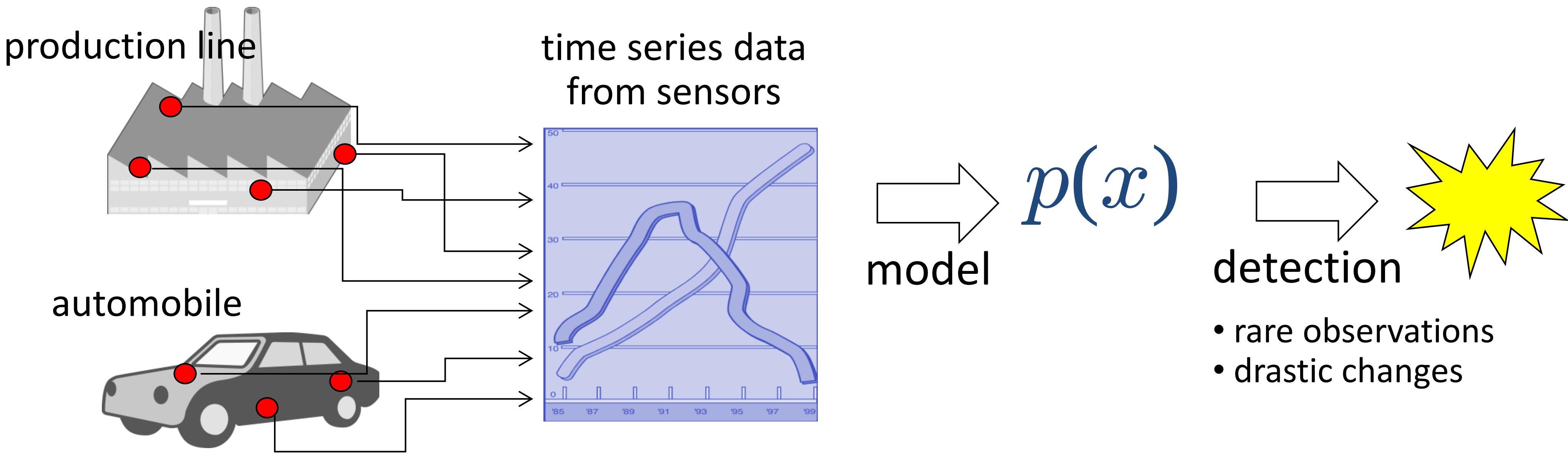


Difficulty in anomaly detection: Failures are often unknown

- Known failures are detected by using supervised learning:
  1. Construct a predictive model from past failure data
  2. Apply the model to system monitoring
- Serious failures are often new ones
  → No past data are available
- There are many cases where supervised learning is not applicable

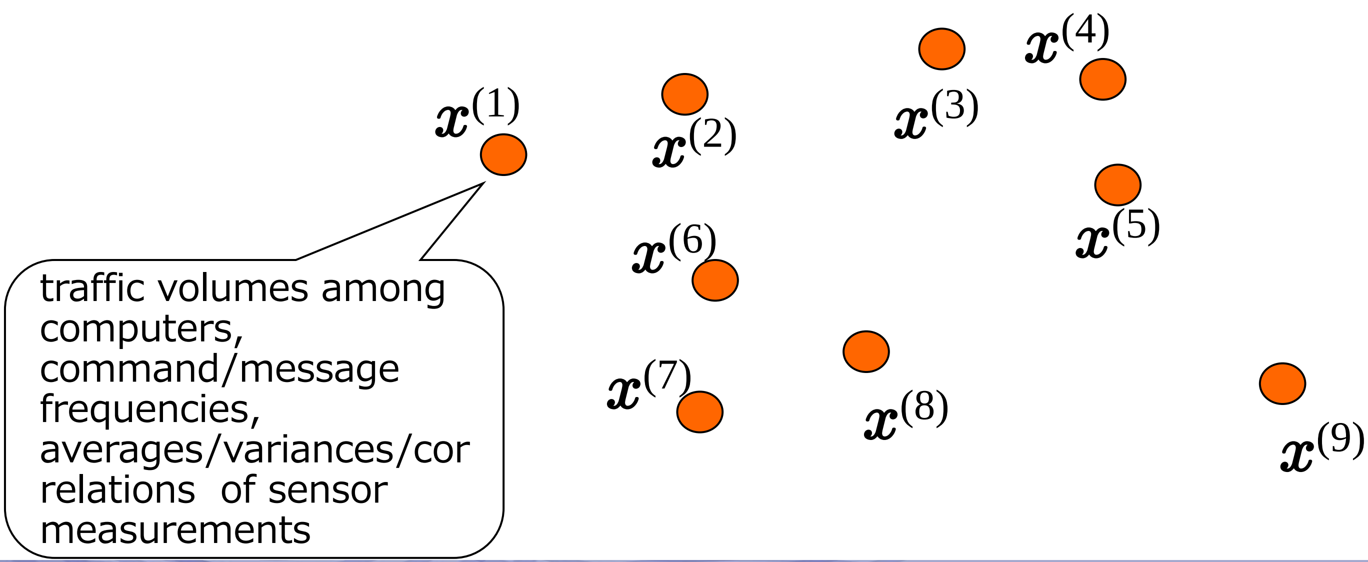
Change the strategy:
Model the normal times, detect deviation from them

- Difficult to model anomalies → Model normal times
  - Data at normal times are abundant
- Observation of rare data is a precursor of failures



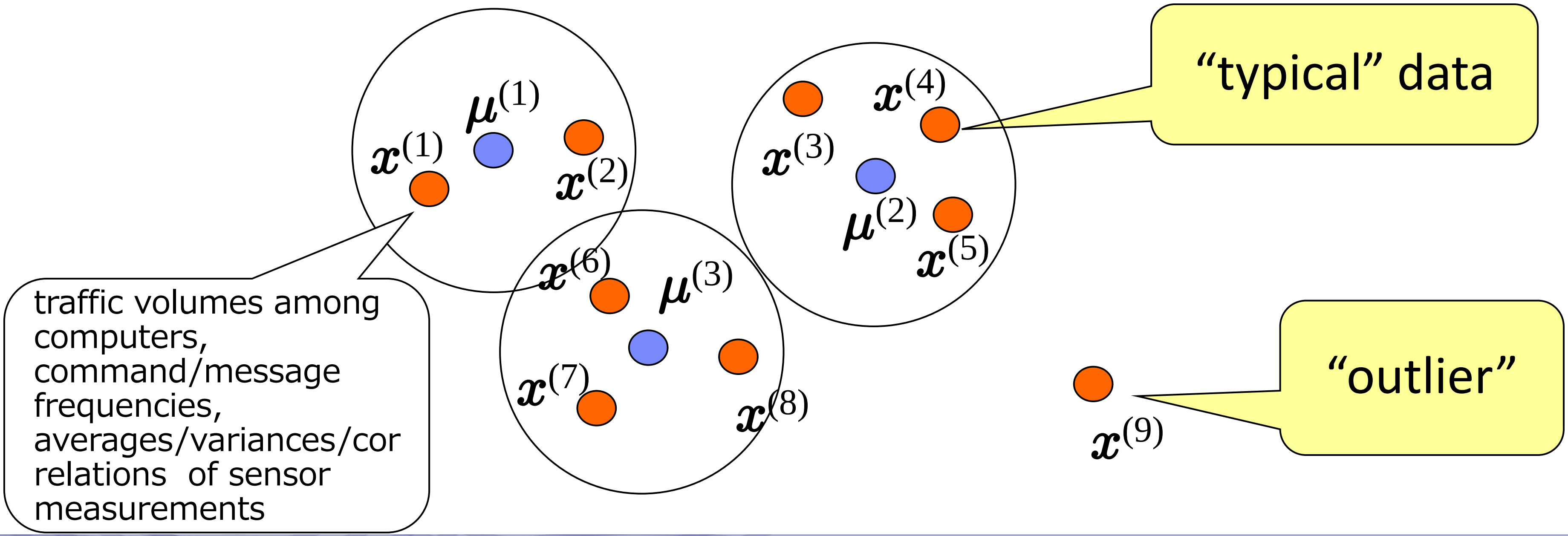
Clustering:
Model the normal times by grouping the data

- Divide normal time data $\{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(N)}\}$ into K groups
  - Group is represented by centers $\{\boldsymbol{\mu}^{(1)}, \boldsymbol{\mu}^{(2)}, \ldots, \boldsymbol{\mu}^{(K)}\}$
- data $\boldsymbol{x}$ is an "outlier" if it lies far from all of the centers
  = system failures, illegal operations, instrument faults



Clustering:
Model the normal times by grouping the data

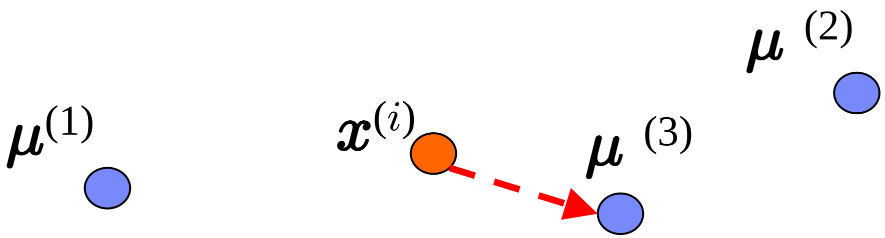
- Divide normal time data $\{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(N)}\}$ into K groups
  - Group is represented by centers $\{\boldsymbol{\mu}^{(1)}, \boldsymbol{\mu}^{(2)}, \ldots, \boldsymbol{\mu}^{(K)}\}$
- data $\boldsymbol{x}$ is an "outlier" if it lies far from all of the centers
  ＝system failures, illegal operations, instrument faults



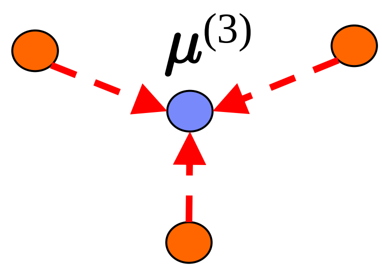
K-means algorithm:
Iterative refinements of groups

- Repeat until convergence:

1. Assign each data $\boldsymbol{x}^{(i)}$ to its nearest center $\boldsymbol{\mu}^{(k)}$



2. Update each center to the center of the assigned data

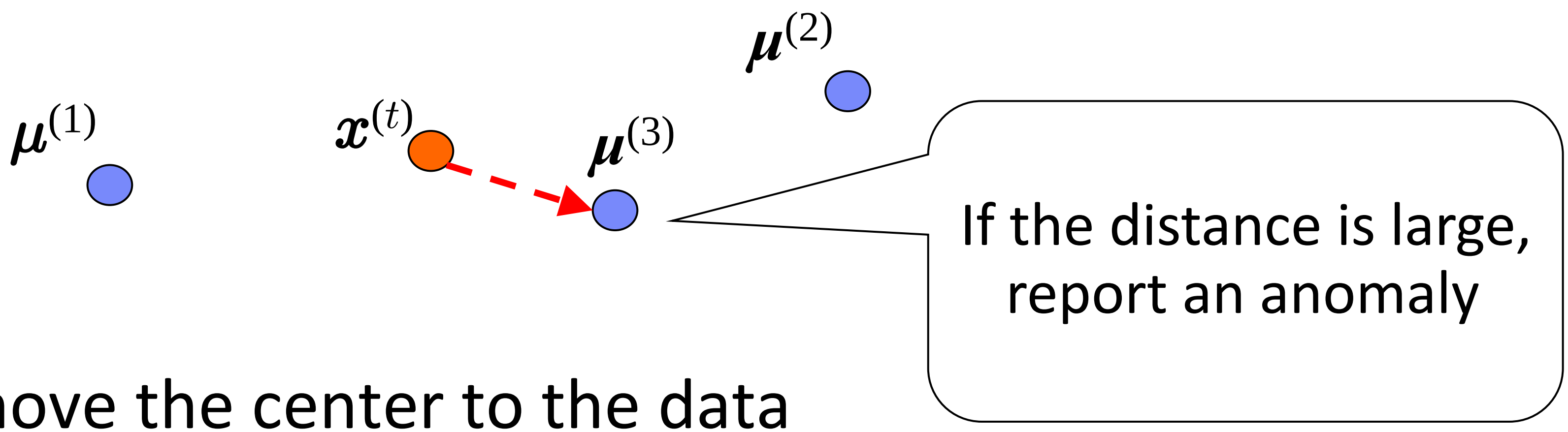


Sequential K-means:

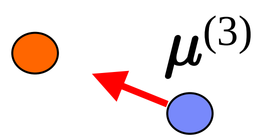
Simultaneous estimation of clusters and outliers

- Data arrives in a streaming manner, and apply clustering and anomaly detection at the same time

1. Assign each data $\boldsymbol{x}^{(t)}$ to its nearest center $\boldsymbol{\mu}^{(k)}$



2. Slightly move the center to the data



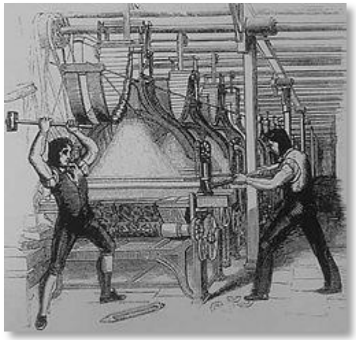
Limitation of anomaly detection: Failures are unknown

- Anomalies are not defined in advance
  → Details of failures are unknown
- Report to system administrators
  → Investigations by administrators
- Change detection: detects model changes

Future Direction

Can machines with big data go beyond humans?: Many instances. But, not always

- Computers overwhelming humans on many “intelligent” tasks:
  - e.g. the victory of IBM’s QA system WATSON trained with big data
- Concerns about replacement of humans by computers:
  - Luddite movement in the industrial revolution in 19th century



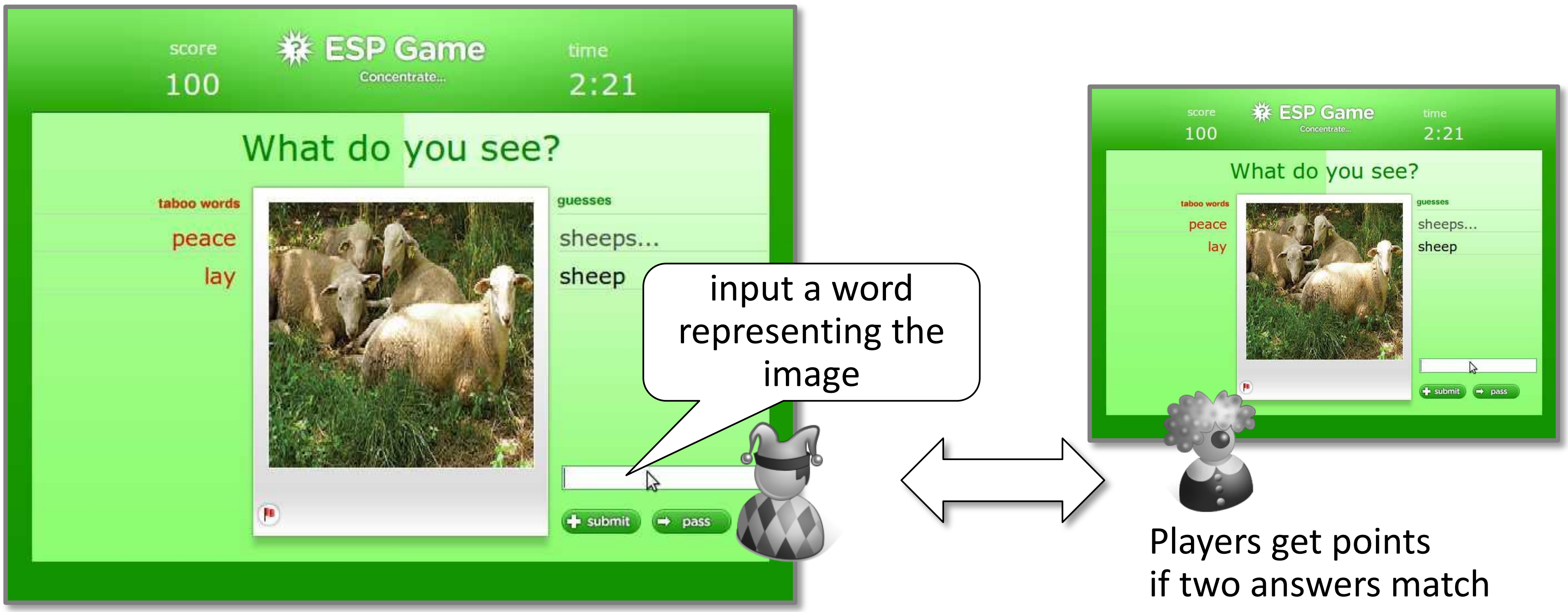
- With big data, do computer always defeat humans?
  - No, there are still many only human can do

ReCAPTCHA：
Authentication with a “hidden” task

- A Turing test discriminating humans from computers
  - Authentication of an access by human to a Web site requires users to read and input given two words
  - Hard for computers, but easy for humans

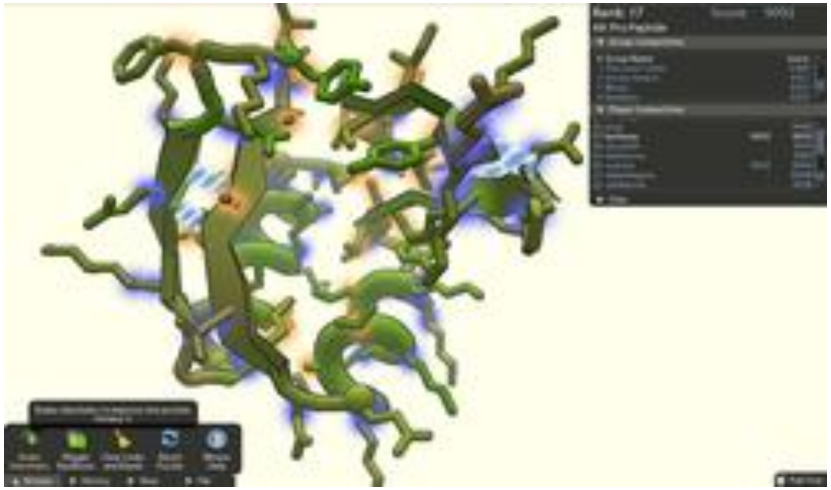
ESP game:
A game with a “hidden” task

- A cooperative game played by two online players
- Players get points if two keywords given by them to a presented image coincide



“Hidden” tasks:
Tasks hard for computers, but easy for humans

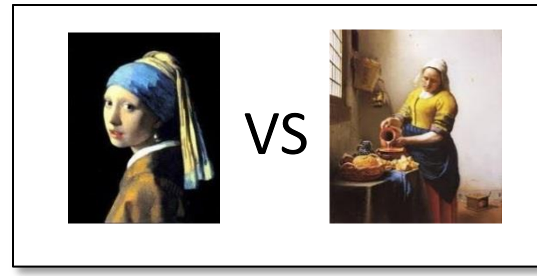
- ReCAPCHA: digitization of paper documents
  - The system does not know one of two presented words
  - Human helps digitization of paper documents
- ESP game: image tagging
  - Tags for images improves image search
  - Human helps tagging of images
  - “Game with a purpose (GWAP)”
    - Music tagging, protein folding, …



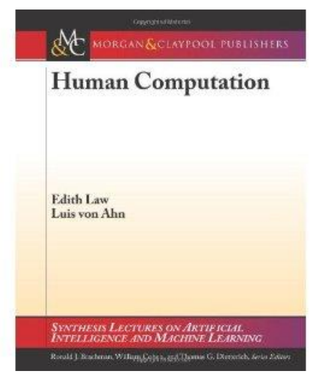
- Both examples cleverly embed computer-unfriendly tasks into other forms

Human computation:
Cooperative Problem solving by machines and humans

- Human computation:
  - regards humans as computational resources
  - solves problems that computers can not solve
- Efficient cooperative problem solving
  - E.g. quicksort with humans



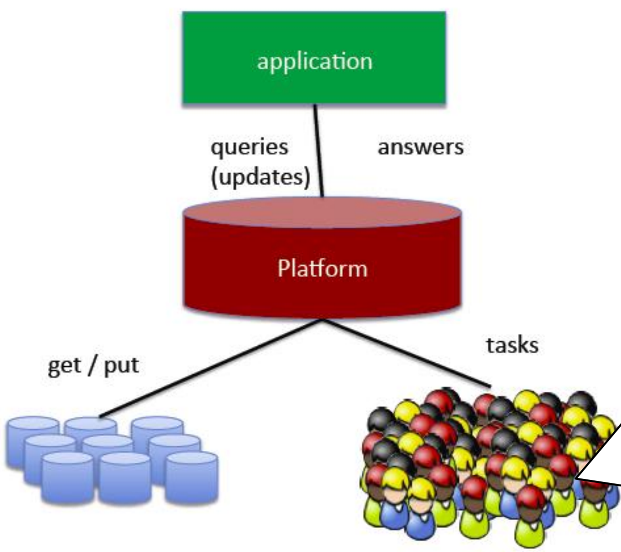
- Employment of human resources:
  - Gamification: Tasks implicitly embedded into games
  - Crowdsourcing (e.g. Mechanical Turk)
  - …



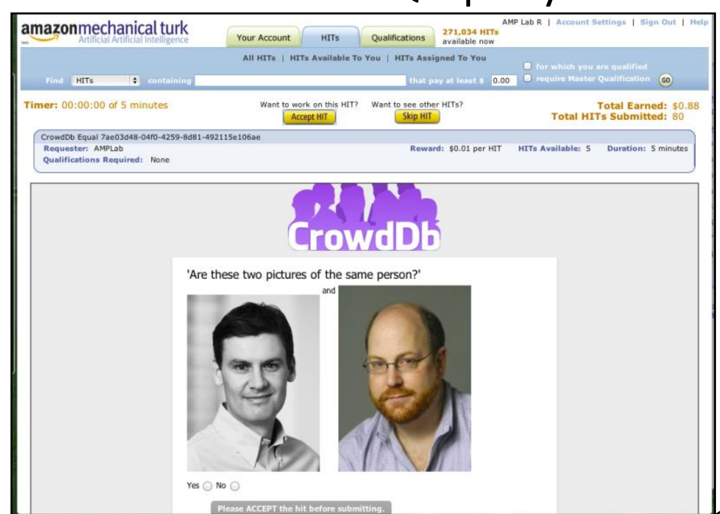
Law & Von Ahn (2012), Human Computation

Human computation in computer science:
NLP, computer vision, HCI, web search, DB, …

- Natural language processing: NL understanding, translation, …
- Computer vision: image understanding, annotation, detection,...
- DB/IR: data generation/integration, evaluation, …
- Machine learning: data collection



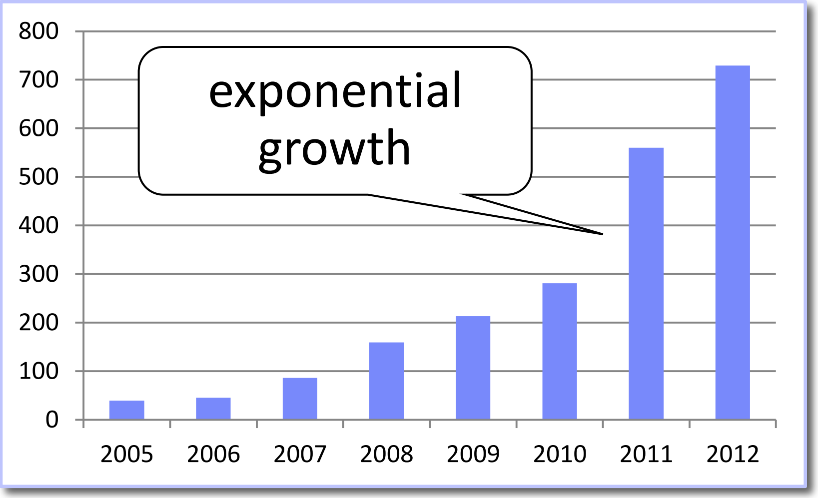
Call crowdsourcing services to evaluate a SQL query



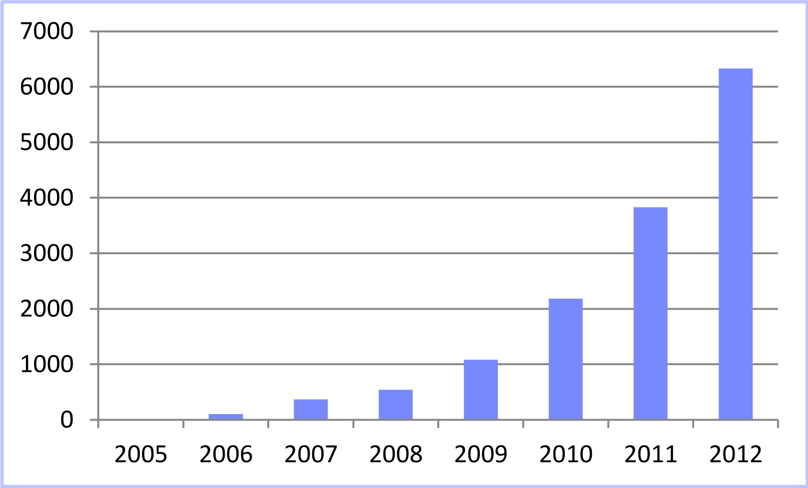
Research trend:
Exponential growth of human computation/crowdsourcing

- Exponential growth of related research

#papers related to "human computation"



#papers related to "crowdsourcing"



※ According to Google Scholar

- 2005: Amazon Mechanical Turk
- 2006: "human computation" "crowdsourcing"
- 2013: The first international conference on human computation and crowdsourcing HCOMP (Human Computation & Crowdsourcing)

Future of artificial intelligence research:
Is human computation a compromise? or a new direction?

- Ask humans to do tasks computers can not do
  - Does it mean giving up “artificial intelligence”?

→ This might be a new research direction

- “Freestyle” chess tournament:
  A team of 2 amateurs and 3 programs won
  - Computers + humans overwhelms best of each
- Not only being a compromise,
  this might be a new artificial intelligence

Summary:
Machine learning and its applications (and beyond)

- Increasing popularity of machine learning:
  - supported by the rise of “big data” and “data scientists”
- Many applications:
  - Supervised machine learning: recommender systems
    - Matrix factorization/tensor decomposition
  - Unsupervised machine learning: Anomaly detection
    - Capture normal times (by clustering)
- Human computation: Collaborative problem solving by machines and humans