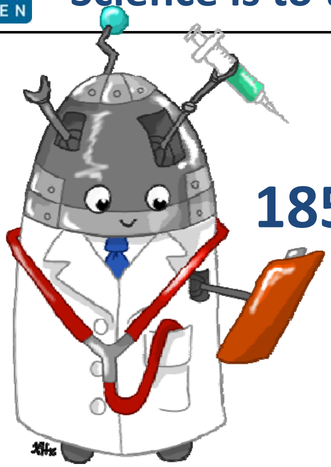


Andreas Holzinger

185.A83 Machine Learning for Health Informatics

2016S, VU, 2.0 h, 3.0 ECTS

Week 25 - 22.06.2016 17:00-20:00

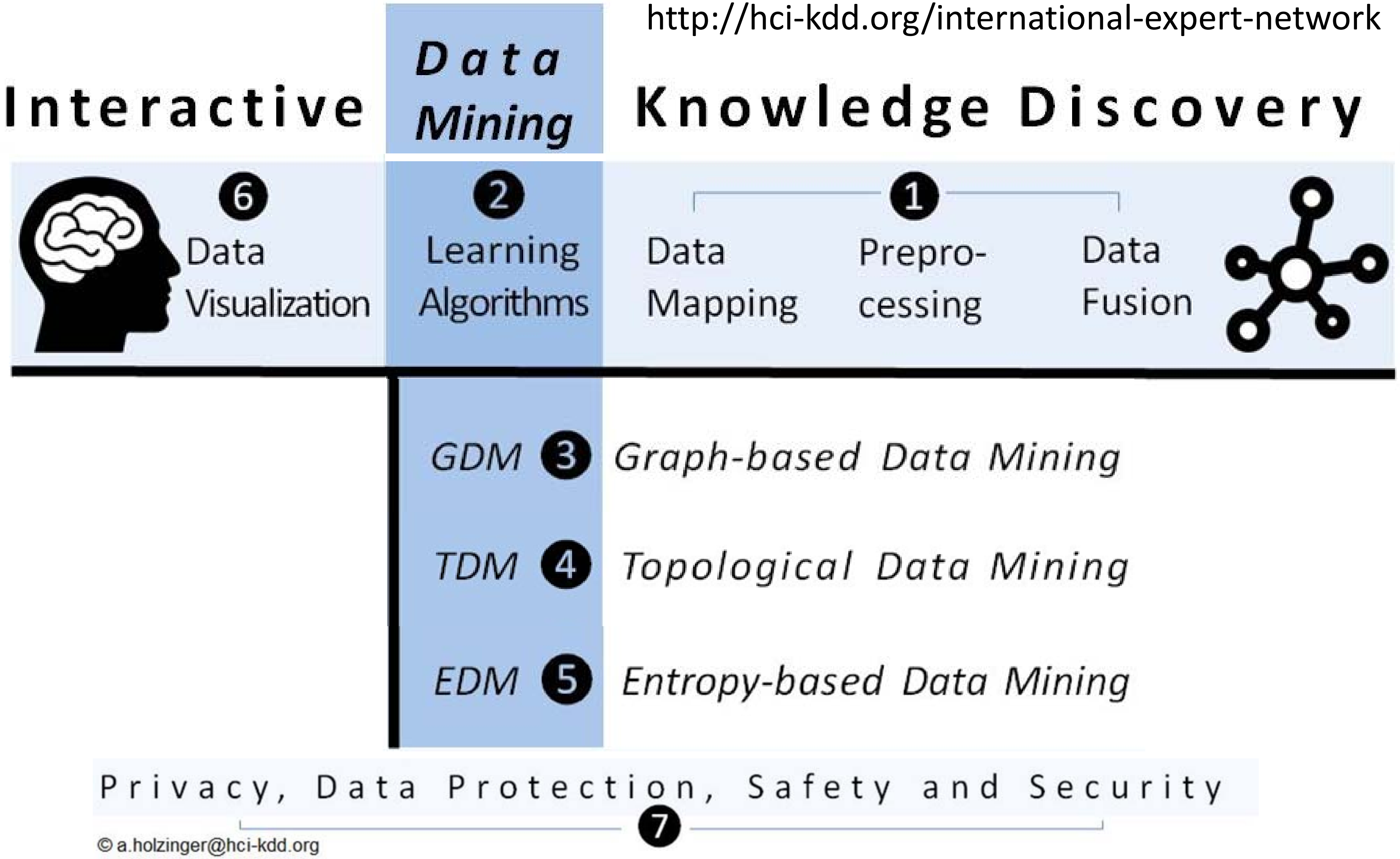
Selected Topics on Active Learning, Multi-Task Learning & Transfer Learning

a.holzinger@hci-kdd.org

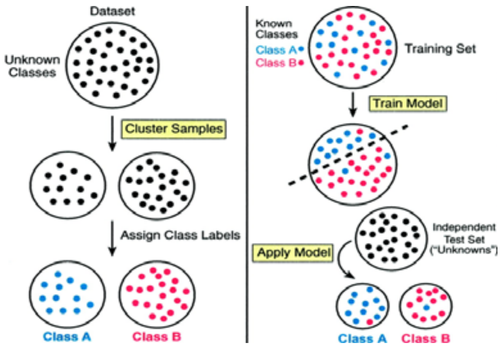
http://hci-kdd.org/machine-learning-for-health-informatics-course

ML needs a concerted effort fostering integrated research

http://hci-kdd.org/international-expert-network

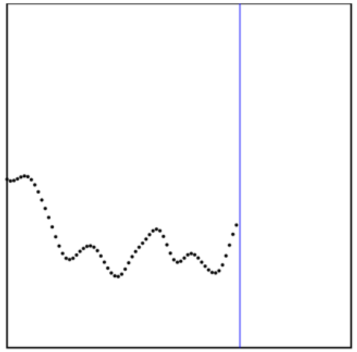


Holzinger, A. 2014. Trends in Interactive Knowledge Discovery for Personalized Medicine: **Cognitive Science meets Machine Learning.** IEEE Intelligent Informatics Bulletin, 15, (1), 6-14.


Dataset
Unknown Classes
Cluster Samples
Assign Class Labels
Class A
Class B
Known Classes
Class A
Class B
Training Set
Train Model
Independent Test Set ("Unknowns")
Apply Model
Class A
Class B

1

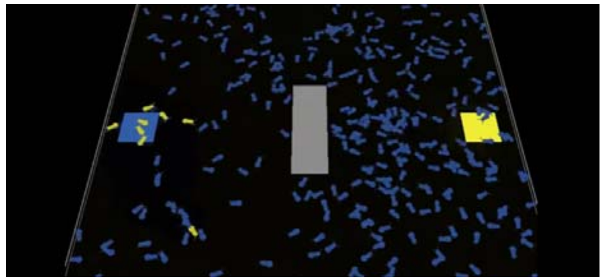
2



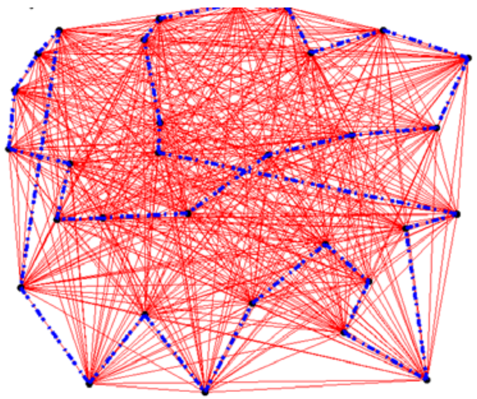
3

4

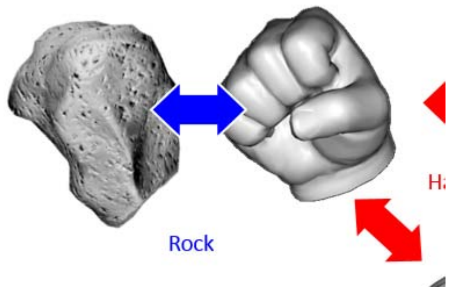
5



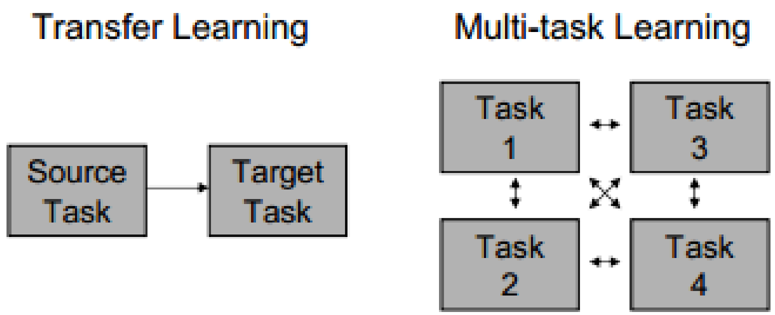
6



7


Rock

- Big data with many training sets (this is good for ML!)
- **Small number of data sets, rare events**
- **Very-high-dimensional problems**
- **Complex data – NP-hard problems**
- **Missing, dirty, wrong, noisy, …, data**
- **GENERALISATION**
- **TRANSFER**



Torrey, L. & Shavlik, J. 2009. Transfer learning. Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques, 242-264, doi:10.4018/978-1-60566-766-9.ch011.

Timeline

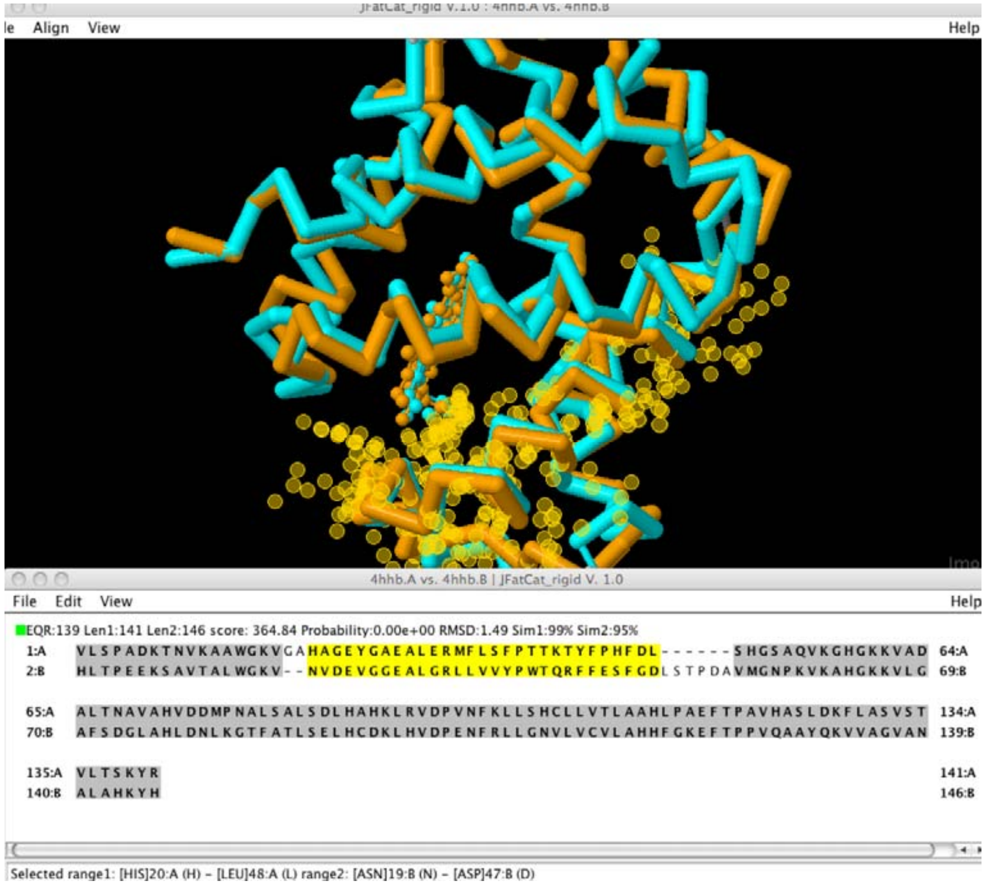
- Probabilistic Learning (1763)
- Reinforcement Learning (1950)
- Preference Learning (1987)
- Active Learning (1996)
- Active Preference Learning (2005)
- Interactive Learning and Optimization (2010)
- Interactive ML with the “human-in-the-loop” …

Red thread through this lecture

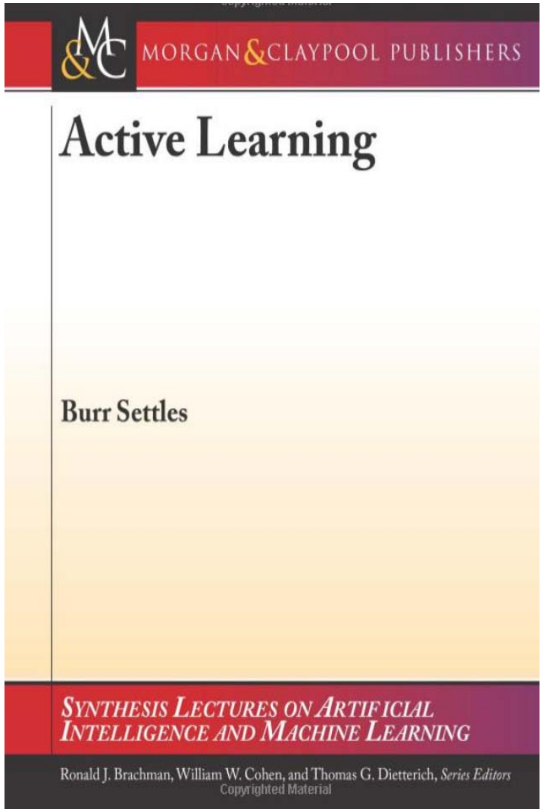
- 1) Active Learning
- 2) Preference Learning
- 3) Active Preference Learning
- 4) Multi-Task Learning
- 5) Transfer Learning

1) Active Learning

Motivation: classic fully supervised learning is expensive

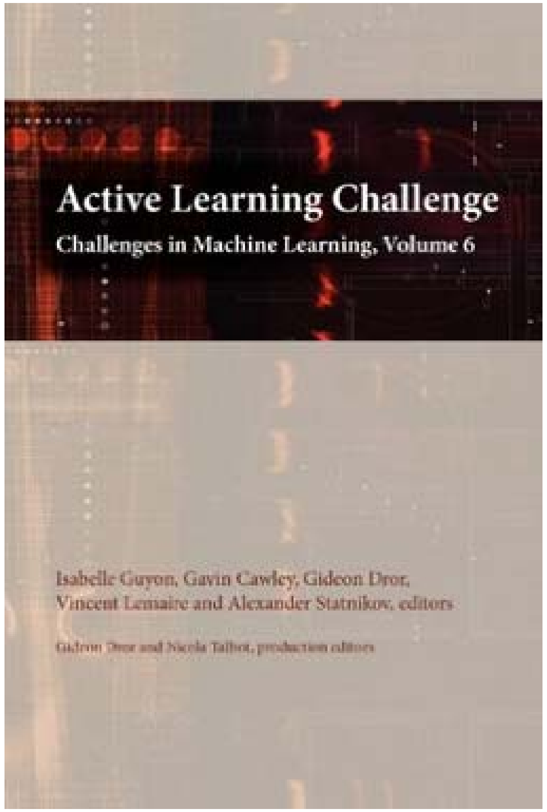


http://www.rcsb.org/pdb/general_information
/new_images/1002_aligdisplay.png



Settles, B. 2012. Active Learning, San Rafael (CA), Morgan & Claypool, doi:10.2200/S00429ED1V01Y201207AIM018.

http://active-learning.net



Guyon, I., Cawley, G., Dror, G., Lemaire, V. & Statnikov, A. 2012. Active learning challenge: Challenges in machine learning, volumn 6. Microtome Publishing, River Edge, NJ, USA.

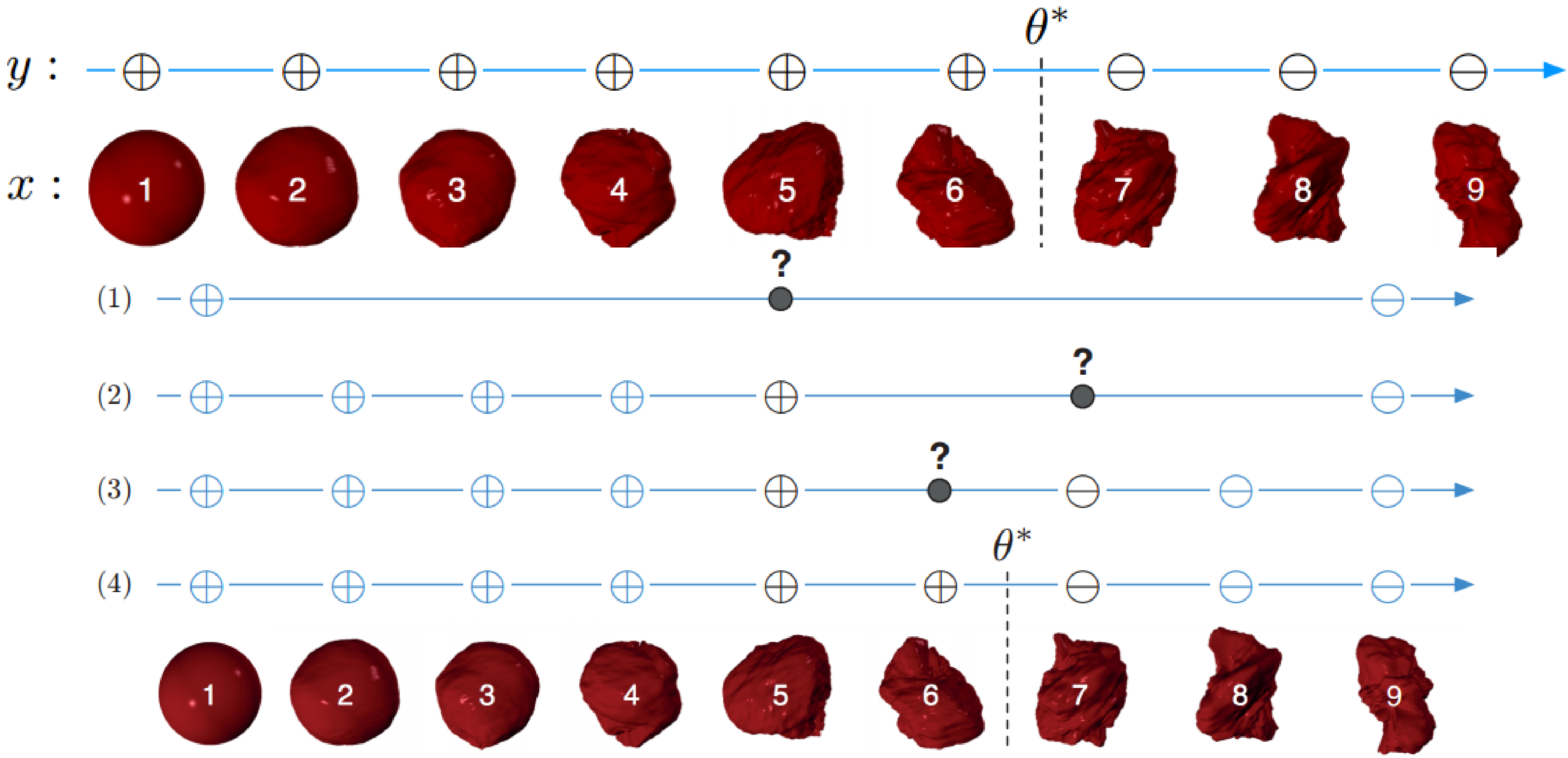
What is this …

https://en.wikipedia.org/wiki/Delphi

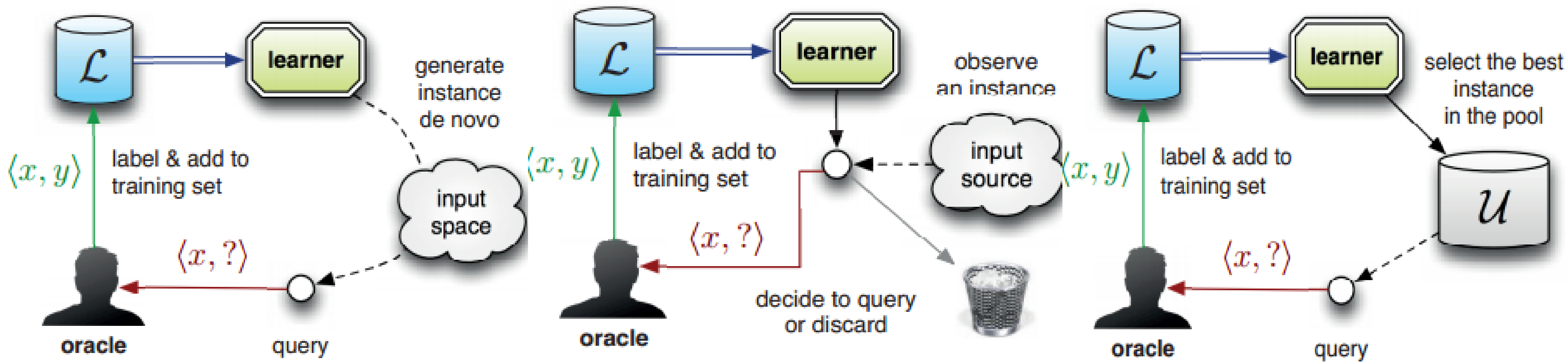
- := ML algorithm can perform better with less training if it is allowed to choose the data from which it learns.
- "Active learner" may pose queries, usually in the form of unlabeled data instances to be labeled by an **"oracle"** (e.g., a human annotator) that understands the nature of the problem.
- It is useful, where unlabeled data is abundant or easy to obtain, but training labels are difficult, time-consuming, or expensive to obtain

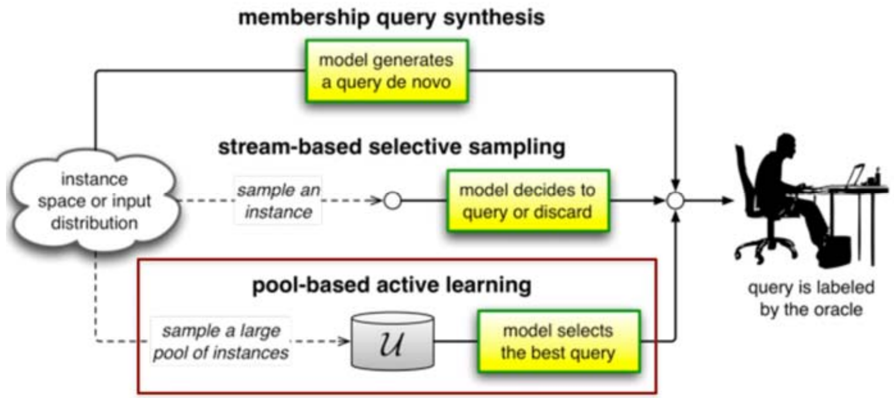
- A classifier to determine objects as a function mapping $h\colon X \rightarrow Y$, parameterized by a threshold θ:

$$h(x;\theta) = \begin{cases} \oplus \text{ safe} & \text{if } x < \theta \text{, and} \\ \ominus \text{ noxious} & \text{otherwise.} \end{cases}$$



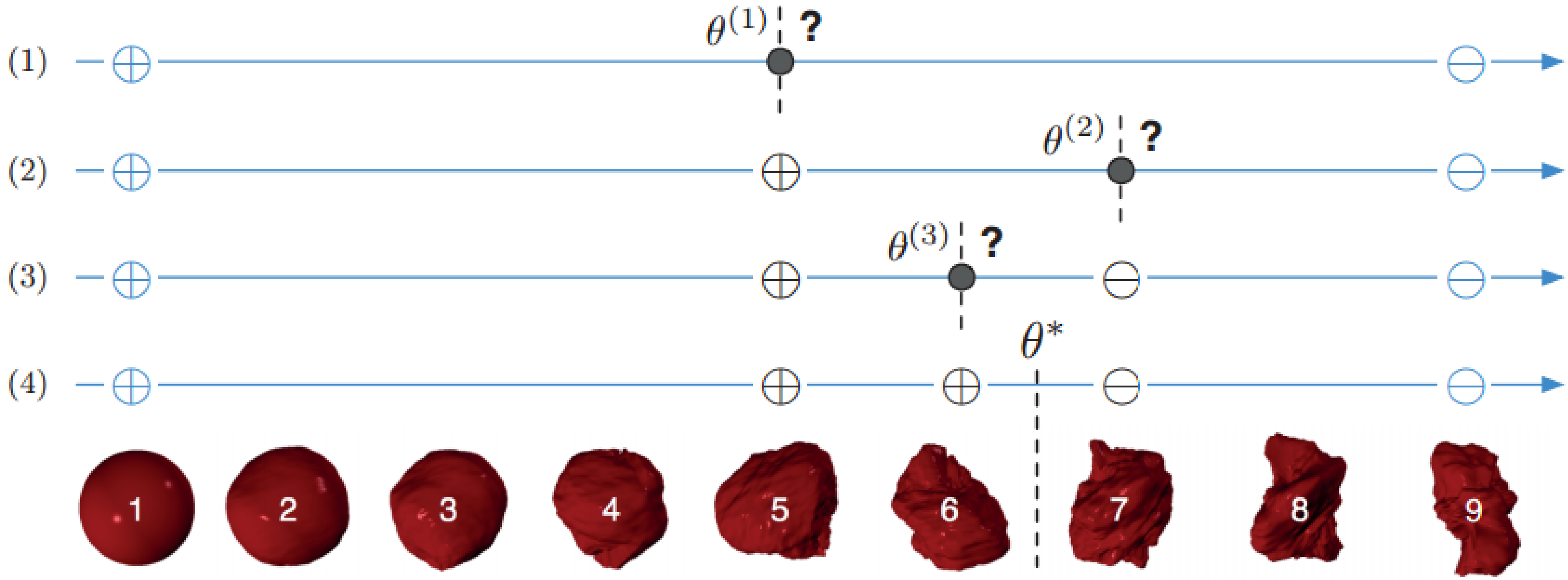
Scenarios for active learning





Settles, B. 2012. Active Learning, San Rafael (CA), Morgan & Claypool, doi:10.2200/S00429ED1V01Y201207AIM018.

Uncertainty Sampling



$\mathcal{U}$ = a pool of unlabeled instances $\{x^{(u)}\}_{u=1}^{U}$
$\mathcal{L}$ = set of initial labeled instances $\{\langle x, y\rangle^{(l)}\}_{l=1}^{L}$
for $t = 1, 2, \ldots$ **do**
    $\theta = \textbf{train}(\mathcal{L})$
    select $x^* \in \mathcal{U}$, the most uncertain instance according to model θ
    query the oracle to obtain label y^*
    add $\langle x^*, y^*\rangle$ to $\mathcal{L}$
    remove x^* from $\mathcal{U}$
end for

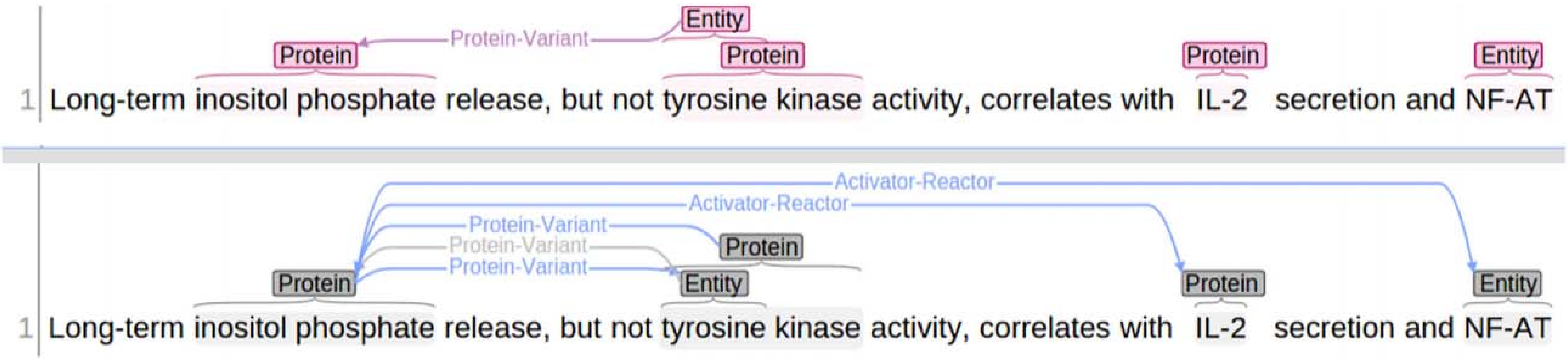
Settles, B. 2012. Active Learning, San Rafael (CA), Morgan & Claypool, doi:10.2200/S00429ED1V01Y201

Multi-Task Active learning

- The typical active learning setting assumes a single machine learner trying to solve a single task.
- In many real-world problems, however, the same data might be labeled in multiple ways for several different subtasks. In such cases, it is probably more economical to label a single instance for all subtasks simultaneously, or to choose instance-task query pairs that provide as much information as possible to all tasks. This motivates the need for multi-task active learning algorithms.
- If we take a multi-task entropy-based uncertainty sampling sort of approach, then we might want to select instances with the highest joint conditional entropy of both labels given the instance: $H\theta\ (Y1, Y2|x)$, where $Y1$ and $Y2$ denote the output variables for the two different tasks.

Settles, B. 2012. Active Learning, San Rafael (CA), Morgan & Claypool, doi:10.2200/S00429ED1V01Y201207AIM018.

Example for the Human-in-the-Loop

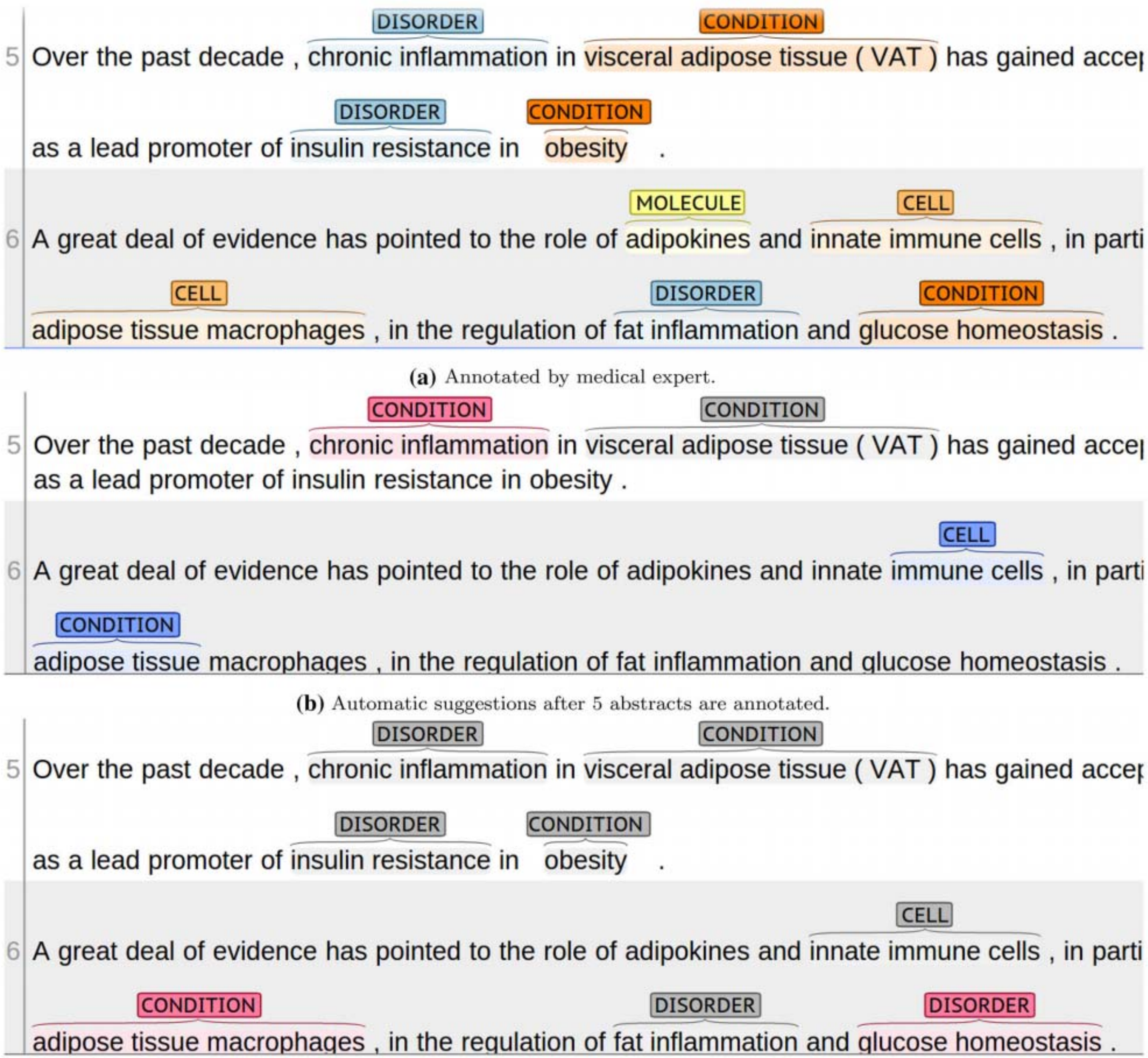


Mode	Annotator type	Recall	Precsion	F-score
Automation				
	Entity	61.94	49.31	54.91
	Protein	57.31	50.97	53.95
Expert				
	Entity	29.11	22.90	25.63
	Protein	71.94	59.28	65.00

Yimam, S. M., Biemann, C., Majnaric, L., Šabanović, Š. & Holzinger, A. 2016. An adaptive annotation approach for biomedical entity and relation recognition. Brain Informatics, 1-12, doi:10.1007/s40708-016-0036-4.

Example for the Human-in-the-Loop

Yimam, S. M., Biemann, C., Majnaric, L., Šabanović, Š. & Holzinger, A. 2016. An adaptive annotation approach for biomedical entity and relation recognition. Brain Informatics, 1-12, doi:10.1007/s40708-016-0036-4.



Functional genomic hypothesis generation and experimentation by a robot scientist

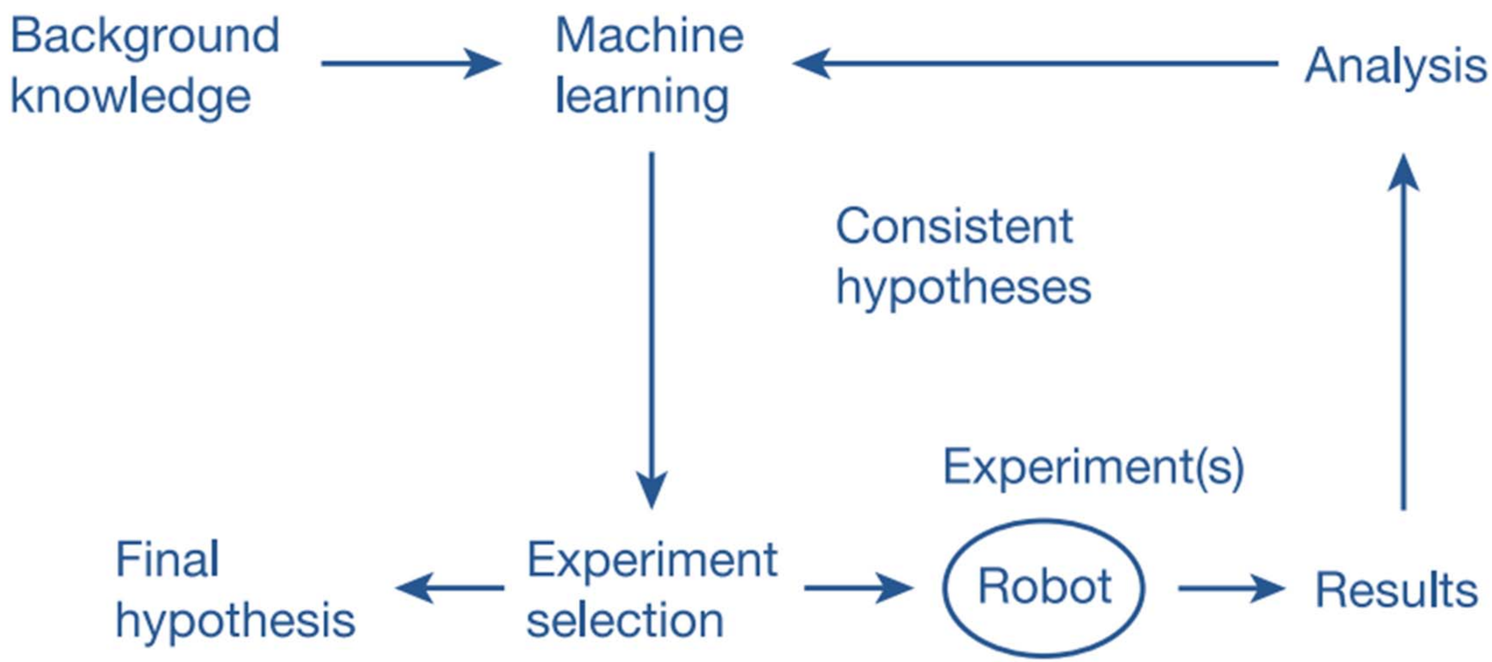
Ross D. King[1], **Kenneth E. Whelan**[1], **Ffion M. Jones**[1], **Philip G. K. Reiser**[1], **Christopher H. Bryant**[2], **Stephen H. Muggleton**[3], **Douglas B. Kell**[4] **& Stephen G. Oliver**[5]

[1] *Department of Computer Science, University of Wales, Aberystwyth SY23 3DB, UK*
[2] *School of Computing, The Robert Gordon University, Aberdeen AB10 1FR, UK*
[3] *Department of Computing, Imperial College, London SW7 2AZ, UK*
[4] *Department of Chemistry, UMIST, P.O. Box 88, Manchester M60 1QD, UK*
[5] *School of Biological Sciences, University of Manchester, 2.205 Stopford Building, Manchester M13 9PT, UK*

The question of whether it is possible to automate the scientific process is of both great theoretical interest[1,2] and increasing practical importance because, in many scientific areas, data are being generated much faster than they can be effectively analysed. We describe a physically implemented robotic system that applies techniques from artificial intelligence[3–8] to carry out cycles of scientific experimentation. The system automatically

Example of a real-world application: Robot Scientist

- Query synthesis: “robot scientist” executes autonomously biological experiments to discover metabolic pathways in yeast (saccharomyces cerevisiae).



King, R. D., Whelan, K. E., Jones, F. M., Reiser, P. G., Bryant, C. H., Muggleton, S. H., Kell, D. B. & Oliver, S. G. 2004. Functional genomic hypothesis generation and experimentation by a robot scientist. Nature, 427, (6971), 247-252.

Approximate estimation of the expected cost

Let $EC(H, T)$ denote the minimum expected cost of experimentation given the set of candidate hypotheses H and the set of candidate trials T

- C_t … price of the trial t
- $p(t)$ … probability of the outcome
- $\lfloor \ldots \rfloor$ … is the "floor" function
- $p(t)$ can be computed as the sum of the probabilities of the
- hypotheses (h) that are consistent with a positive outcome of t

$$EC(\emptyset, T) = 0$$

$$EC(\{h\}, T) = 0$$

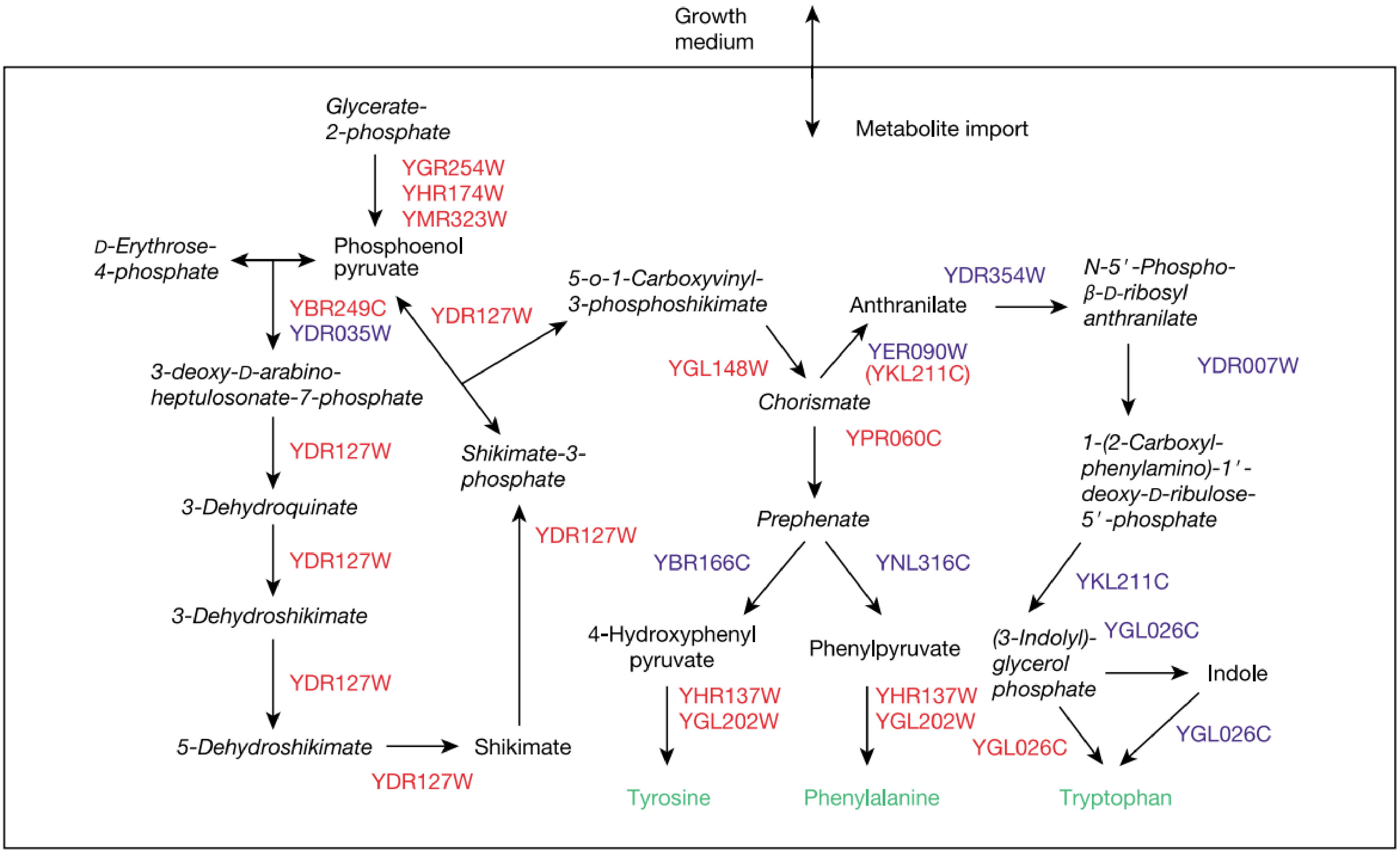
$$EC(H, T) \approx \min_{t \in T}[C_t + p(t)(\text{mean}_{t' \in (T-t)})C_{t'})J_{H[t]} + (1 - p(t))$$

$$\times(\text{mean}_{t' \in (T-t)}C_{t'})J_{H[t]}]$$

$$J_H = -\Sigma_{h \in H} p(h) \lfloor \log_2(p(h)) \rfloor$$

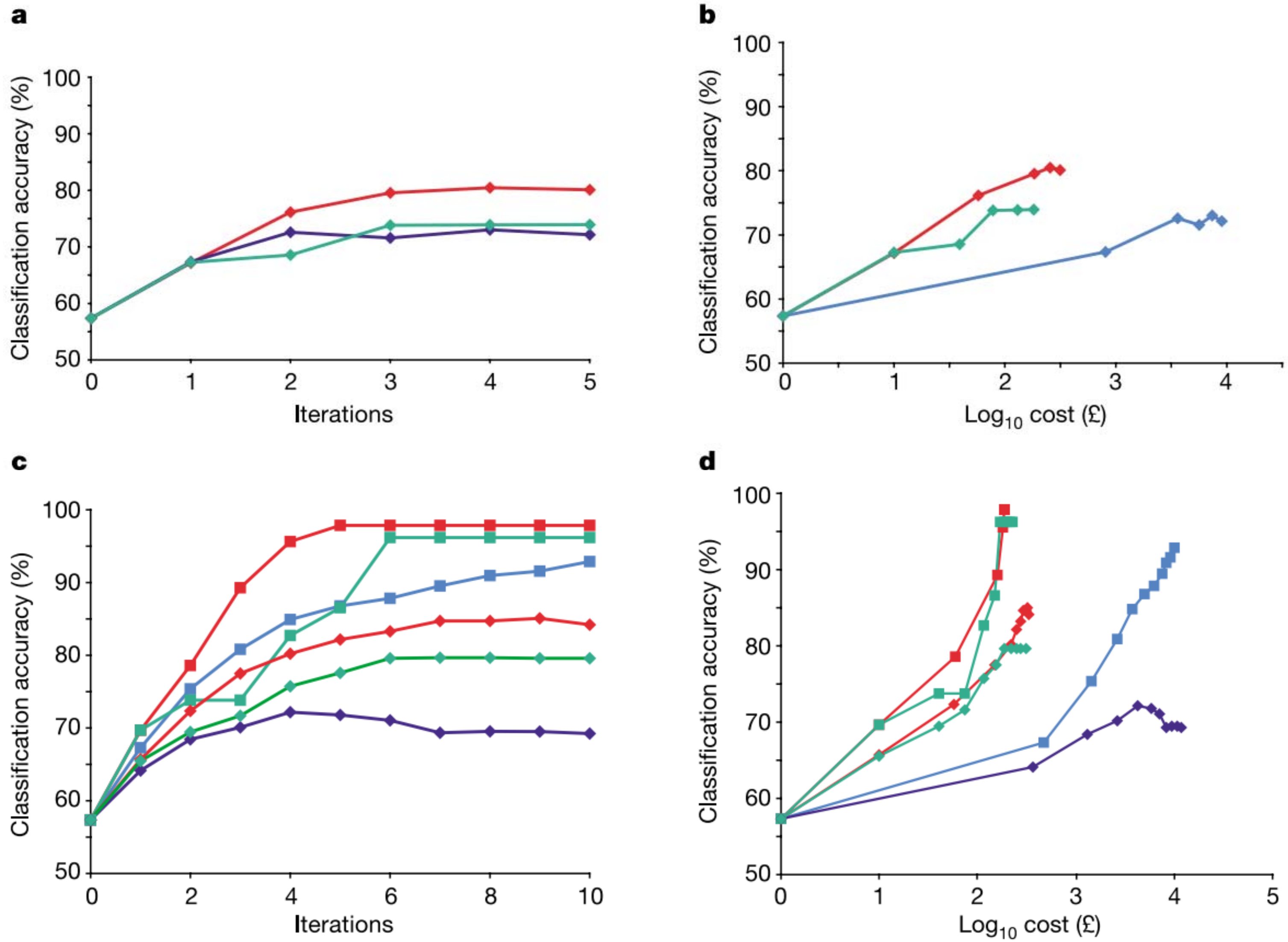
King, R. D., Whelan, K. E., Jones, F. M., Reiser, P. G., Bryant, C. H., Muggleton, S. H., Kell, D. B. & Oliver, S. G. 2004. Functional genomic hypothesis generation and experimentation by a robot scientist. Nature, 427, (6971), 247-252.

Logical model of the aromatic amino acid yeast pathway



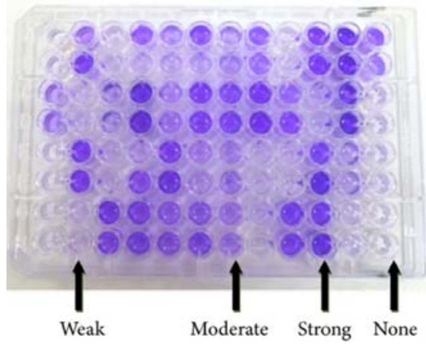
King, R. D., Whelan, K. E., Jones, F. M., Reiser, P. G., Bryant, C. H., Muggleton, S. H., Kell, D. B. & Oliver, S. G. 2004. Functional genomic hypothesis generation and experimentation by a robot scientist. Nature, 427, (6971), 247-252.

In science, as in industry, “time is money” (although the conversion rate may be unclear)

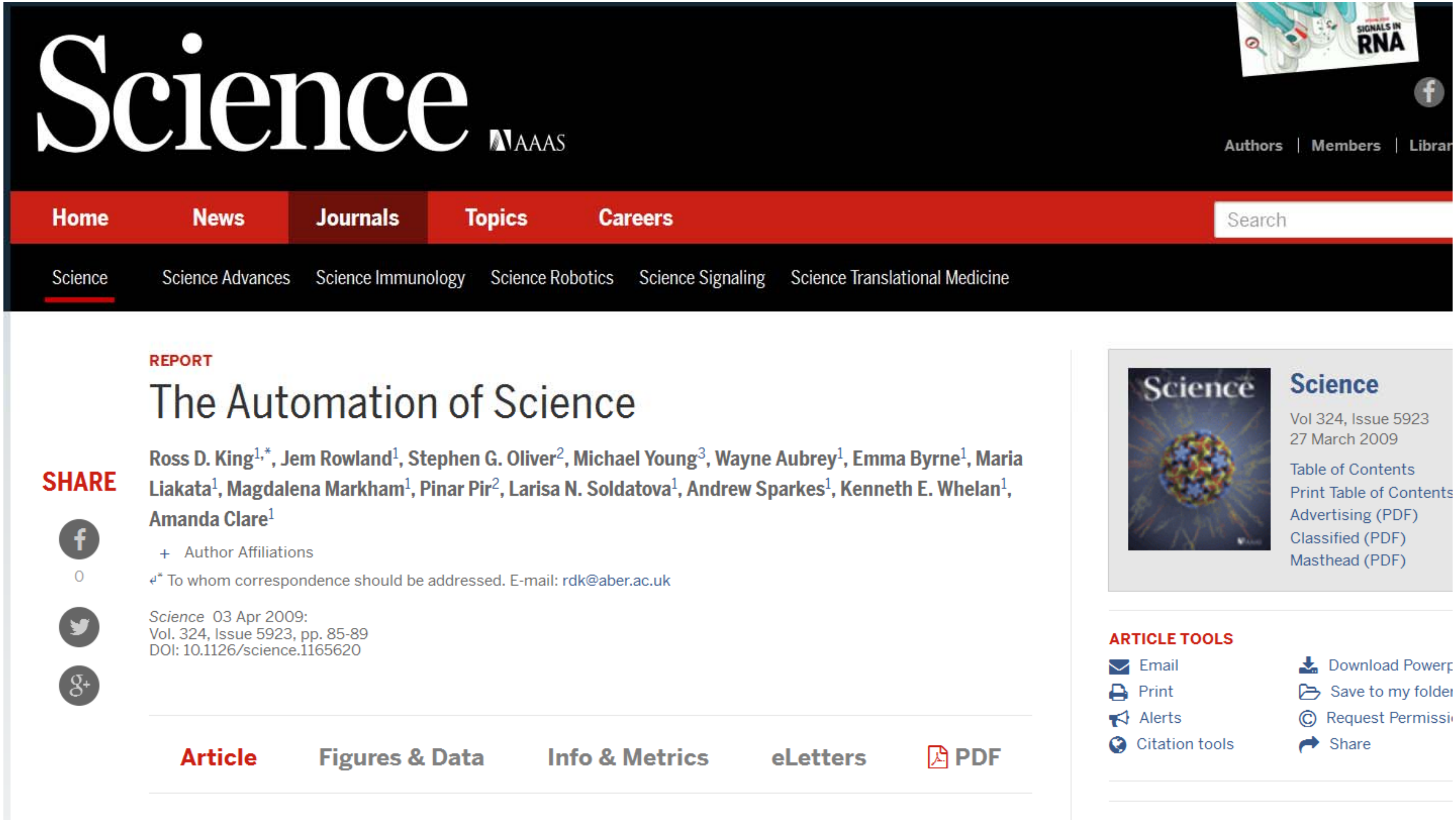


King, R. D., Whelan, K. E., Jones, F. M., Reiser, P. G., Bryant, C. H., Muggleton, S. H., Kell, D. B. & Oliver, S. G. 2004. Functional genomic hypothesis generation and experimentation by a robot scientist. Nature, 427, (6971), 247-252.

- The Robot Scientist automates the task of liquid handling and conducts assays by pipetting/mixing liquids on micro-titres.
- The robot is controlled using Tcl (Tool command language)
- A compiler translates Prolog commands into Tcl robot operations.
- The robot was programmed to automatically plate out the yeast and media into the correct wells. The micro-titre plates were measured with the adjacent plate reader and the results were returned to the LIMS.
- However, transfer of plates from the robot to the incubator, and from the incubator to the plate reader, was done manually.



King, R. D., Whelan, K. E., Jones, F. M., Reiser, P. G., Bryant, C. H., Muggleton, S. H., Kell, D. B. & Oliver, S. G. 2004. Functional genomic hypothesis generation and experimentation by a robot scientist. Nature, 427, (6971), 247-252.



Science AAAS

Authors | Members | Librar

Home News Journals Topics Careers

Search

Science Science Advances Science Immunology Science Robotics Science Signaling Science Translational Medicine

REPORT

The Automation of Science

Ross D. King[1,*], Jem Rowland[1], Stephen G. Oliver[2], Michael Young[3], Wayne Aubrey[1], Emma Byrne[1], Maria Liakata[1], Magdalena Markham[1], Pinar Pir[2], Larisa N. Soldatova[1], Andrew Sparkes[1], Kenneth E. Whelan[1], Amanda Clare[1]

+ Author Affiliations

↵* To whom correspondence should be addressed. E-mail: rdk@aber.ac.uk

Science 03 Apr 2009:
Vol. 324, Issue 5923, pp. 85-89
DOI: 10.1126/science.1165620

SHARE

0

Article Figures & Data Info & Metrics eLetters PDF

Science

Vol 324, Issue 5923
27 March 2009

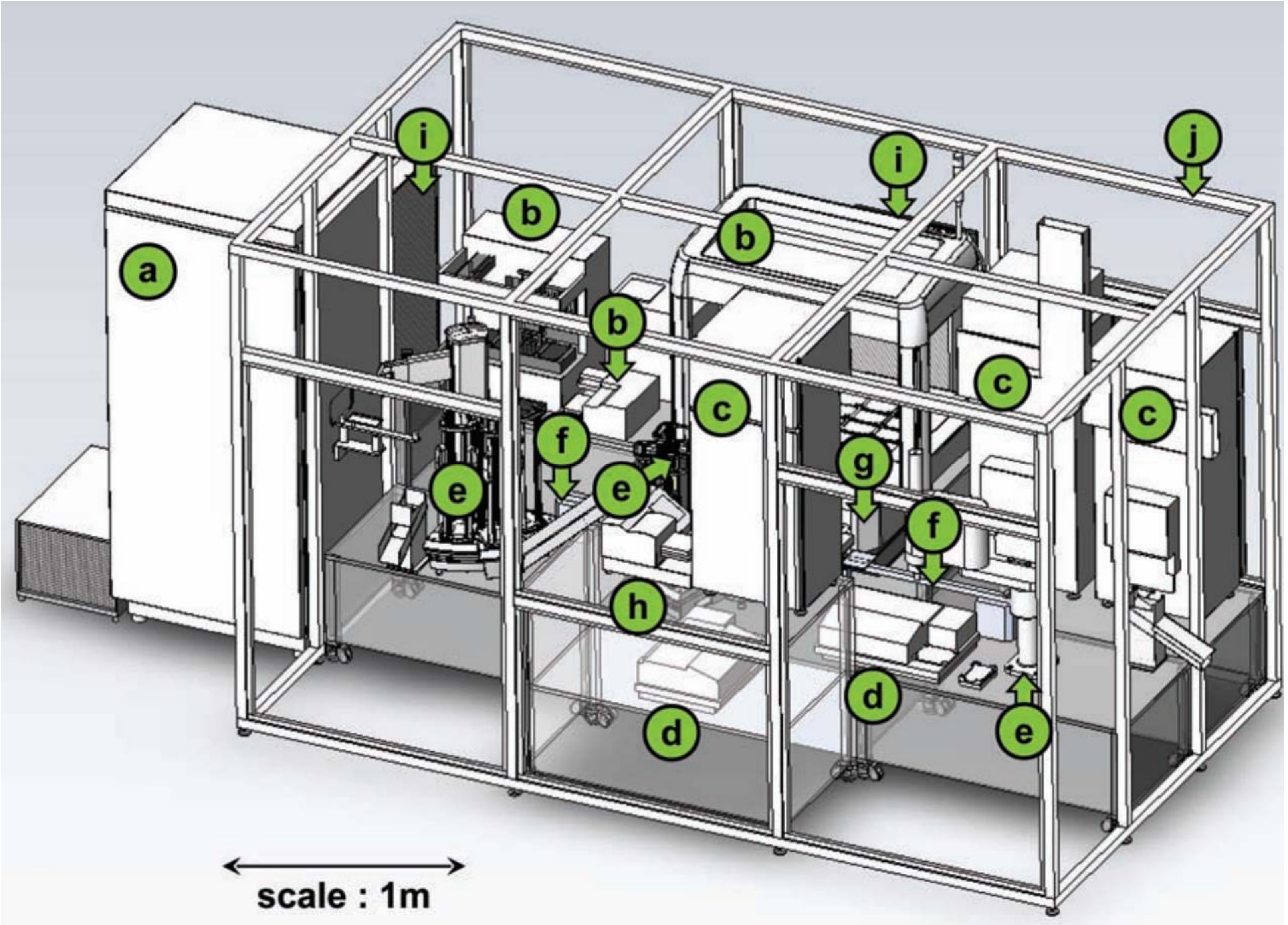
Table of Contents
Print Table of Contents
Advertising (PDF)
Classified (PDF)
Masthead (PDF)

ARTICLE TOOLS

Email
Print
Alerts
Citation tools
Download Powerp
Save to my folder
Request Permissi
Share

King, R. D., Rowland, J., Oliver, S. G., Young, M., Aubrey, W., Byrne, E., Liakata, M., Markham, M., Pir, P. & Soldatova, L. N. 2009. The automation of science. Science, 324, (5923), 85-89, doi:10.1126/science.1165620.

The Automation of Science: The Robot Scientist Adam



King, R. D., Rowland, J., Oliver, S. G., Young, M., Aubrey, W., Byrne, E., Liakata, M., Markham, M., Pir, P. & Soldatova, L. N. 2009. The automation of science. Science, 324, (5923), 85-89, doi:10.1126/science.1165620.

The Automation of Science: The Robot Scientists

1999-2004 Initial Robot Scientist Project

- Limited Hardware
- Collaboration with Douglas Kell (Aber Biology), Steve Oliver (Manchester), Stephen Muggleton (Imperial)

King *et al.* (2004) *Nature*, **427**, 247-252

2004-2011 Adam Project

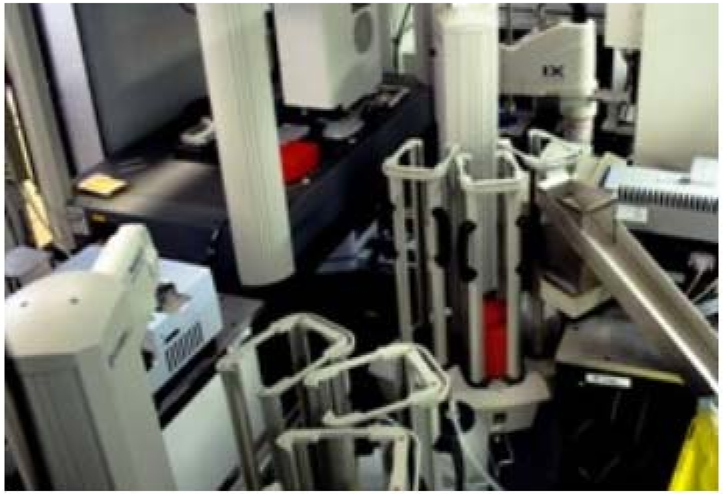
- Sophisticated Laboratory Automation
- Collaboration with Steve Oliver (Cambridge).

King *et al.* (2009) *Science*, **324**, 85-89

2008-2011 Eve Project

King, R. D., Rowland, J., Oliver, S. G., Young, M., Aubrey, W., Byrne, E., Liakata, M., Markham, M., Pir, P. & Soldatova, L. N. 2009. The automation of science. Science, 324, (5923), 85-89, doi:10.1126/science.1165620.

The Automation of Science: The Robot Scientist Adam



King, R. D., Rowland, J., Oliver, S. G., Young, M., Aubrey, W., Byrne, E., Liakata, M., Markham, M., Pir, P. & Soldatova, L. N. 2009. The automation of science. Science, 324, (5923), 85-89, doi:10.1126/science.1165620.

Statistical learning – Probable Approximate Correct (PAC)

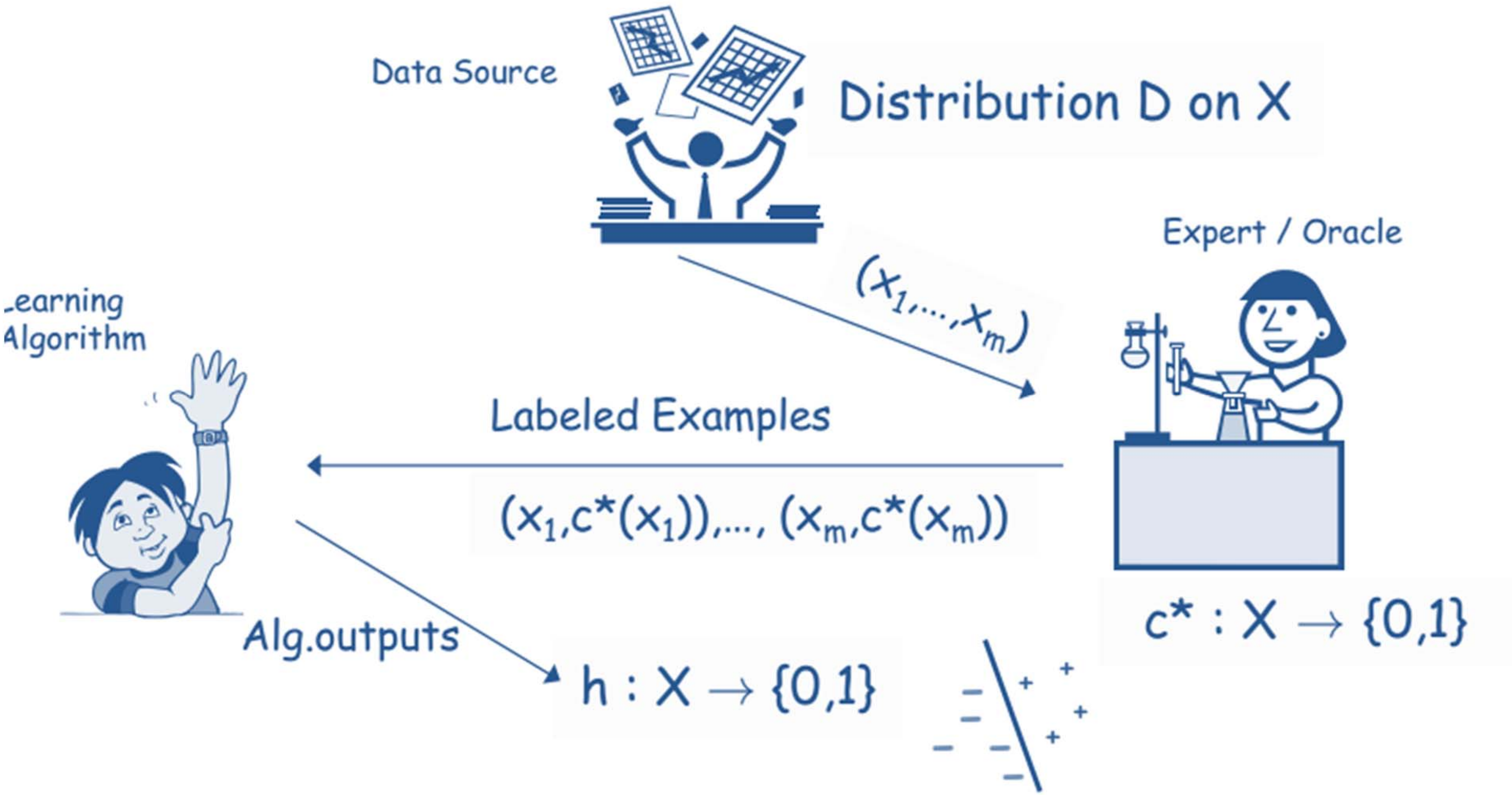


Image credit to Maria-Florina Balcan, CMU

- Design a predictor based on unlabeled and few randomly labeled data sets
- Assumption: The knowledge of marginal density may simplify prediction, e.g. similar sets have similar labels

$$\{(X_i, Y_i)\}_{i=1}^{n} \quad \{X_j\}_{j=1}^{m} \Rightarrow \boxed{\text{Learning algorithm}} \Rightarrow \widehat{f}_{m,n}$$

2) Preference Learning

www.preference-learning.org

Log in

>PL

Home | Software | Datasets | Research Groups | Workshops | Tutorials | Books | Publications

You are here: Home

Welcome to the preference learning site

Software

- LPCforSOS Framework (Johannes Fürnkranz)
- Weka extension for Label Ranking (Eyke Hüllermeier)
- SVM-Rank (Thorsten Joachims)
- Preference Learning Toolbox (Georgios Yannakakis)

Datasets

- The Sushi Preference Dataset
- LETOR Benchmark Collection for Learning to Rank
- Label ranking data (semi-synthetic)
- Car Preference Dataset

Research Groups

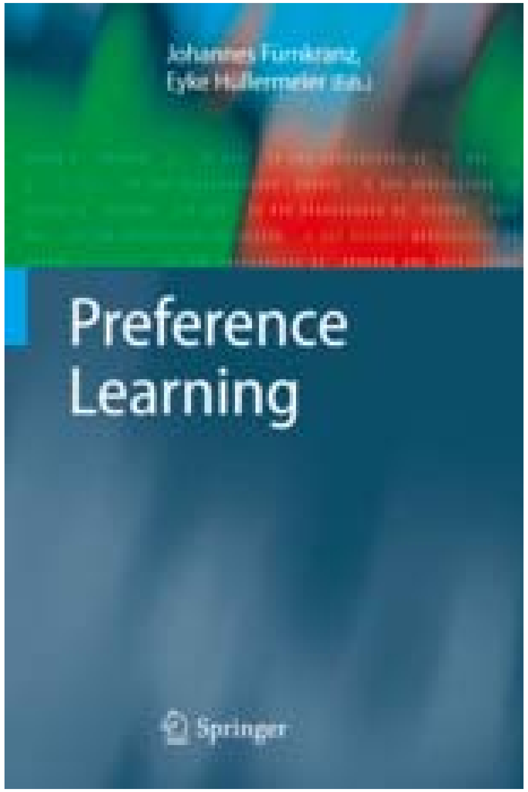
- Univ. degli Studi di Padova (Fabio Aiolli)
- Universiteit Gent (Bernard De Baets)
- TU Darmstadt (Johannes Fürnkranz)
- Philipps-Universität Marburg (Eyke Hüllermeier)
- Cornell University (Thorsten Joachims)
- Microsoft Research Asia (Tie-Yan Liu)
- MIT (Cynthia Rudin)
- Austalian National University (Scott Sanner)
- Institute of Digital Games (Georgios Yannakakis)

Workshops

- Reinforcement Learning with Generalized Feedback: Beyond Numeric Rewards (PBRL-13) at ECML/PKDD-13
- Preference Learning: Problems and Applications in Artificial Intelligence (PL-12) at ECAI-12
- Choice Models and Preference Learning (CMPL-11) at NIPS-11
- Preference Learning (PL-10) at ECML/PKDD 2010

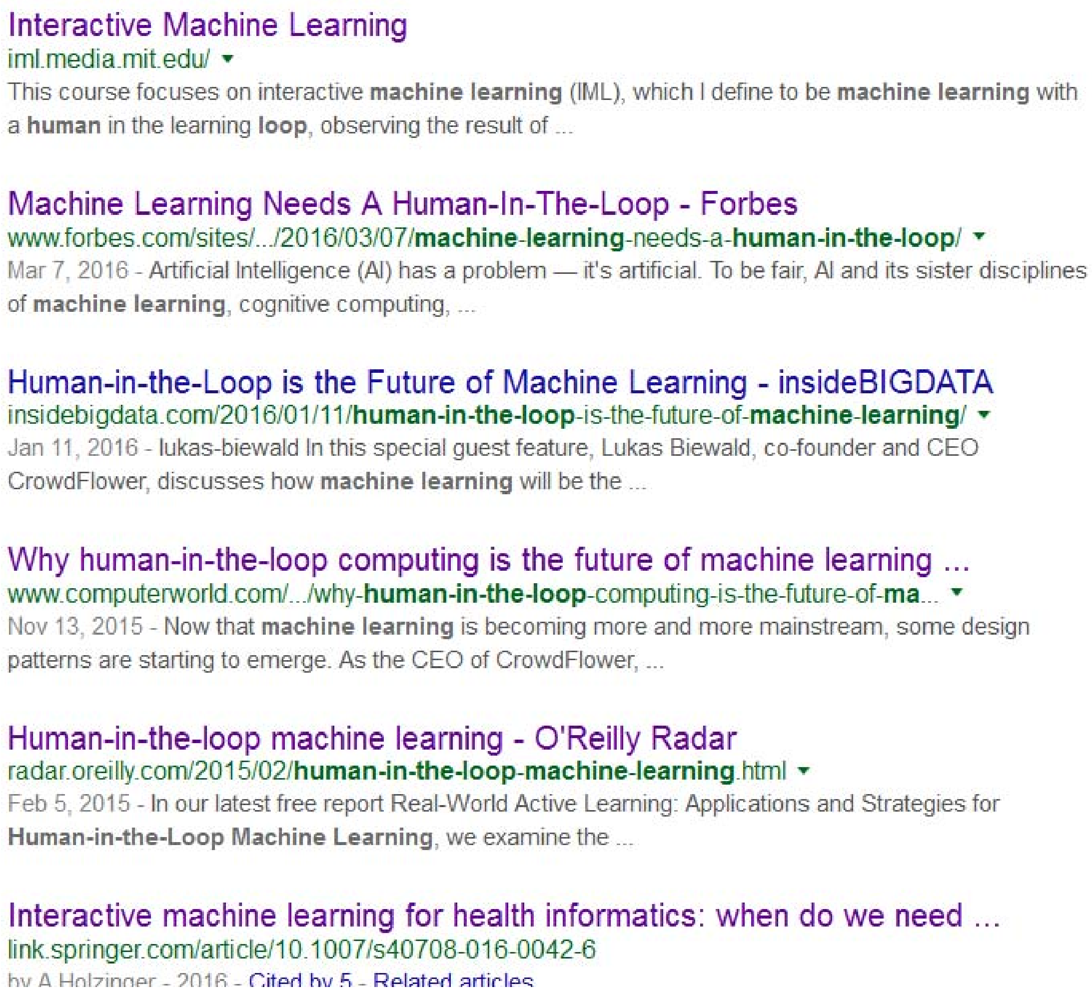
CONTENTS

“The” Recommended Book



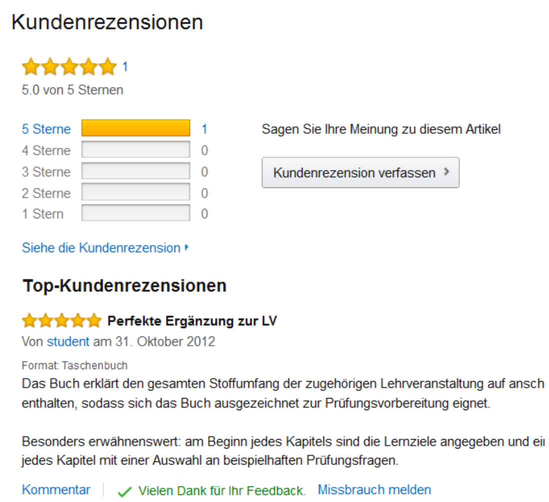
Fuernkranz, J. & Hüllermeier, E. 2010. Preference learning, Berlin Heidelberg, Springer.

What are “preferences”?

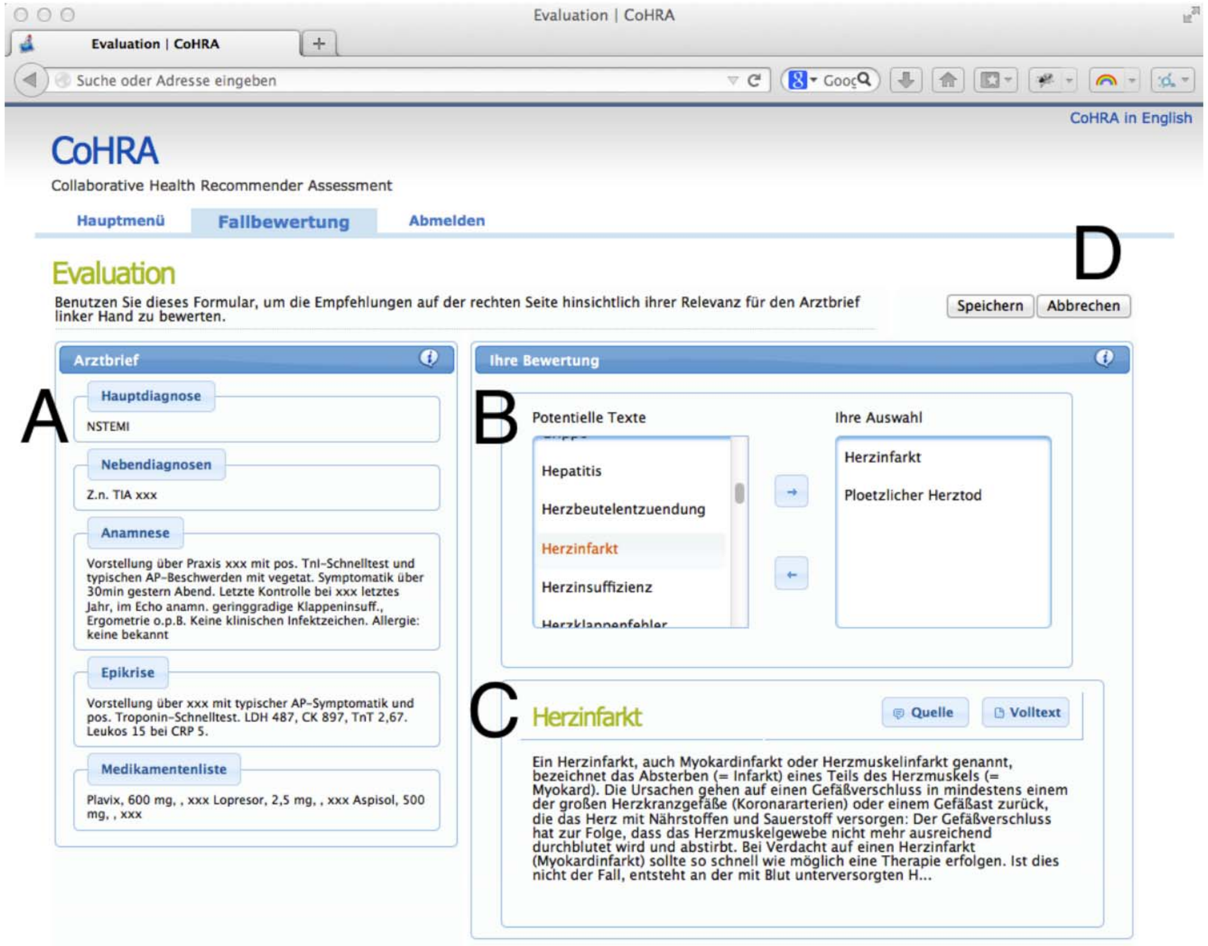


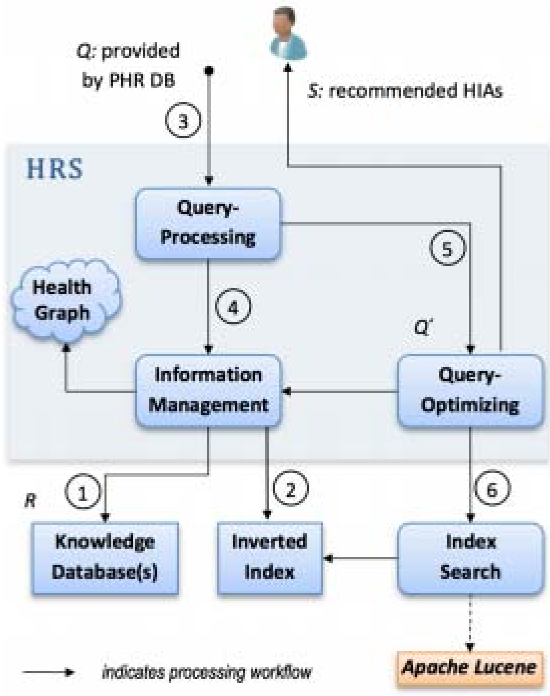
“clicked”

“NOT clicked”



Is Preference Learning relevant for Health Informatics ?





Wiesner, M. & Pfeifer, D. 2014. Health recommender systems: concepts, requirements, technical basics and challenges. International journal of environmental research and public health, 11, (3), 2580-2607.

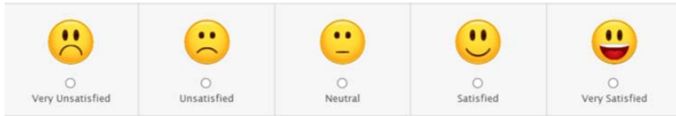
Preference Learning is relevant for Health Informatics

- Deals with the learning of predictive preference models from observed (extracted) preference information – highly relevant for decision theory
- User preferences play a key role in
  - Recommender systems
  - Autonomous agents and games
  - Adaptive user interfaces
  - Adaptive .... x ... systems

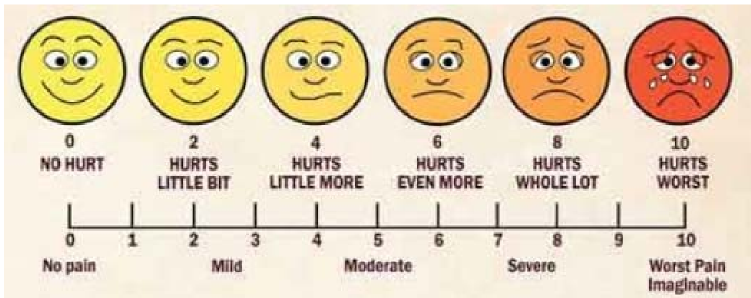
Fuernkranz, J. & Hüllermeier, E. 2010. Preference learning, Berlin Heidelberg, Springer.

PL Basics pose a lot of learning problems

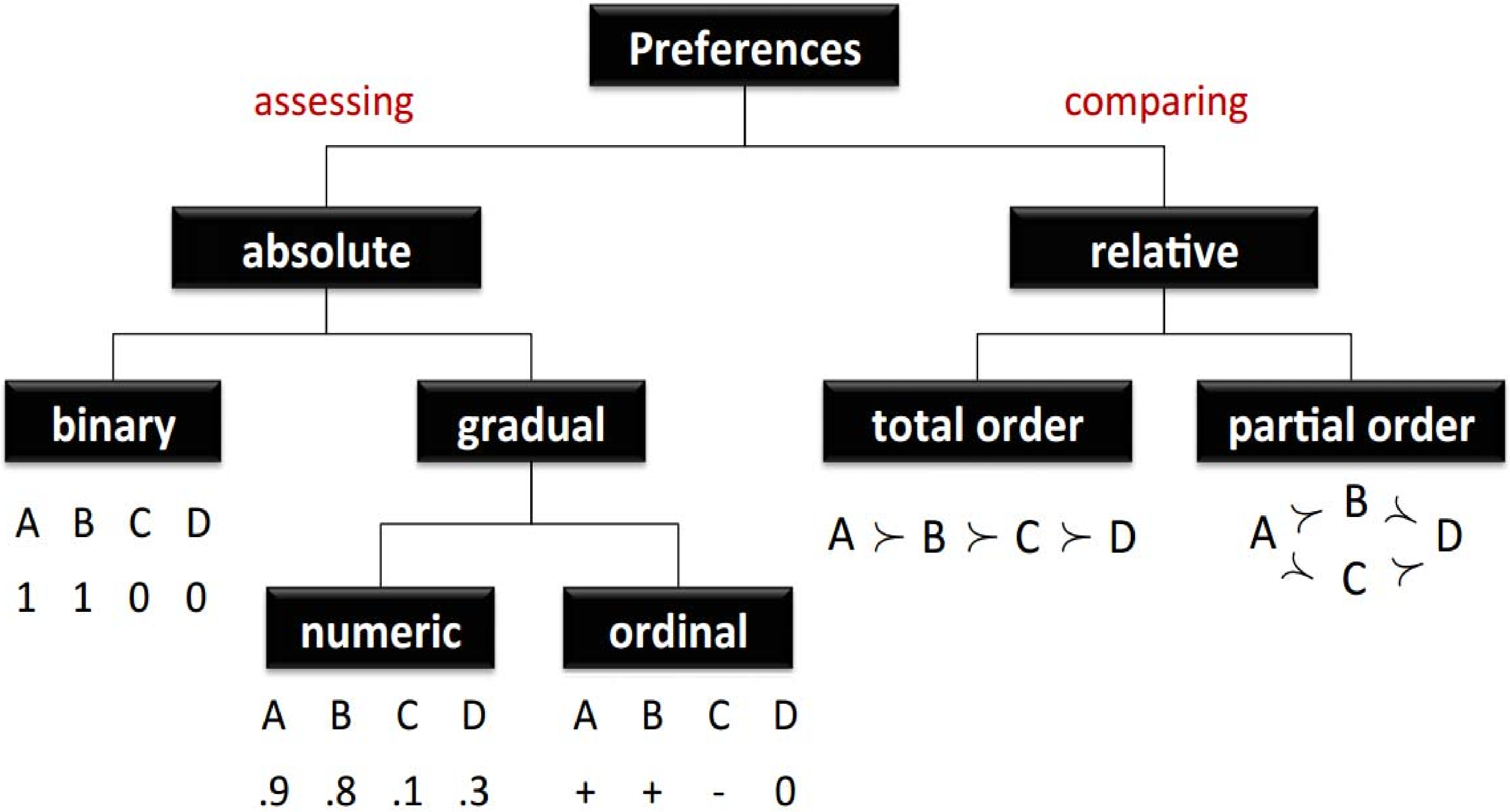
- Single vs. multi-dimensional
- Explicit vs. implicit (e.g. direct vs. click-through)
- Absolute vs. relative (e.g. assessing vs. comparing)
- Structured vs. unstructured (ratings vs. free-text)
- Single-User vs. multiple users (social tagging)
- Binary vs. graded (relevance judges vs. ratings)



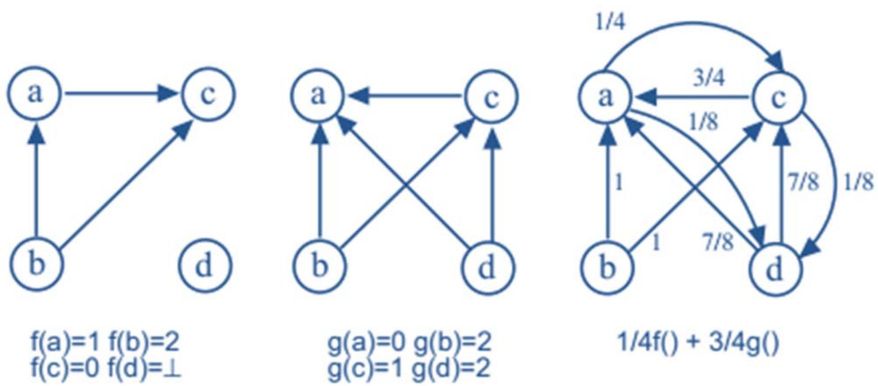
- Three main types of problem dimensions:
- 1) Representation of preferences
  - Utility function (e.g. ordinal, numeric, …)
  - Preference relation (partial order, ranking, …)
  - Logical representation
- 2) Description of individuals/users and alternatives/items
  - Identifiers
  - Feature vectors
  - Structured objects
- 3) Type of training input
  - Direct or indirect feedback
  - Complete or incomplete relation
  - Utilities, …



Fuernkranz, J. & Hüllermeier, E. 2010. Preference learning, Berlin Heidelberg, Springer.



Fuernkranz, J. & Hüllermeier, E. 2010. Preference learning, Berlin Heidelberg, Springer.



$$R_f(u,v) = \begin{cases} 1 & \text{if } f(u) > f(v) \\ 0 & \text{if } f(u) < f(v) \\ \frac{1}{2} & \text{otherwise.} \end{cases}$$

We call R_f a rank ordering for X into S. If $R_f\ (u;\ v)\ =\ 1$, then we say that u is preferred to v, or u is ranked higher than v.

Cohen, W. W., Schapire, R. E. & Singer, Y. 1999. Learning to Order Things. Journal of Artificial Intelligence Research, 10, 243-270.

Allocate Weights for Ranking Experts

Parameters: $\beta \in [0,1]$, initial weight vector $\mathbf{w}^1 \in [0,1]^N$ with $\sum_{i=1}^N w_i^1 = 1$
N ranking experts, number of rounds T

Do for $t = 1, 2, \ldots, T$

1. Receive a set of elements X^t and ordering functions $f_1^t, \ldots, f_N^t$. Let R_i^t denote the preference function induced by f_i^t.
2. Compute a total order $\hat{\rho}^t$ which approximates

$$\mathrm{PREF}^t(u,v) = \sum_{i=1}^N w_i^t R_i^t(u,v)$$

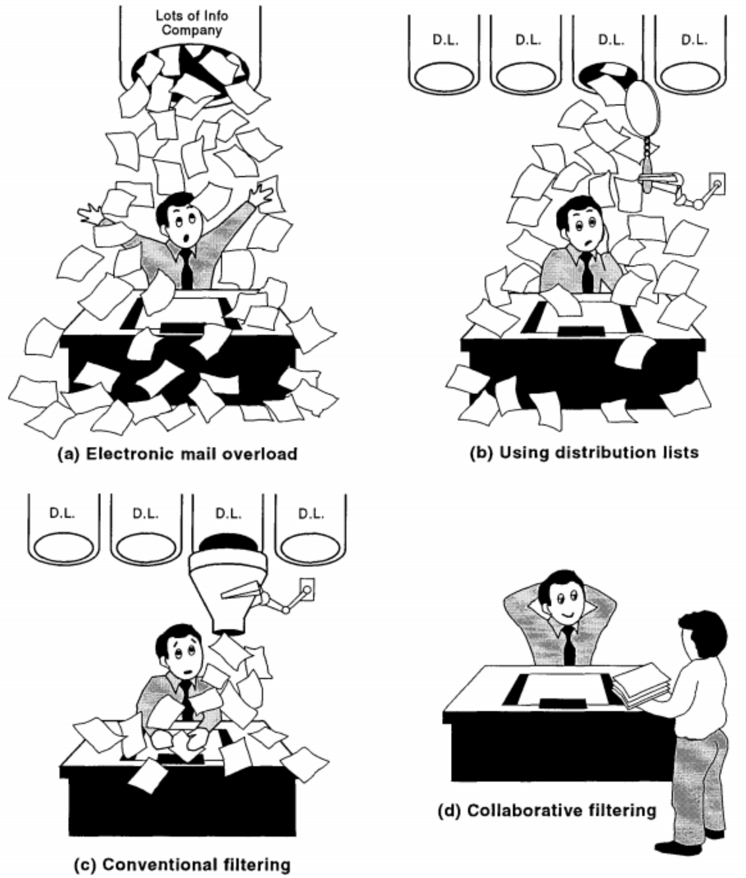
(Sec. 4 describes several ways of approximating a preference function with a total order.)

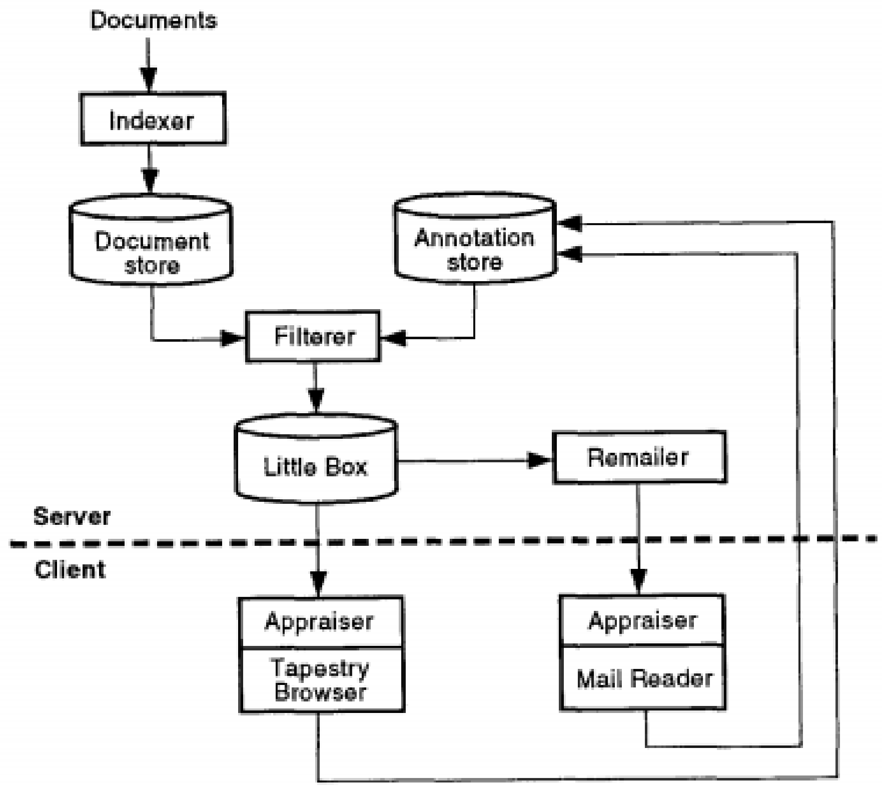
3. Order X^t using $\hat{\rho}^t$.
4. Receive feedback F^t from the user.
5. Evaluate losses $\mathrm{Loss}(R_i^t, F^t)$ as defined in Eq. (1).
6. Set the new weight vector

$$w_i^{t+1} = \frac{w_i^t \cdot \beta^{\mathrm{Loss}(R_i^t, F^t)}}{Z_t}$$

where Z_t is a normalization constant, chosen so that $\sum_{i=1}^N w_i^{t+1} = 1$.

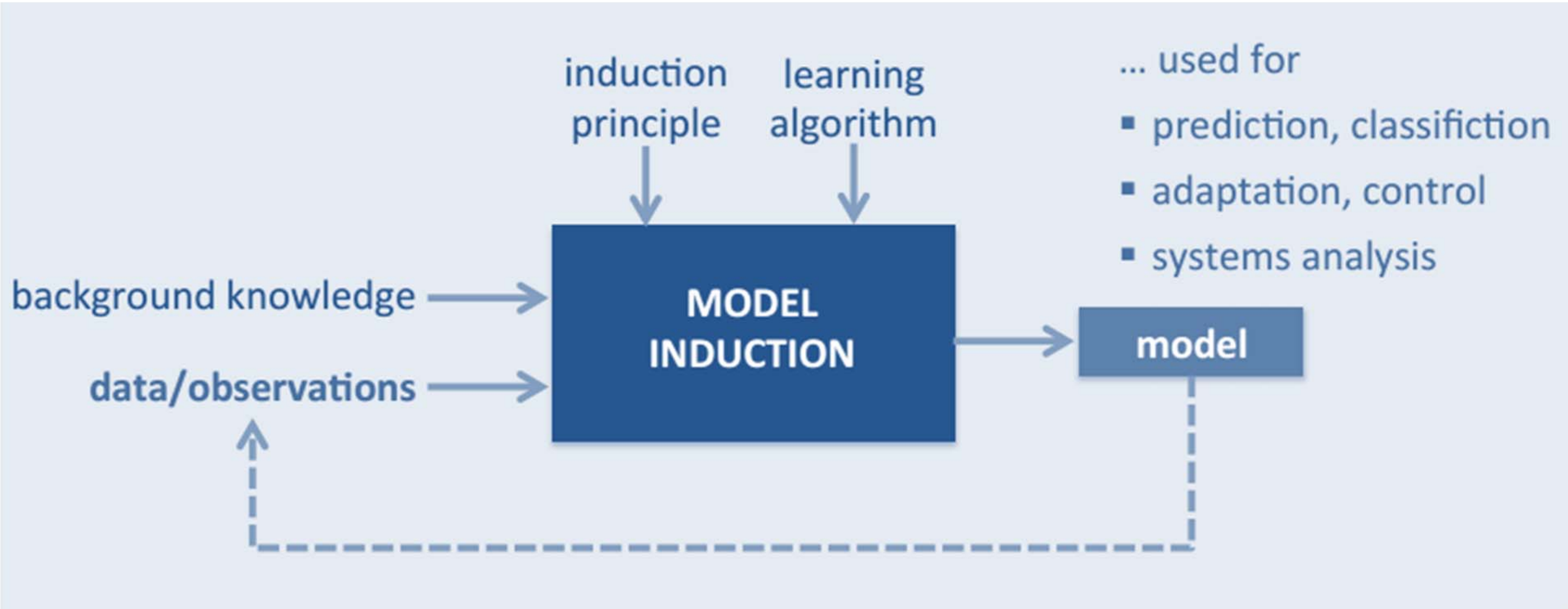
Cohen, W. W., Schapire, R. E. & Singer, Y. 1999. Learning to Order Things. Journal of Artificial Intelligence Research, 10, 243-270.





Goldberg, D., Nichols, D., Oki, B. M. & Terry, D. 1992. Using collaborative filtering to weave an information tapestry. Communications of the ACM, 35, (12), 61-70.

	OBJECT RANKING	COLLABORATIVE FILTERING
product description	features	identifier
preference description	relative	absolute
predictions	ranking	utility degrees
number of users/models	single	many



Fuernkranz, J. & Hüllermeier, E. 2010. Preference learning, Berlin Heidelberg, Springer.

- 1) Reduction to simpler problems
  - transform the problem to apply standard ML
- 2) Extension of classification algorithms
  - Generalization of standard ML – so to make them applicable to label ranking data
- 3) Probabilistic modeling and statistical inference
  - Using statistical models for ranking data and parameter estimation methods

Fuernkranz, J. & Hüllermeier, E. 2010. Preference learning, Berlin Heidelberg, Springer.

3) Active Preference Learning

- Previous work: Optimizing the coffee taste
  Herdy et al., 96
- Black box optimization:
- F : $\Omega \rightarrow R$ Find $\arg\max F$
- The user in the loop replaces F
- Optimizing visual rendering Brochu et al., 07
- Optimal recommender Viappiani & Boutilier, 10
- Information retrieval Shivaswamy & Joachims, 12

- Loop:

1. Algorithm presents an expert a pair of behaviours
2. Expert emits preferences y1 over y2
3. Algorithm learns expert’s utility function
4. Algorithm searches for behaviour with best utility

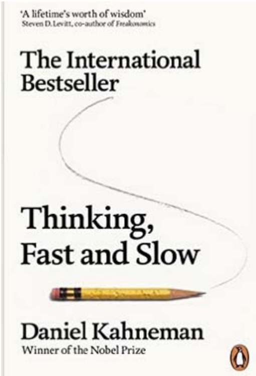
- Problem: Accounts for **human noise**

Schoenauer, M., Akrour, R., Sebag, M. & Souplet, J.-C. Programming by Feedback. Proceedings of the 31st International Conference on Machine Learning (ICML-14), 2014 Beijing. 1503-1511.

Akrour, R., Schoenauer, M. & Sebag, M. 2012. APRIL: Active Preference Learning-Based Reinforcement Learning. In: Flach, P. A., De Bie, T. & Cristianini, N. (eds.) Machine Learning and Knowledge Discovery in Databases, Lecture Notes in Computer Science LNCS 7524. Berlin Heidelberg: Springer, pp. 116-131.

Problem of a Human-in-the-loop

- Humans are irrational, inconsistent, lacking robustness, error-prone, adaptive, subjective, …
- Problem: Preferences often are biased, subjective, constructed on the fly, or even do not exist …
- (Daniel Kahnemann, Nobel-Prize 2002)



Kahneman, D. 2011. Thinking, fast and slow, New York, Macmillan.

Formal setting

$\mathcal{X}$ Search space, solution space — controllers, $\mathbb{R}^D$

$\mathcal{Y}$ Evaluation space, behavior space — trajectories, $\mathbb{R}^d$

$$\Phi : \mathcal{X} \mapsto \mathcal{Y}$$

Utility function

$$U^* \quad \mathcal{Y} \mapsto \mathbb{R} \quad U^*(\mathrm{y}) = \langle \mathbf{w}^*, \mathrm{y} \rangle$$

behavior space

Requisites

- Evaluation space: simple to learn from few queries
- Search space: sufficiently expressive

4) Multi-Task Learning

April 24–26, 2014
SIAM SDM14

Unterstützt von / Supported by
Alexander von Humboldt
Stiftung / Foundation

MAX-PLANCK-GESELLSCHAFT

MLCB

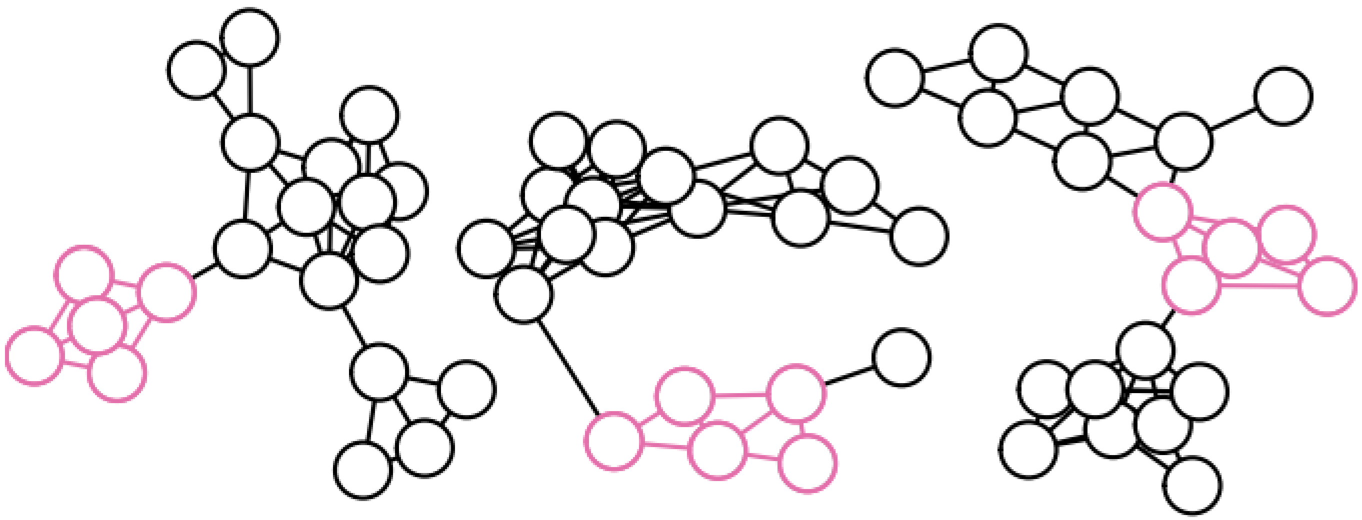
Multi-Task Feature Selection on Multiple Networks via Maximum Flows

Mahito Sugiyama[1 (,2)], Chloé-Agathe Azencott[3], Dominik Grimm[2,4], Yoshinobu Kawahara[1], Karsten Borgwardt[2,4]

[1]Osaka University, [2]Max Planck Institutes Tübingen, [3]Mines ParisTech, Institut Curie, INSERM, [4]Eberhard Karls Universität Tübingen

Sugiyama, M., Azencott, C.-A., Grimm, D., Kawahara, Y. & Borgwardt, K. M. Multi-Task Feature Selection on Multiple Networks via Maximum Flows. SDM, 2014. 199-207.

- Given multiple graphs
- Find features (=vertices), which are associated with the target response and tend to be connected to each other



Result: New formulation of MTF-Selection

$$\underbrace{\underset{S_1,\dots,S_K \subset V}{\operatorname{argmax}}}_{K \text{ tasks}} \sum_{i=1}^{K} \Big(\underbrace{f_i(S_i)}_{\text{association}} \underbrace{- g_i(S_i) \Big) - \sum_{i<j} h(S_i, S_j)}_{\text{penalty}},$$

$$f_i(S_i) := \sum_{v \in S_i} q_i(v), \quad g_i(S_i) := \lambda \underbrace{\sum_{e \in B_i} w_i(e)}_{\text{connectivity}} + \eta \underbrace{|S_i|}_{\text{sparsity}},$$

$$h(S_i, S_j) := \mu |S_i \,\Delta\, S_j| = \mu |(S \cup S') \setminus (S \cap S')|$$

- efficiently solved by max-flow algorithms
- performance is superior to Lasso-based methods

Sugiyama, M., Azencott, C.-A., Grimm, D., Kawahara, Y. & Borgwardt, K. M. Multi-Task Feature Selection on Multiple Networks via Maximum Flows. SDM, 2014. 199-207.

- Networks (graphs) are everywhere in health informatics
- Biological pathways (KEGG), chemical compounds, (PubChem), social networks, …
- Question often: Which part of the network is responsible for performing a particular function?
- → Feature selection on networks
- – Features = vertices (nodes)
- – Network topology = a priori knowledge of relationships between features
- **Multi-task feature selection should be considered for more effectiveness**

- Single task feature selection on a network
- Given a weighted graph $G = (V, E)$
- – Each $v \in V$ has a relevance score $q(v)$
- – If you have a design matrix $\mathbf{X} \in \mathbb{R}^{N \times |V|}$
- and a response vector $\mathbf{y} \in \mathbb{R}^{N}$, $q(v)$ is the association of $\mathbf{y}$ and each feature of $\mathbf{X}$

Goal: Find a subset $S \subset V$ which maximizes

$$f(S) := \sum_{v \in S} q(v)$$

while S is small and vertices are connected

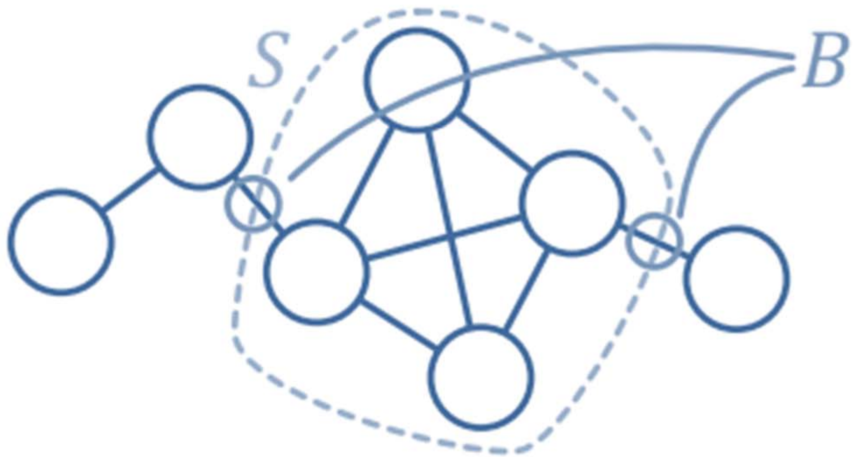
Azencott, C.-A., Grimm, D., Sugiyama, M., Kawahara, Y. & Borgwardt, K. M. 2013. Efficient network-guided multi-locus association mapping with graph cuts. Bioinformatics, 29, (13), i171-i179.

Formulation of SConES

- $\operatorname{argmax}_{S \subset V} f(S) - g(S)$

$$f(S) := \sum_{v \in S} q(v), \quad g(S) := \underbrace{\lambda \sum_{e \in B} w(e)}_{\text{connectivity}} + \underbrace{\eta |S|}_{\text{sparsity}}$$

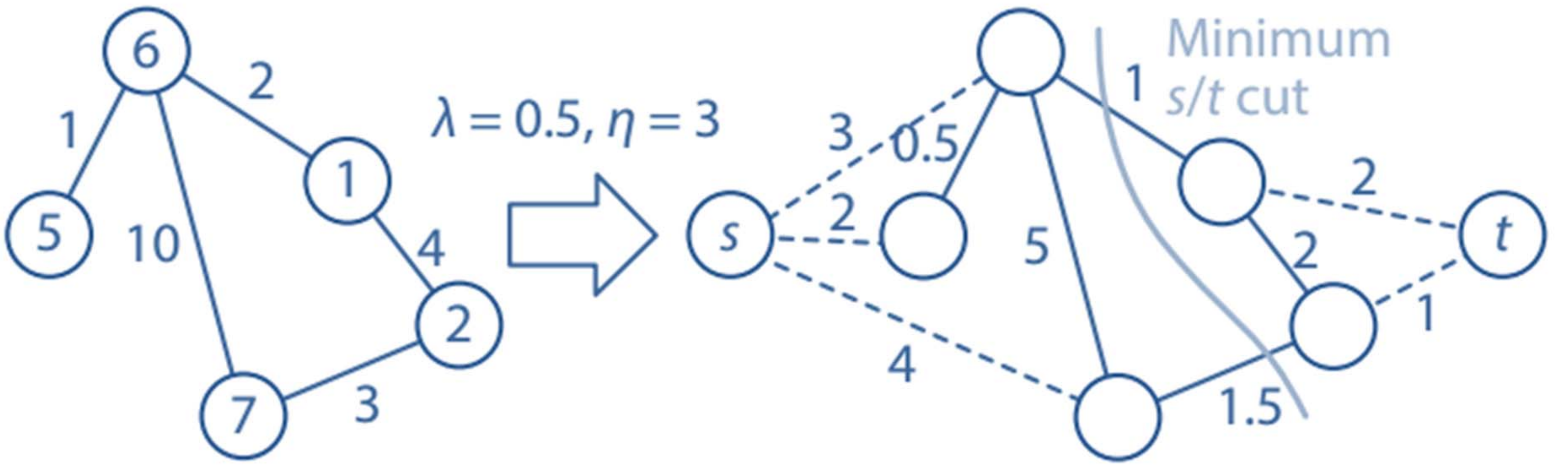
  - $B = \{\{v, u\} \in E \mid v \in V \setminus S,\ u \in S\}$ (boundary)
  - $w : E \rightarrow \mathbb{R}^+$ is a weighting function



Azencott, C.-A., Grimm, D., Sugiyama, M., Kawahara, Y. & Borgwardt, K. M. 2013. Efficient network-guided multi-locus association mapping with graph cuts. Bioinformatics, 29, (13), i171-i179.

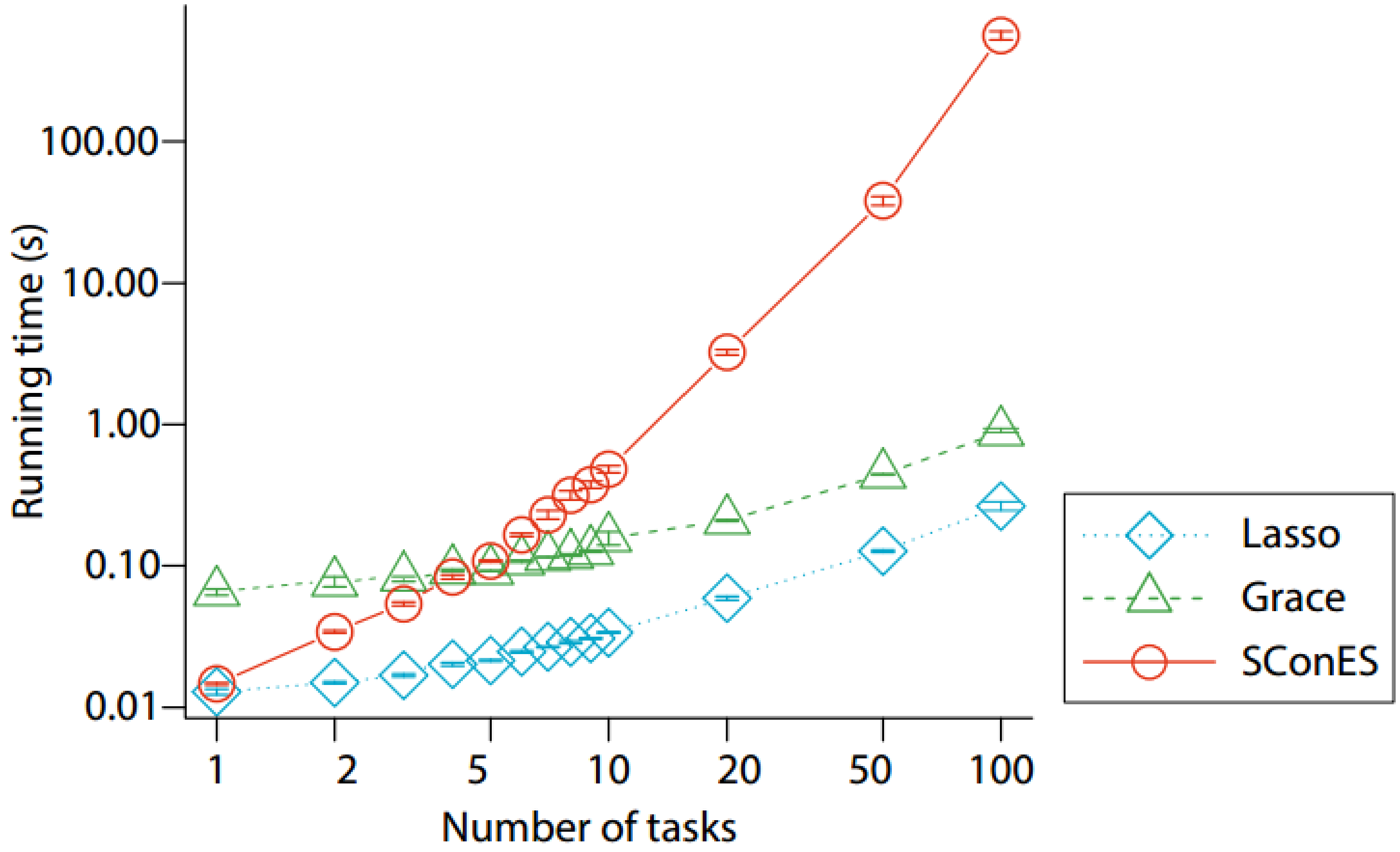
Solution of SConES via Maximum Flow

- The s/t-network $M(G) = (V \cup \{s,t\}, E \cup S \cup T)$ with $S = \{\{s,v\} \mid v \in V,\ q(v) > \eta\}$, $T = \{\{t,v\} \mid v \in V,\ q(v) < \eta\}$ and set the capacity $c : E' \to \mathbb{R}^+$ to
$$c(\{v,u\}) = \begin{cases} |\,q(u) - \eta\,| & \text{if } u \in \{s,t\} \text{ and } v \in V, \\ \lambda w(\{v,u\}) & \text{otherwise} \end{cases}$$
- The minimum s/t cut of $M(G)$ = the solution of SConES



Azencott, C.-A., Grimm, D., Sugiyama, M., Kawahara, Y. & Borgwardt, K. M. 2013. Efficient network-guided multi-locus association mapping with graph cuts. Bioinformatics, 29, (13), i171-i179.

Better performance

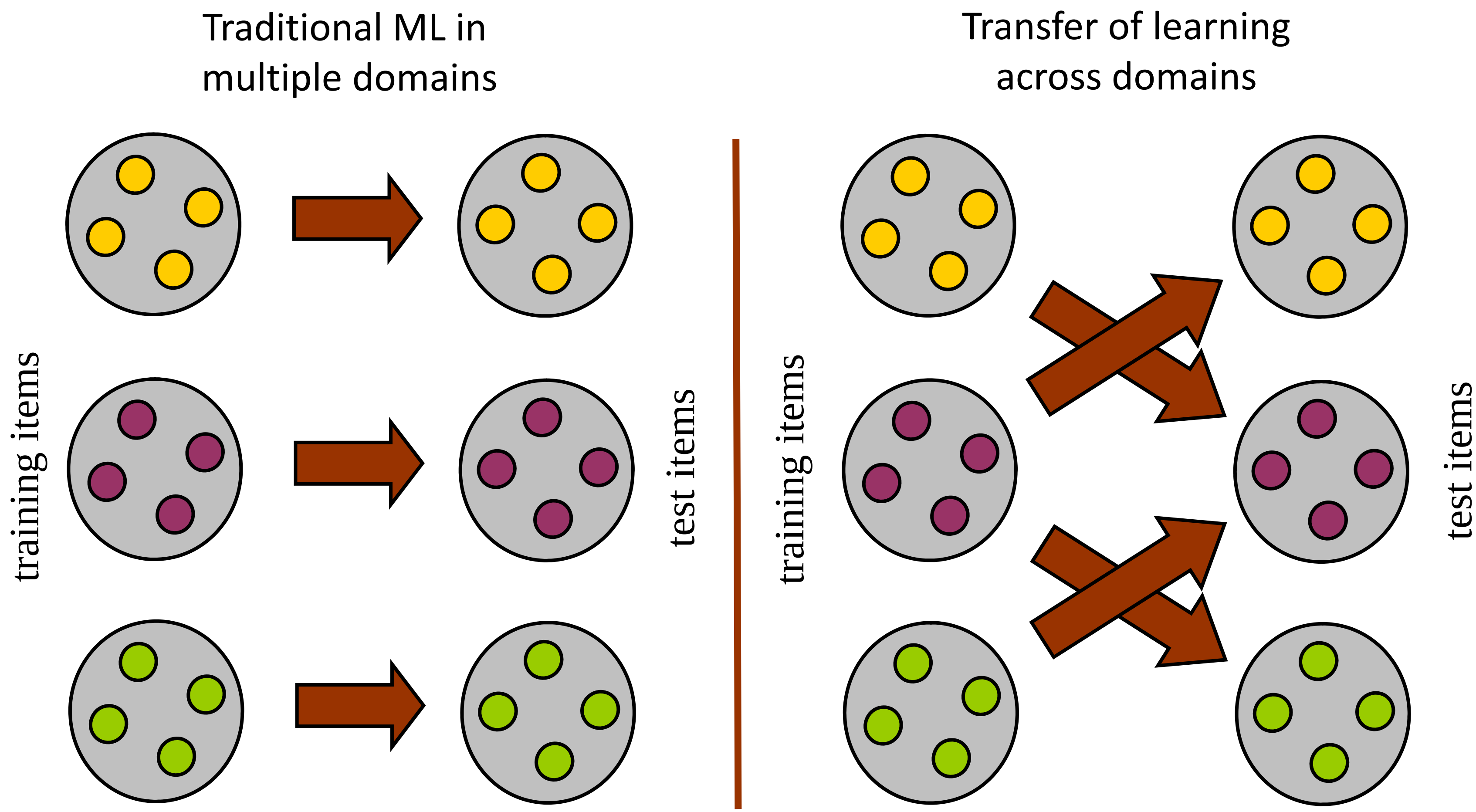


Azencott, C.-A., Grimm, D., Sugiyama, M., Kawahara, Y. & Borgwardt, K. M. 2013. Efficient network-guided multi-locus association mapping with graph cuts. Bioinformatics, 29, (13), i171-i179.

5) Transfer Learning

Transfer Learning is an old topic

- Learning or performance on prior experience
- Thorndike & Woodworth (1901) explored how individuals would transfer a knowledge in one context to another context
- context that share similar characteristics.
- C++ → Java
- Mathematics -> Computer Science
- Definition: Ability of a system to recognize and apply knowledge and skills learned in previous tasks to novel tasks or new domains, which share some commonalities
- Challenge: Given a target task: How to identify the commonality between the task and previous (source) tasks, and transfer knowledge from the previous tasks to the target one?

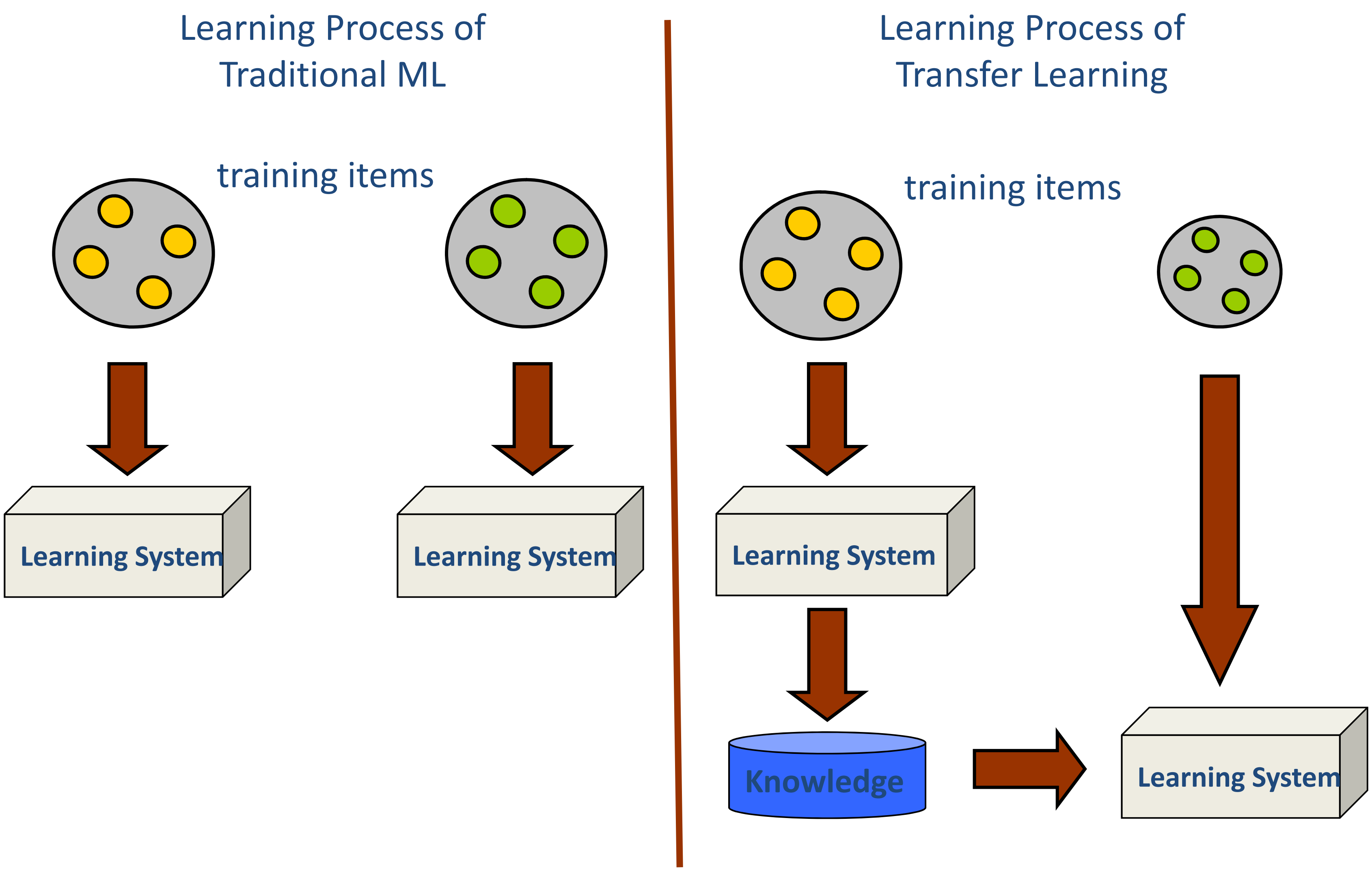


Humans can learn in many domains.

Humans can also transfer from one domain to other domains.

Pat Langley, 2006

What is the difference?



Note: Major Assumption

- Training and future (test) data come from a same task and a same domain.
- Represented in same feature and label spaces.
- Follow a same distribution.

Distinguish between Domain and Task

Domain:

It consists of two components: A feature space $\mathcal{X}$, a marginal distribution

$\mathcal{P}(X)$, where $X = \{x_1, x_2, ..., x_n\} \in \mathcal{X}$

In general, if two domains are different, then they may have different feature spaces or different marginal distributions.

Task:

Given a specific domain and label space $\mathcal{Y}$ for each x_i in the domain, to predict its corresponding label y_i, where $y_i \in \mathcal{Y}$

In general, if two tasks are different, then they may have different label spaces or different conditional distributions

$\mathcal{P}(Y|X)$, where $Y = \{y_1, ..., y_n\}$ and $y_i \in \mathcal{Y}$

- Source domain:

$$\mathcal{P}(X_S), \text{ where } X_S = \{x_{S_1}, x_{S_2}, ..., x_{S_{n_S}}\} \in \mathcal{X}_S$$

- Task in the source domain:

$$\mathcal{P}(Y_S|X_S), \text{ where } Y_S = \{y_{S_1}, y_{S_2}, ..., y_{S_{n_S}}\} \text{ and } y_{S_i} \in \mathcal{Y}_S$$

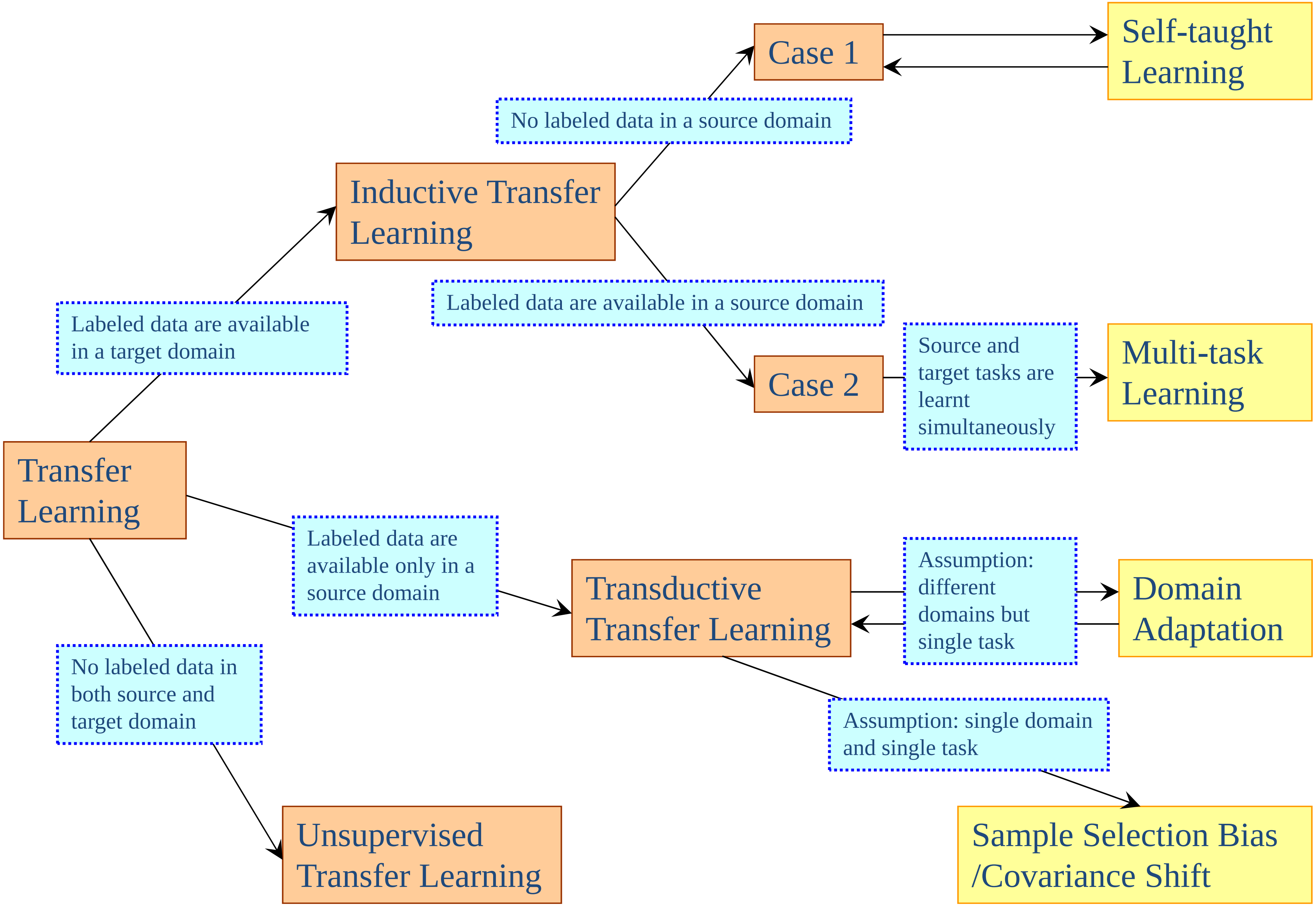
- Target domain:

$$\mathcal{P}(X_T), \text{ where } X_T = \{x_{T_1}, x_{T_2}, ..., x_{T_{n_T}}\} \in \mathcal{X}_T$$

- Task in the target domain

$$\mathcal{P}(Y_T|X_T), \text{ where } Y_T = \{y_{T_1}, y_{T_2}, ..., y_{T_{n_T}}\} \text{ and } y_{T_i} \in \mathcal{Y}_T$$

Pan, S. J. & Yang, Q. 2010. A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22, (10), 1345-1359.



Case 1
Self-taught Learning
No labeled data in a source domain
Inductive Transfer Learning
Labeled data are available in a source domain
Labeled data are available in a target domain
Case 2
Source and target tasks are learnt simultaneously
Multi-task Learning
Transfer Learning
Labeled data are available only in a source domain
Transductive Transfer Learning
Assumption: different domains but single task
Domain Adaptation
No labeled data in both source and target domain
Assumption: single domain and single task
Unsupervised Transfer Learning
Sample Selection Bias /Covariance Shift

4 different TL approaches

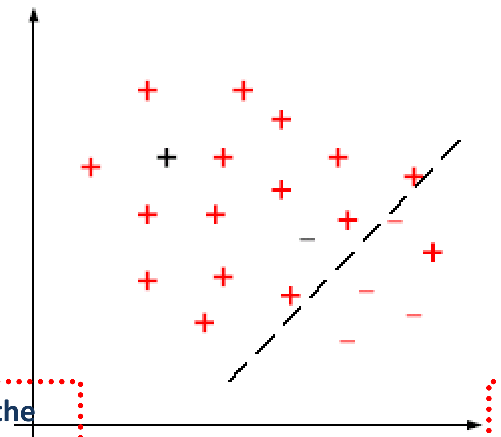
Transfer learning approaches	Description
Instance-transfer	*To re-weight some labeled data in a source domain for use in the target domain*
Feature-representation-transfer	Find a "good" feature representation that reduces difference between a source and a target domain or minimizes error of models
Model-transfer	Discover shared parameters or priors of models between a source domain and a target domain
Relational-knowledge-transfer	Build mapping of relational knowledge between a source domain and a target domain.

3 types of TL on the 4 TL approaches

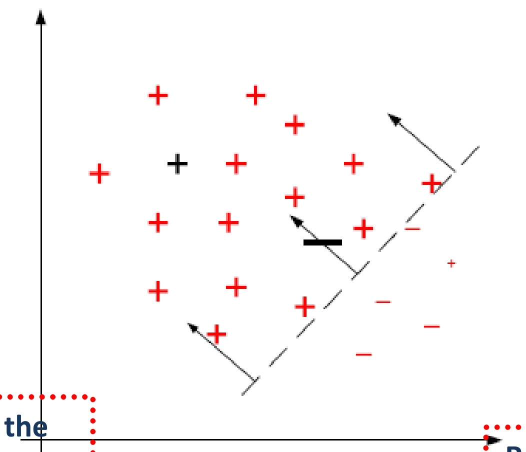
	Inductive Transfer Learning	Transductive Transfer Learning	Unsupervised Transfer Learning
Instance-transfer	☺	☺	
Feature-representation-transfer	☺	☺	☺
Model-transfer	☺		
Relational-knowledge-transfer	☺		

- Assumption: the source domain and target domain data use exactly the same features and labels.
- Motivation: Although the source domain data can not be reused directly, there are some parts of the data that can still be reused by re-weighting.
- Main Idea: Discriminatively adjust weighs of data in the source domain for use in the target domain.

Uniform weights



Correct the decision boundary by re-weighting



Loss function on the target domain data

Loss function on the source domain data

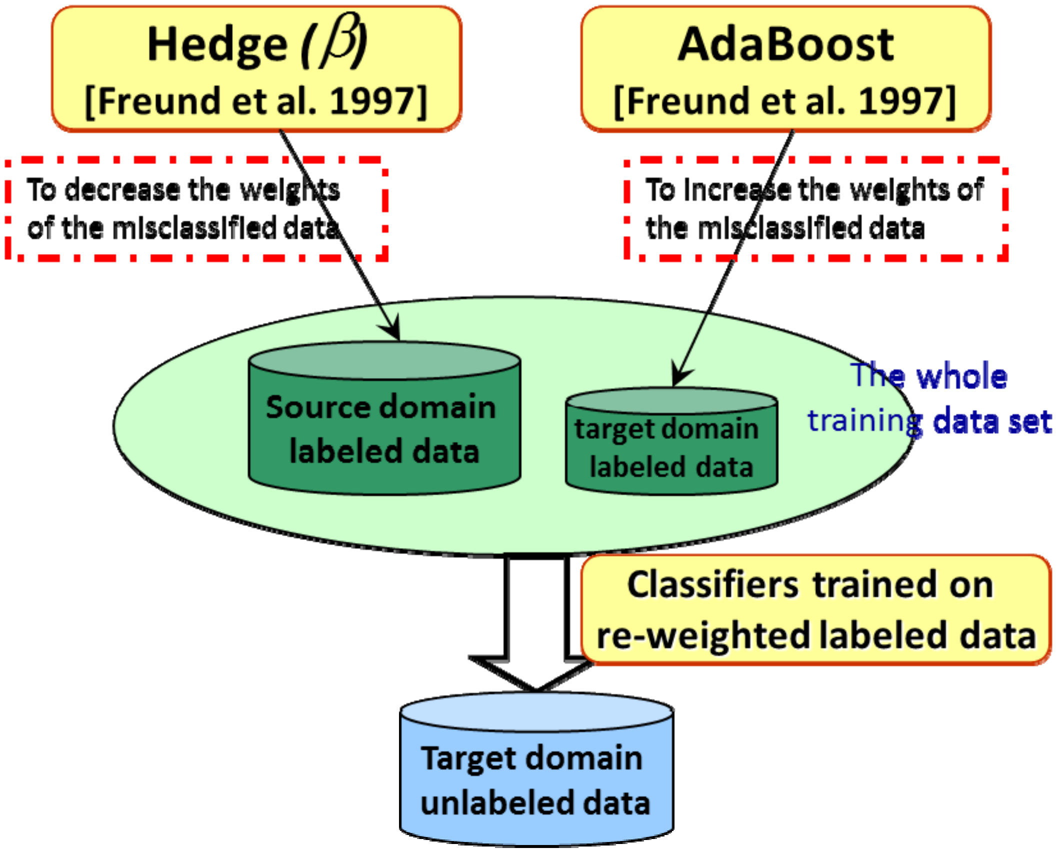
Regularization term

- Differentiate the cost for misclassification of the target and source data

$$J(h) = \sum_{i}^{n_T} L(h(x_{T_i}), y_{T_i}) + \lambda \sum_{j}^{n_S} L(h(x_{S_j}), y_{T_j}) + R(h)$$

Wu, P. & Dietterich, T. G. Improving SVM accuracy by training on auxiliary data sources. Proceedings of the twenty-first international conference on Machine learning, 2004. ACM, 110.

Inductive TL Instance transfer approach TrAdaBoost



Dai, W., Yang, Q., Xue, G.-R. & Yu, Y. Boosting for transfer learning. Proceedings of the 24th international conference on Machine learning, 2007. ACM, 193-200.

Algorithm 1 TrAdaBoost

Input the two labeled data sets T_d and T_s, the unlabeled data set S, a base learning algorithm **Learner**, and the maximum number of iterations N.

Initialize the initial weight vector, that $\mathbf{w}^1 = (w_1^1, \ldots, w_{n+m}^1)$. We allow the users to specify the initial values for $\mathbf{w}^1$.

For $t = 1, \ldots, N$

1. Set $\mathbf{p}^t = \mathbf{w}^t / (\sum_{i=1}^{n+m} w_i^t)$.
2. Call **Learner**, providing it the combined training set T with the distribution $\mathbf{p}^t$ over T and the unlabeled data set S. Then, get back a hypothesis $h_t : X \rightarrow Y$ (or $[0, 1]$ by confidence).
3. Calculate the error of h_t on T_s:
$$\epsilon_t = \sum_{i=n+1}^{n+m} \frac{w_i^t \cdot |h_t(x_i) - c(x_i)|}{\sum_{i=n+1}^{n+m} w_i^t}.$$
4. Set $\beta_t = \epsilon_t / (1 - \epsilon_t)$ and $\beta = 1/(1 + \sqrt{2 \ln n / N})$. Note that, ϵ_t is required to be less than $1/2$.
5. Update the new weight vector:
$$w_i^{t+1} = \begin{cases} w_i^t \beta^{|h_t(x_i) - c(x_i)|}, & 1 \leq i \leq n \\ w_i^t \beta_t^{-|h_t(x_i) - c(x_i)|}, & n+1 \leq i \leq n+m \end{cases}$$

Output the hypothesis

$$h_f(x) = \begin{cases} 1, & \prod_{t=\lceil N/2 \rceil}^{N} \beta_t^{-h_t(x)} \geq \prod_{t=\lceil N/2 \rceil}^{N} \beta_t^{-\frac{1}{2}} \\ 0, & \text{otherwise} \end{cases}$$

Self-Taught Learning

Self-taught Learning: Transfer Learning from Unlabeled Data

Rajat Raina RAJATR@CS.STANFORD.EDU
Alexis Battle AJBATTLE@CS.STANFORD.EDU
Honglak Lee HLLEE@CS.STANFORD.EDU
Benjamin Packer BPACKER@CS.STANFORD.EDU
Andrew Y. Ng ANG@CS.STANFORD.EDU
Computer Science Department, Stanford University, CA 94305 USA

Abstract

We present a new machine learning framework called "self-taught learning" for using unlabeled data in supervised classification tasks. We do not assume that the unlabeled data follows the same class labels or generative distribution as the labeled data. Thus, we would like to use a large number of unlabeled images (or audio samples, or text documents) randomly downloaded from the Internet to improve performance on a given image (or audio, or text) classification task. Such unlabeled data is significantly easier to obtain than in typical semi-supervised or transfer learning settings, making self-taught learning widely applicable to many practical learning problems. We describe an approach to self-taught learning that uses sparse coding to construct higher-level fea-

ately also provide the class labels.) This makes the classification task quite hard with existing algorithms for using labeled and unlabeled data, including most semi-supervised learning algorithms such as the one by Nigam et al. (2000). In this paper, we ask how unlabeled images from *other* object classes—which are much easier to obtain than images specifically of elephants and rhinos—can be used. For example, given unlimited access to unlabeled, randomly chosen images downloaded from the Internet (probably none of which contain elephants or rhinos), can we do better on the given supervised classification task?

Our approach is motivated by the observation that even many randomly downloaded images will contain basic visual patterns (such as edges) that are similar to those in images of elephants and rhinos. If, therefore, we can learn to recognize such patterns from the unlabeled data, these patterns can be used for the supervised learning task of interest, such as recognizing

Algorithm 1 Self-taught Learning via Sparse Coding

input Labeled training set
$T = \{(x_l^{(1)}, y^{(1)}), (x_l^{(2)}, y^{(2)}), \ldots, (x_l^{(m)}, y^{(m)})\}$.
Unlabeled data $\{x_u^{(1)}, x_u^{(2)}, \ldots, x_u^{(k)}\}$.
output Learned classifier for the classification task.
algorithm Using unlabeled data $\{x_u^{(i)}\}$, solve the optimization problem (1) to obtain bases b.
Compute features for the classification task to obtain a new labeled training set $\hat{T} = \{(\hat{a}(x_l^{(i)}), y^{(i)})\}_{i=1}^{m}$, where
$\hat{a}(x_l^{(i)}) = \arg\min_{a^{(i)}} \|x_l^{(i)} - \sum_j a_j^{(i)} b_j\|_2^2 + \beta \|a^{(i)}\|_1$.
Learn a classifier C by applying a supervised learning algorithm (e.g., SVM) to the labeled training set $\hat{T}$.
return the learned classifier C.

Step1:

$$\min_{a,b} \sum_i \|x_{S_i} - \sum_j a_{S_i}^j b_j\|_2^2 + \beta \|a_{S_i}\|_1$$

$$s.t. \quad \|b_j\|_2 \leq 1, \forall j \in 1, \ldots, s$$

Input: Source domain data $X_S = \{x_{S_i}\}$ and coefficient β

Output: New representations of the source domain data $A_S = \{a_{S_i}\}$ and new bases $B = \{b_i\}$

Step2:

$$a_{T_i}^* = \arg\min_{a_{T_i}} \|x_{T_i} - \sum_j a_{T_i}^j b_j\|_2^2 + \beta \|a_{T_i}\|_1$$

Input: Target domain data $X_T = \{x_{T_i}\}$ coefficient β and bases $B = \{b_i\}$

Output: New representations of the target domain data $A_T = \{a_{T_i}\}$

Raina, R., Battle, A., Lee, H., Packer, B. & Ng, A. Y. Self-taught learning: transfer learning from unlabeled data. Proceedings of the 24th international conference on Machine learning, 2007. ACM, 759-766.

Inductive Transfer Learning (Model transfer approach)

Assumption: If t tasks are related to each other, then they may share some parameters among individual models.

Assume $f_t = w_t \cdot x$ be a hyper-plane for task , where $t \in \{T, S\}$ and

$$w_S = w_0 + v_S \qquad w_T = w_0 + v_T$$

Common part

Specific part for individual task

Regularization terms for multiple tasks

Encode them into SVMs:

$$\min_{w_0, v_t, \xi_{t_i}} \left\{ J(w_0, v_t, \xi_{t_i}) = \sum_{t \in \{S,T\}} \sum_{i=1}^{n_t} \xi_{t_i} + \frac{\lambda_1}{2} \sum_{t \in \{S,T\}} \|v_t\|^2 + \lambda_2 \|w_0\|^2 \right\}$$

$$s.t. \quad y_{t_i}(w_0 + v_t) \cdot x_{t_i} \geq 1 - \xi_{t_i}, \; \xi_{t_i} \geq 0, \; i \in \{1, 2, ..., n_t\} \; and \; t \in \{S, T\}$$

Evgeniou, T. & Pontil, M. Regularized multi-task learning. Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, 2004. ACM, 109-117.

- Motivation: If two domains are related to each other, then there may exist some "pivot" features across both domain.
- Pivot features are features that behave in the same way for discriminative learning in both domains.
- Main Idea: To identify correspondences among features from different domains by modeling their correlations with pivot features.
- Non-pivot features form different domains that are correlated with many of the same pivot features are assumed to correspond, and they are treated similarly in a discriminative learner.

Input: labeled source data $\{(\mathbf{x}_t, y_t)_{t=1}^{T}\}$,
unlabeled data from both domains $\{\mathbf{x}_j\}$

Output: predictor $f : X \to Y$

1. Choose m pivot features. Create m binary prediction problems, $p_\ell(\mathbf{x})$, $\ell = 1 \dots m$

2. For $\ell = 1$ to m
$$\hat{\mathbf{w}}_\ell = \underset{\mathbf{w}}{\operatorname{argmin}} \left(\sum_j L(\mathbf{w} \cdot \mathbf{x}_j, p_\ell(\mathbf{x}_j)) + \lambda ||\mathbf{w}||^2 \right)$$
end

3. $W = [\hat{\mathbf{w}}_1 | \dots | \hat{\mathbf{w}}_m]$, $[U \; D \; V^T] = \mathrm{SVD}(W)$, $\theta = U^T_{[1:h,:]}$

4. Return f, a predictor trained on $\left\{ \left(\begin{bmatrix} \mathbf{x}_t \\ \theta \mathbf{x}_i \end{bmatrix}, y_t \right)_{t=1}^{T} \right\}$

a) Heuristically choose m pivot features, which is task specific.

b) Transform each vector of pivot feature to a vector of binary values and then create corresponding prediction problem.

Learn parameters of each prediction problem

Do Eigen Decomposition on the matrix of parameters and learn the linear mapping function.

Use the learnt mapping function to construct new features and train classifiers onto the new representations.

Conclusion

	Inductive Transfer Learning	Transductive Transfer Learning	Unsupervised Transfer Learning
Instance-transfer	√	√	
Feature-representation-transfer	√	√	√
Model-transfer	√		
Relational-knowledge-transfer	√		

How to avoid negative transfer need to be attracted more attention!

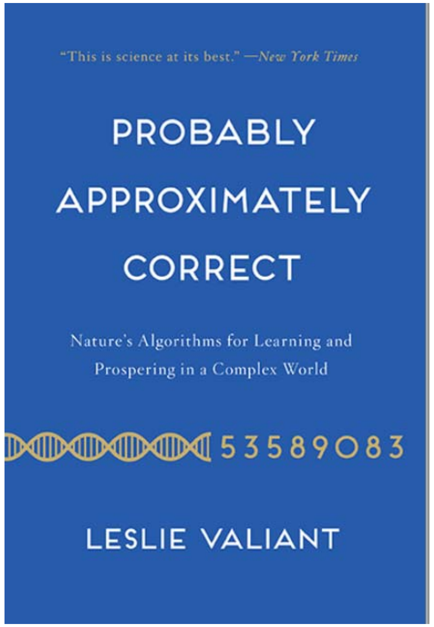
Big Problem: How to avoid negative transfer?

Thank you!

Sample Questions

- What is Active Learning?
- Where are the advantages of AL?
- Describe a few scenarios for AL?
- How does the robot scientist by King et al (2004) work?
- What does "Probable Approximate Correct" mean?
- What is the basic assumption of PL?
- What is the core essence of the "programming by feedback" approach?
- What could be huge disadvantages with the "human-in-the-loop"?
- What is a utility function?
- Why is multi-task learning of extreme importance for future research?
- When are humans better in TL ?
- Explain the 3 types of TL and the 4 TL approaches!
- What is the main idea of inductive TL?

Recommended Book



Valiant, L. 2013. Probably Approximately Correct: Nature's Algorithms for Learning and Prospering in a Complex World, New York, Basic Books.

http://people.seas.harvard.edu/~valiant/

Solutions to the Quiz

- ad 1) the typical ML-tasks: right=class prediction supervised learning; left=class discovery, unsupervised learning;
- ad 2) Bird flocking behaviour is a good example for evolutionary computing; the simple rules of birds are: Separation - avoid crowding neighbors; alignment - towards average heading of neighbors, and cohesion - steer towards average position of neighbors;
- ad 3) Experiment by Wilson et al. (2015) participants were asked to extrapolate from several functions, where the true underlying relationships were draws from a Gaussian process with a rational quadratic kernel
- ad 4) MAB problem models an agent that simultaneously attempts to acquire new knowledge (called "exploration") and optimize the decisions based on existing knowledge (called "exploitation"). The agent attempts to balance the competing tasks in order to maximize a value over time; this is very important e.g. for clinical trials investigating the effects of different experimental treatments whilst min. patient loss
- ad 5) Ants are food foraging, algorithmically this can be used as probabilistic method to find optimal paths through graphs
- ad 6) TSP appears as NP-hard problem in many domains, e.g. DNA, protein folding, etc.
- ad 7) similarity is an important concept and similarity learning is a type of supervised learning related to regression and classification – the goal is to learn a similarity function from examples (very important in recommender systems).