Chapter 8. Classification: Ensembles

Meng Jiang

Data Science

Today's Lecture

- Describe ensemble methods
  - Bagging: Random Forest (Bagged Decision Trees)
  - Boosting: AdaBoost (Adaptive Boosting)

Turing Award Recipients

2007	Edmund M. Clarke, E. Allen Emerson and Joseph Sifakis	For their roles in developing model checking into a **highly effective verification technology**, widely adopted in the hardware and software industries…
2008	Barbara Liskov	**Programming language and system design**…
2009	Charles P. Thacker	**Xerox Alto, the 1st modern PC, Ethernet and Tablet PC**….
2010	Leslie G. Valiant	Theory of computation…
2011	Judea Pearl	**Artificial intelligence** through the development of a calculus for **probabilistic and causal reasoning**…
2012	Silvio Micali Shafi Goldwasser	Transformative work that laid the complexity-theoretic foundations for the science of **cryptography**…
2013	Leslie Lamport	Contributed to **distributed and concurrent systems**…
2014	Michael Stonebraker	Concepts underlying **modern database systems**…
2015	Martin E. Hellman Whitfield Diffie	Introduced **public-key cryptography**, the foundation for the most regularly-used security protocols on the Internet…
2016	Tim Berners-Lee	Invented the **World Wide Web** and the **first web browser**…

Marvin Minsky

- Marvin Lee Minsky (August 9, 1927 – January 24, 2016) was an American cognitive scientist concerned largely with research of **artificial intelligence** (AI), co-founder of the Massachusetts Institute of Technology's AI laboratory, and author of several texts concerning AI and philosophy.
- Awards
  - **Turing Award (1969)**
  - Japan Prize (1990)
  - IJCAI Award for Research Excellence (1991)
  - Benjamin Franklin Medal (2001)
  - Computer History Museum Fellow (2006)
  - BBVA Foundation Frontiers of Knowledge Award (2013)

Logical vs. Analogical
or
Symbolic vs. Connectionist
or
Neat vs. Scruffy

https://web.media.mit.edu/~minsky/papers/SymbolicVs.Connectionist.html

Marvin Minsky

In Artificial Intelligence at MIT, Expanding Frontiers, Patrick H. Winston (Ed.), Vol.1, MIT Press, 1990. Reprinted in AI Magazine, Summer 1991.

Ensembles

"To solve really hard problems, we'll have to use *several different representations*…

It is time to *stop arguing* over *which* type of pattern-classification technique *is best*…

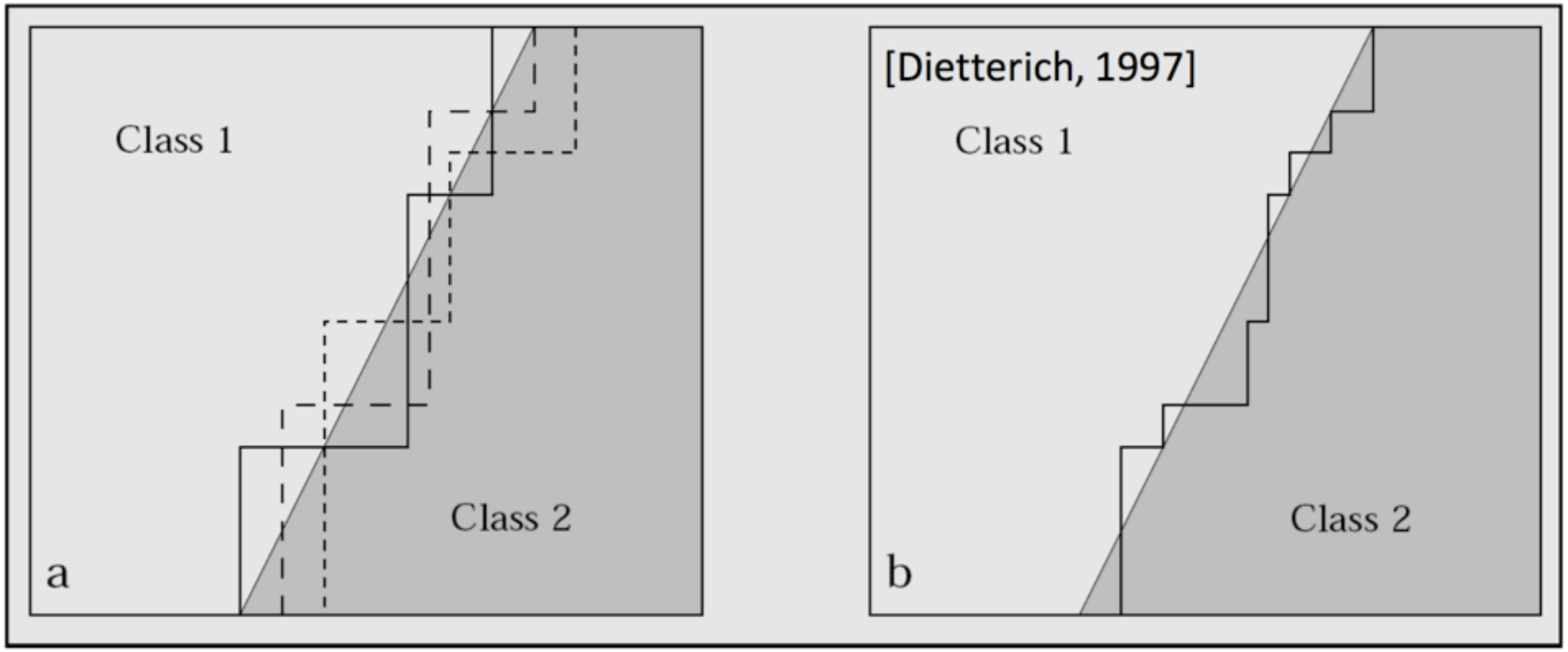
Instead we should work at a higher level of organization and discover how to build *managerial systems* to exploit the *different virtues* and evade the *different limitations* of each of these ways of comparing things." [**Minsky, 1991**]

Ensembles (cont.)

- An ensemble is a set of classifiers that learn a target function, and their *individual predictions are combined* (weighted or un-weighted) to classify new examples
  - Each classifier should be *more accurate than by chance*, and independent of one another.
  - Usually *more accurate than a single classifier*.
- Ensembles generally improve the *generalization performance* of a set of classifiers on a domain.

Ensemble Methods

- Ensemble methods
  - Use a *combination of models* to ***increase accuracy***
  - Combine a series of *k* learned models, M_1, M_2, …, M_k, with the aim of creating an *improved model* M*



Ensemble Methods (cont.)

- Popular ensemble methods
  - **Bagging:** averaging the prediction over a collection of classifiers
  - **Boosting:** weighted vote with a collection of classifiers
    - **AdaBoost (Adaptive Boosting):** adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers.

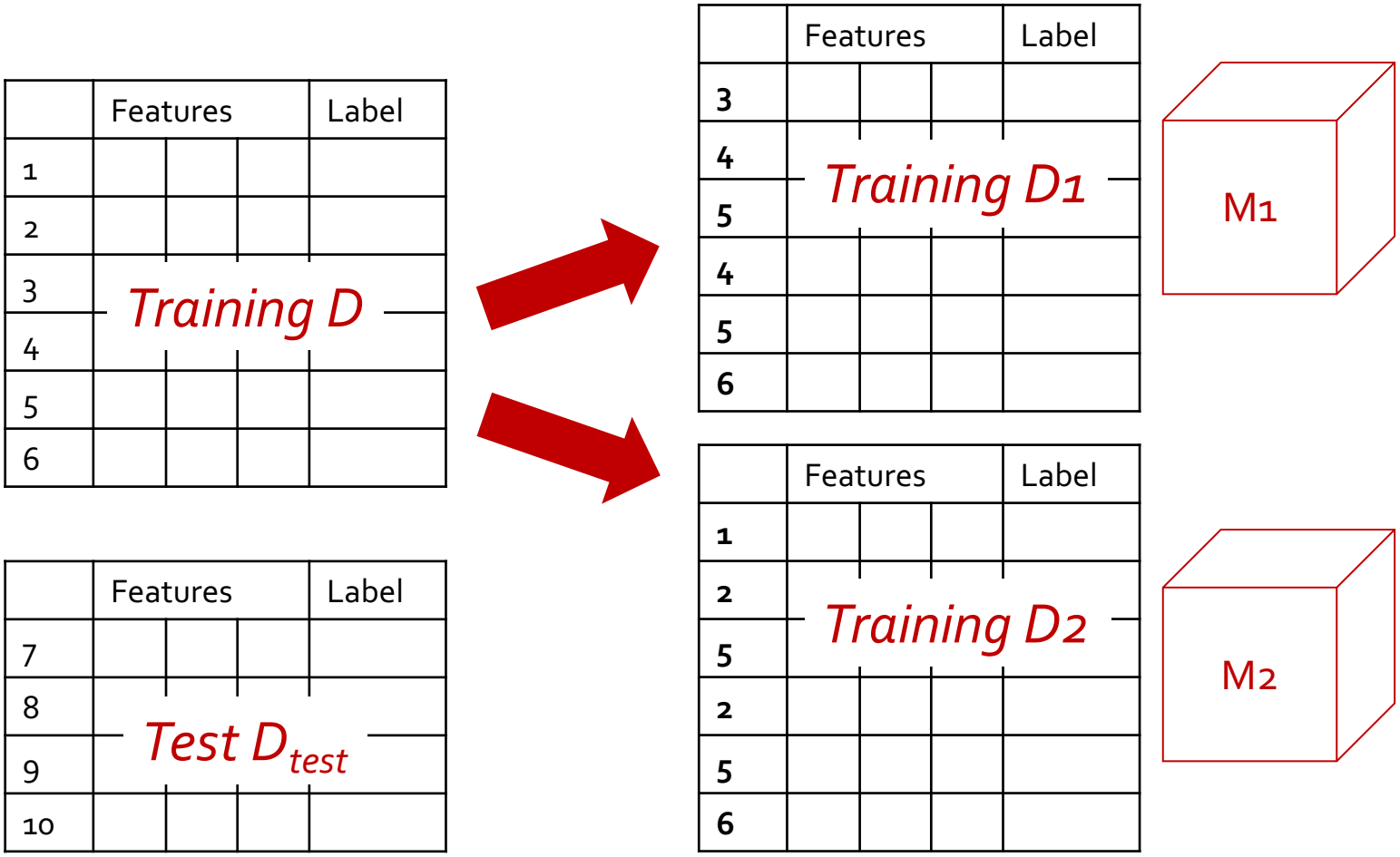
Bagging

- Training
  - Given a data set D of *d* instances, a classifier model M_i is learned for a training set D_i of *d* instances that is *sampled with replacement* from D (i = 1…k)
  - As a result of the *sampling-with-replacement* procedure, each classifier is trained on approximately *63.2%* of the training examples
  - For a dataset with *d* instances, each instance has a probability of $1 - (1 - 1/d)^d$ of being selected at least once in the *d* samples.
    - For $d \rightarrow \infty$, this number converges to $(1 - 1/e)$ or 0.632 [Bauer and Kohavi, 1999]

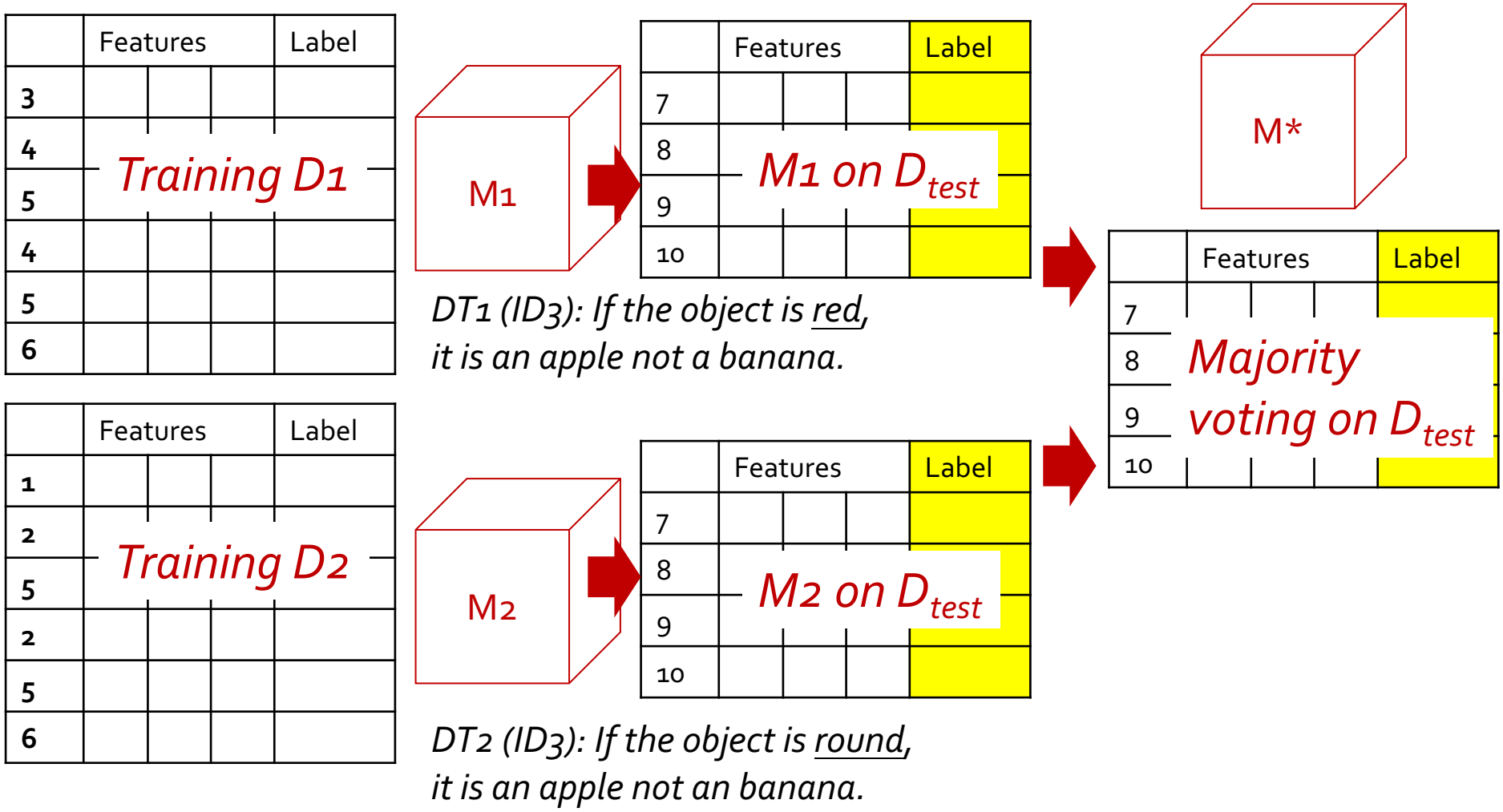
Bagging (cont.)

- Classification: classify an unknown sample **X**
  - Each classifier M_i returns its class prediction
  - The bagged classifier M* *counts the votes* and assigns the class with the *most votes* to **X**
- Accuracy: Proved improved accuracy in prediction
  - *Often significantly better* than a single classifier derived from D

Bagging (cont.)



Bagging (cont.)



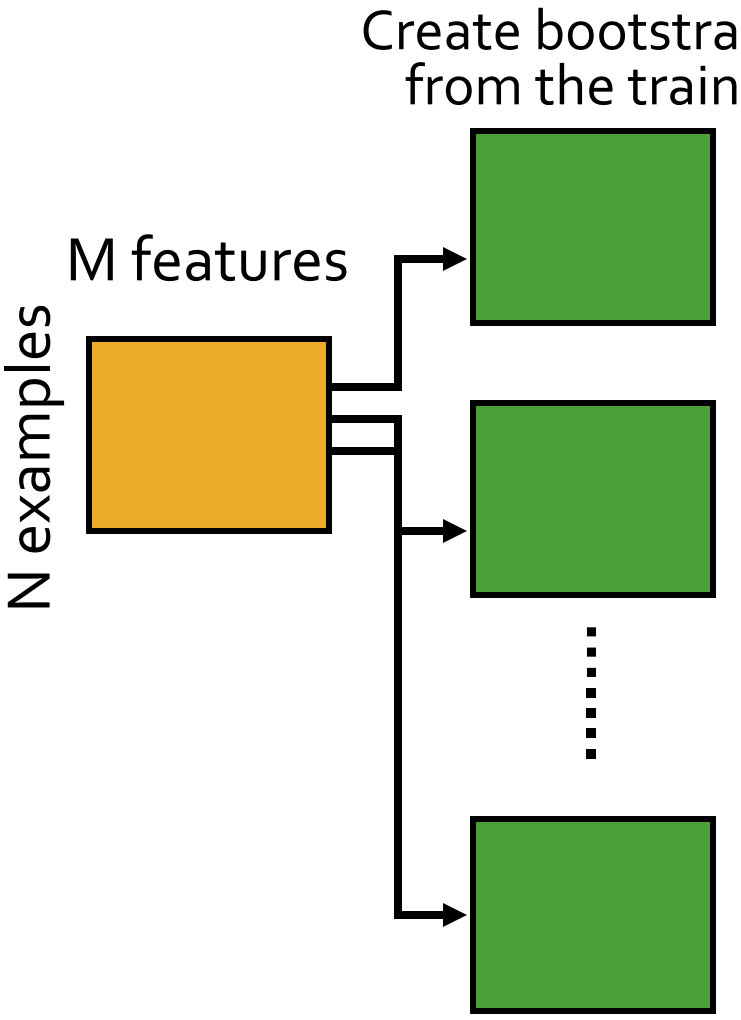
Random Forest model: Decision Trees + Bagging

- **Random forest** (or random forests) is an **ensemble** classifier that consists of **many decision trees** and outputs the class that is the mode of the class's output by individual trees.
- The term came from random decision forests that was first proposed by **Tin Kam Ho** of **Bell Labs** in 1995.
- The method combines Breiman's "**bagging**" idea and the random selection of features.

Bagging in Random Forest

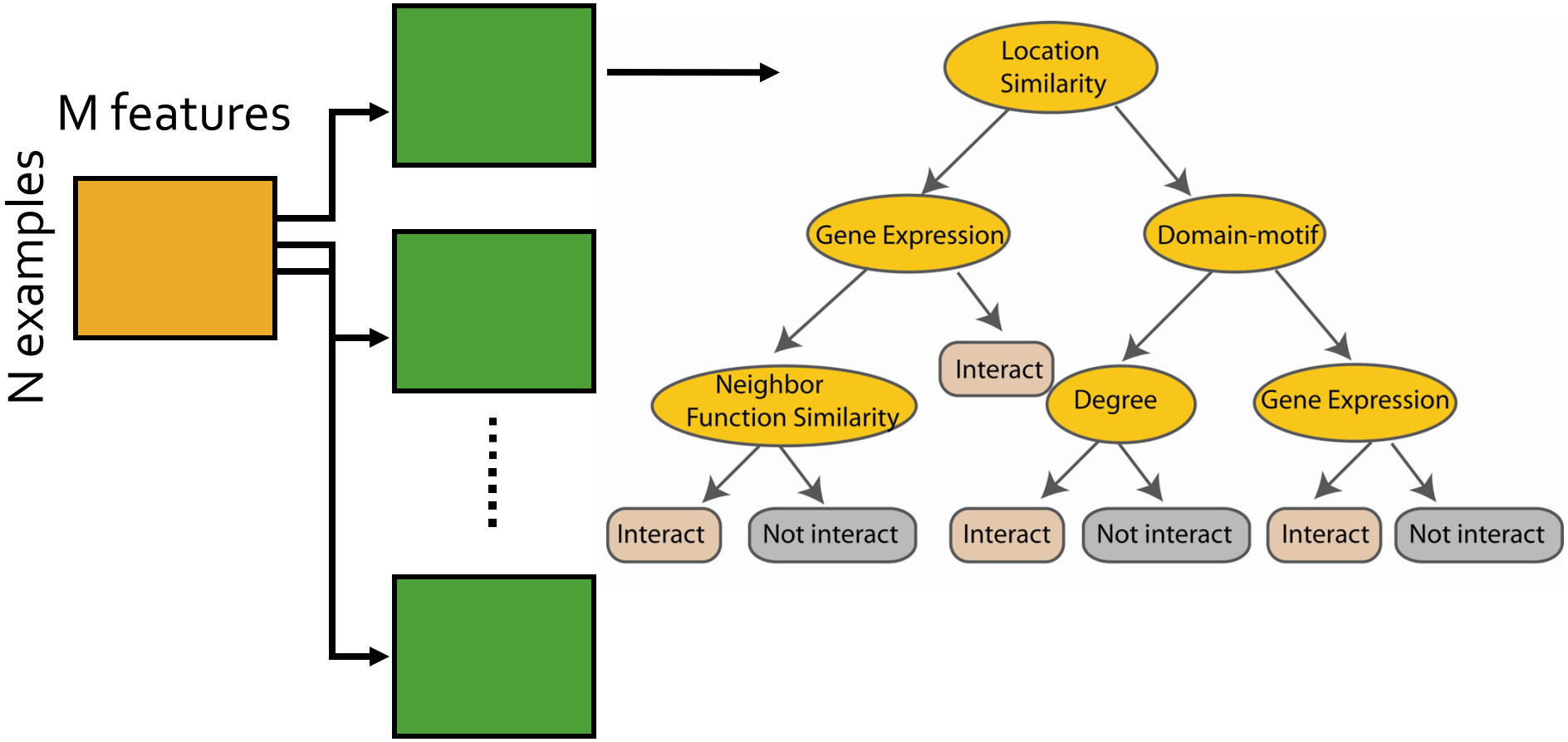
- Bagging or *bootstrap aggregation* a technique for reducing the variance of an estimated prediction function.
- For classification, a **committee** of trees each cast a **vote** for the predicted class.

Bagging: Bootstrap Samples

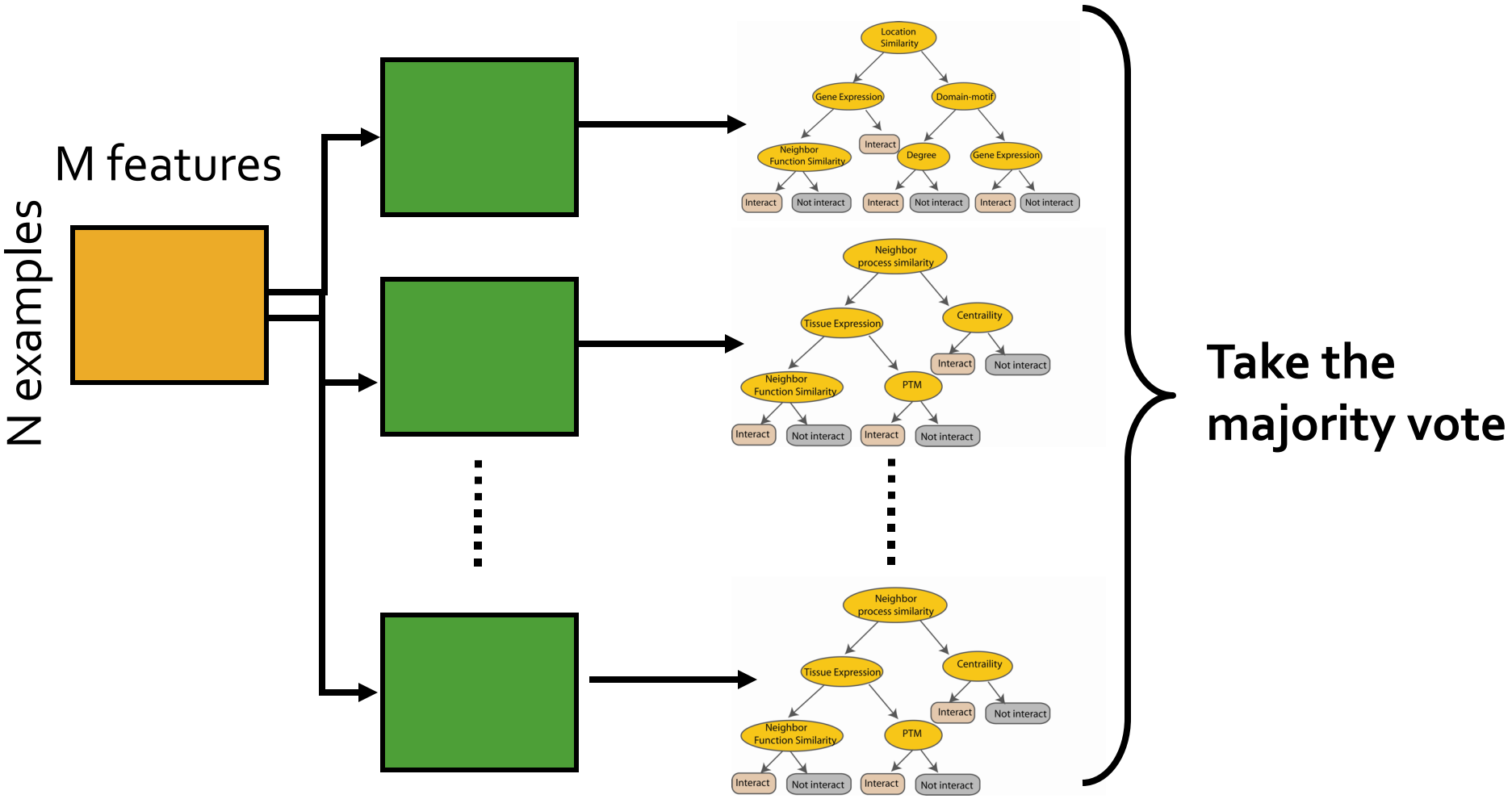


Bagging: Decision Tree

Construct a decision tree



Bagging: Random Forest



Bagging: Simulated Example

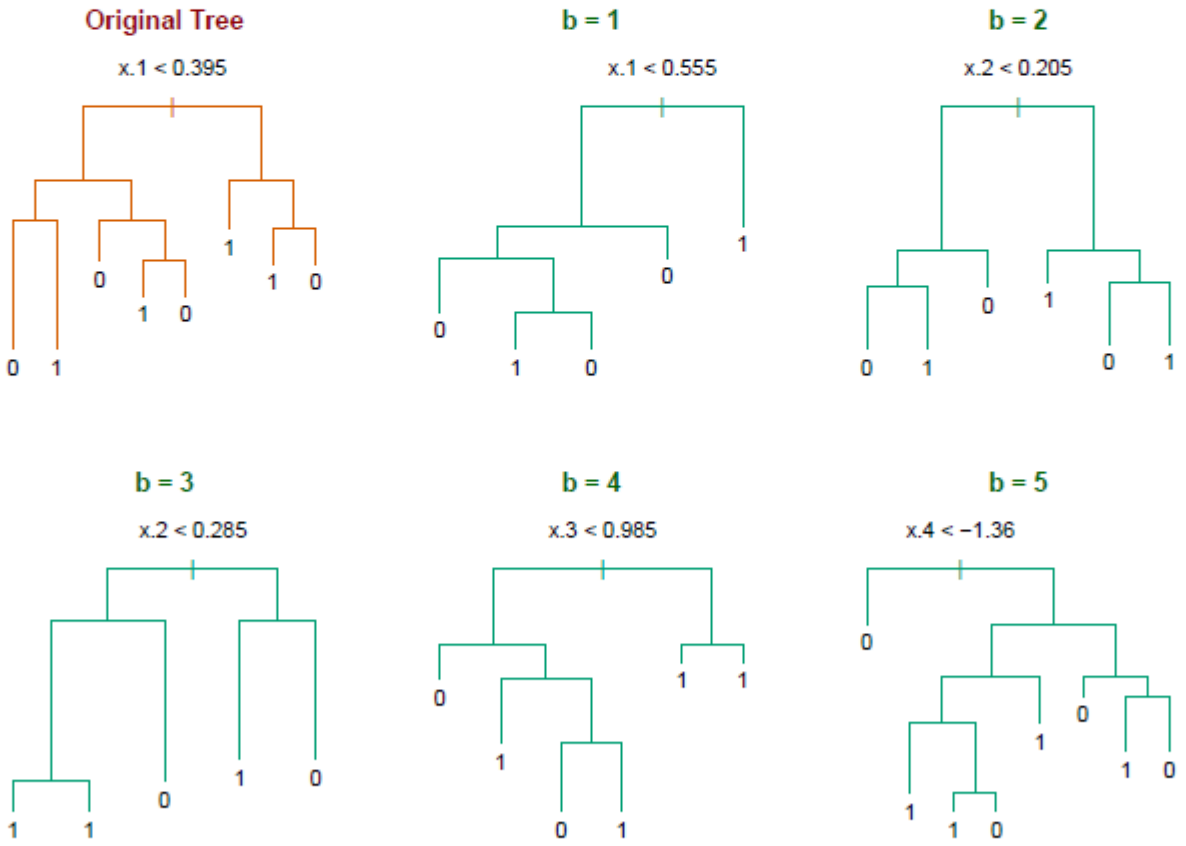
Generated a sample of size $N = 30$, with two classes and $p = 5$ features, each having a standard Gaussian distribution with pairwise correlation 0.95.

The response Y was generated according to

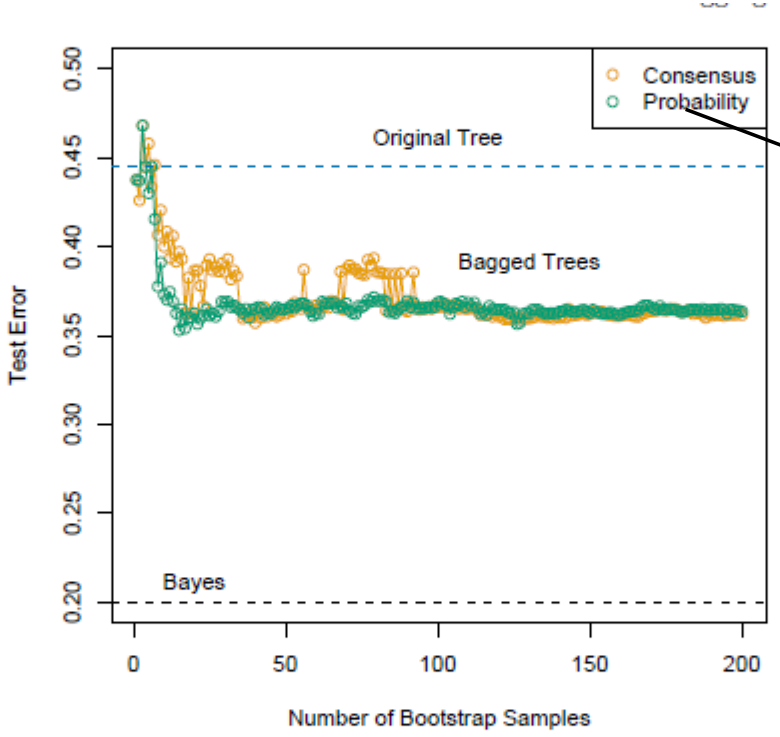
- $\Pr(Y = 1|X1 \leq 0.5) = 0.2$,
- $\Pr(Y = 0|X1 > 0.5) = 0.8$.

Bagging Trees on the Example

Notice the bootstrap trees are different than the original tree



Bagging



Treat the voting Proportions as probabilities

FIGURE 8.10. *Error curves for the bagging example of Figure 8.9. Shown is the test error of the original tree and bagged trees as a function of the number of bootstrap samples. The orange points correspond to the consensus vote, while the green points average the probabilities.*

bagging helps under squared-error loss, in short because averaging reduces

http://www-stat.stanford.edu/~hastie/Papers/ESLII.pdf Example 8.7.1

Boosting

- Training
  - *Weights* are assigned to each training instance
  - A series of *k* classifiers is *iteratively* learned
  - After a classifier M_i is learned, the *weights* are updated to allow the subsequent classifier, M_{i+1}, to pay more attention to the *training* instances that were *misclassified* by M_i
- Classification
  - The final M* *combines the votes* of each individual classifier, where the *weight* of each classifier's vote is a function of its *accuracy* on classifying training instances
- Comparing with Bagging: Boosting tends to have *greater accuracy*, but it risks *overfitting* the model to misclassified data

	Features			Label	Weight
1					1.0
2					1.0
3					1.0
4					1.0
5					1.0
6					1.0

Training D

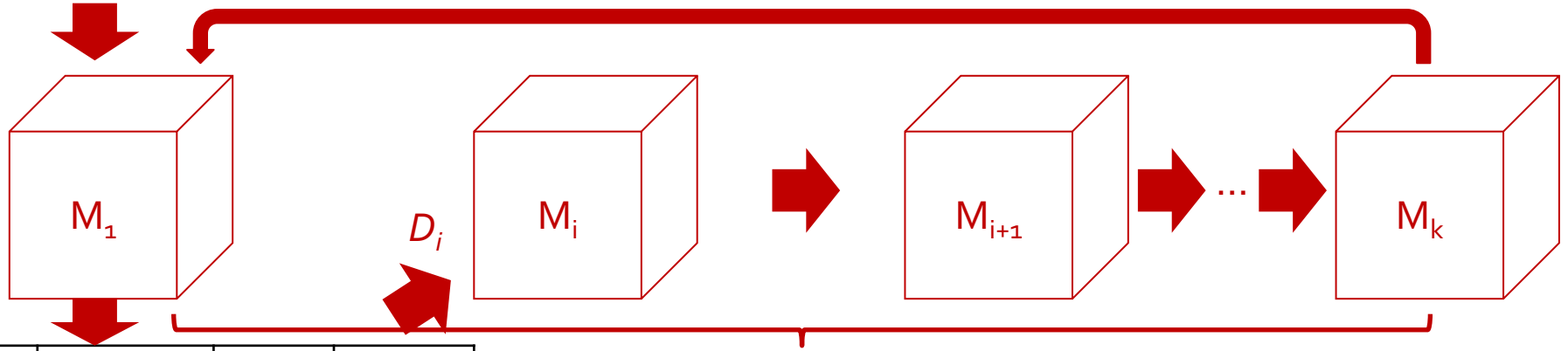
	Features			Label
3				
4				
5				
4				
5				
6				

Training D1

	Features			Label
3				
...				
...				...

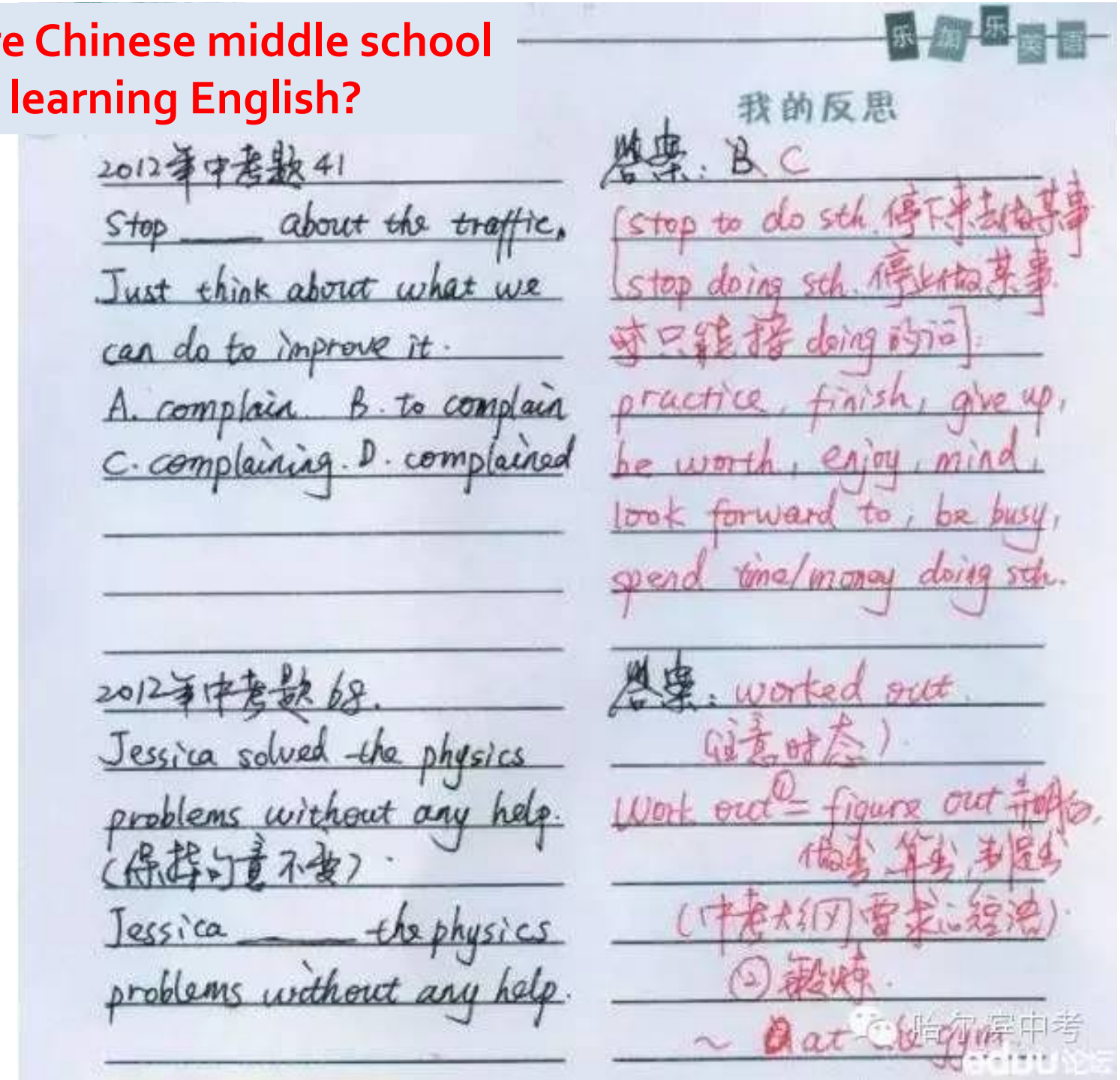
Training D_i

Iteratively



	Predicted	Label	Weight
3	Yes	No	3.0
4	Yes	Yes	1.0

How were Chinese middle school students learning English?



我的反思

2012年中考题41

Stop ____ about the traffic. Just think about what we can do to improve it.

A. complain B. to complain C. complaining D. complained

答案：B C

[stop to do sth. 停下来去做某事
stop doing sth. 停止做某事
时只能接 doing 的话]：

practice, finish, give up, be worth, enjoy, mind, look forward to, be busy, spend time/money doing sth.

2012年中考题68.

Jessica solved the physics problems without any help.
(保持句意不变)

Jessica ____ the physics problems without any help.

答案：worked out

(注意时态)

Work out ① = figure out 计算出，
得出，算出，判定
(中考经常要求短语)

② 锻炼

~ at 在……

AdaBoost (Adaptive Boosting)

- Given a set of *d* class-labeled instances, ($\mathbf{X_1}$, y_1), …, ($\mathbf{X_d}$, y_d)
- Initially, all the *weights* of instances are set the same (1/*d*)
- Generate *k* classifiers in *k* rounds. At round *i*,
  - Instances from D are *sampled with replacement* to form a training set D_i of the same size
  - Each instance's chance of being selected is based on its *weight*
  - A classification model M_i is derived from D_i
  - Its *error rate* is calculated *using D_i as a "test set"*
  - If an instance is misclassified, its *weight* is increased, otherwise it is decreased

AdaBoost (cont.)

- *Error rate:* *err($\mathbf{X}_j$)* is the misclassification error of instance $\mathbf{X}_j$. Classifier M_i's error rate is the sum of the weights of the misclassified instances:

$$error(M_i) = \sum_{j}^{d} w_j \times err(\mathbf{X_j})$$

- The *weight* of classifier M_i's vote is

$$\log \frac{1 - error(M_i)}{error(M_i)}$$

AdaBoost (cont.)

	Features			Label	Weight
1					1.0
2					1.0
3					1.0
4					1.0
5					1.0
6					1.0

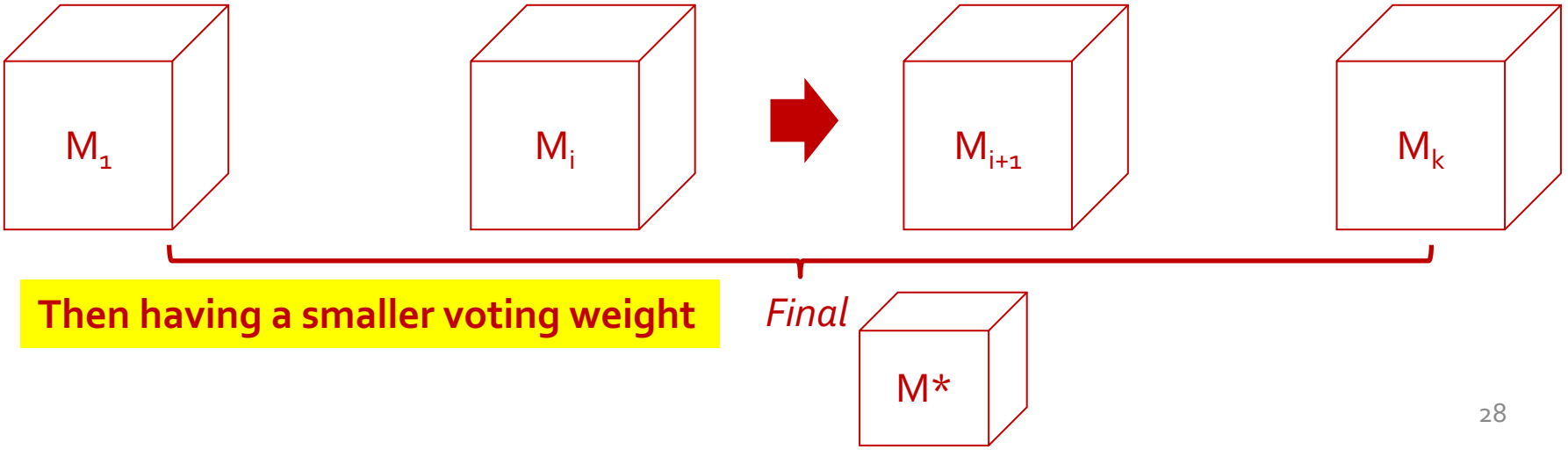
Training D

If being misclassified

	Features			Label	Weight
					3.0
					2.1
3					0.5
4					1.5
5					1.0
6					1.0

Then higher chance of being sampled

If having a higher error rate



Then having a smaller voting weight

Is  a fish?

Summary

- Describe ensemble methods
  - Bagging: Random Forest (Bagged Decision Trees)
  - Boosting: AdaBoost (Adaptive Boosting)

Philosophies behind ensemble methods!

An Example

"SetExpan: Corpus-Based Set Expansion via Context Feature Selection and Rank Ensemble"

by Shen et al. at UIUC

Macedonia


France
Switzerland
Austria
Budapest
Hungary
Moldova
Chisinau
Odessa
Одеса
Milan
Slovenia
Zagreb
Romania
Belgrade
Београд
Croatia
Bucharest
Bosnia and
Herzegovina
Serbia
Monaco
Sarajevo
Sofia
София
Black Sea
Italy
Andorra
Podgorica
Подгорица
Kosovo
Bulgaria
Barcelona
Rome
Macedonia
(FYROM)
Tirana
Albania
Istanbul
Bursa
Ankara
Tyrrhenian Sea
Greece
Turkey
Athens
Αθήνα
Izmir
Algiers
مدينة الجزائر
Tunis
تونس
Antalya
Adana
Malta

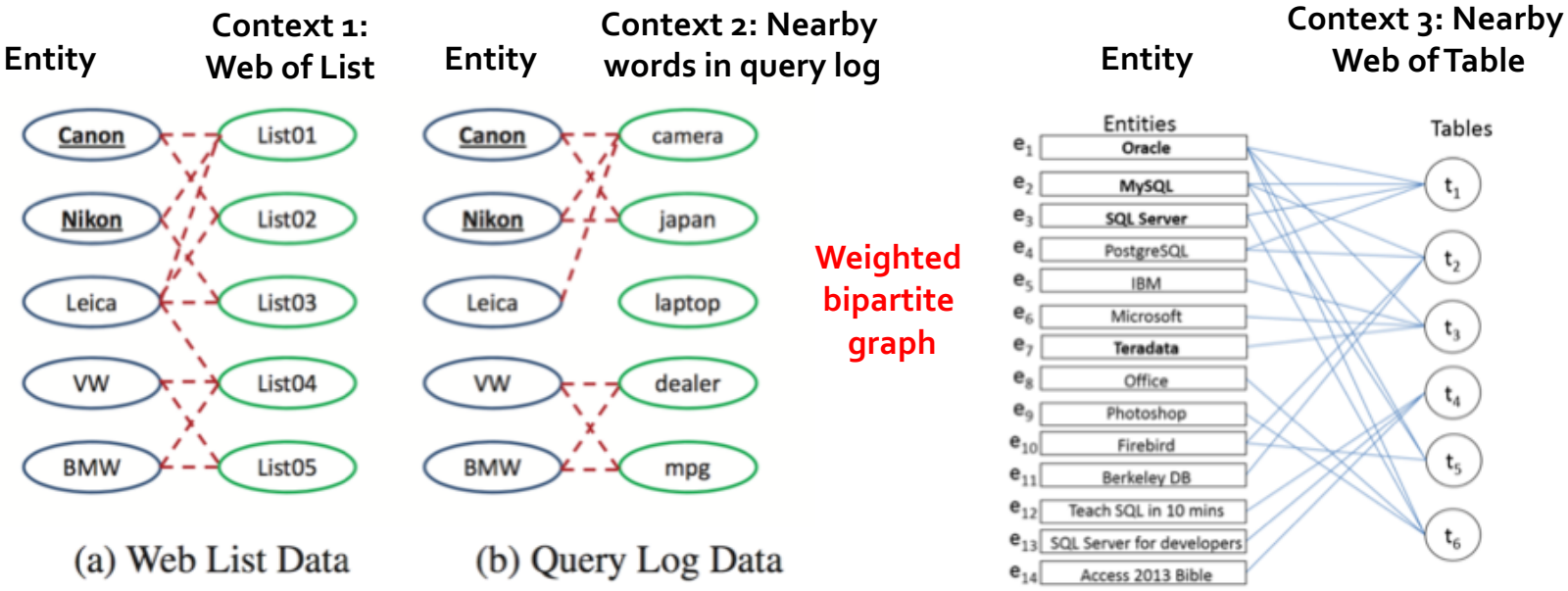
Skopje

Problem Definition

- Given a raw corpus and several seed entities for multiple semantic classes, we aim to find a set of "sibling" entities for each class such that the entities and seeds belong to the same class.
- Examples:
  - Given Wikipedia corpus, Class 1: {Texas, Illinois}, Class 2: {Quebec, Ontario} -> Class 1: all US states, Class 2: all Canada provinces.
  - Givens PubMed corpus, Class 1: {hypertension, heart disease}, Class 2: {neuroblastoma, astrocytoma} -> Class 1: all cardiovascular diseases, Class 2: all cancers.

Existing Work: Structured Context Feature

- Find **structured** context features for each entity



What if we don't have structured context features?

SEISA -- Set Expansion by Iterative Similarity Aggregation, WWW 2011

Concept Expansion Using Web Tables, WWW 2015

Existing Work: Unstructured Context Feature (1)

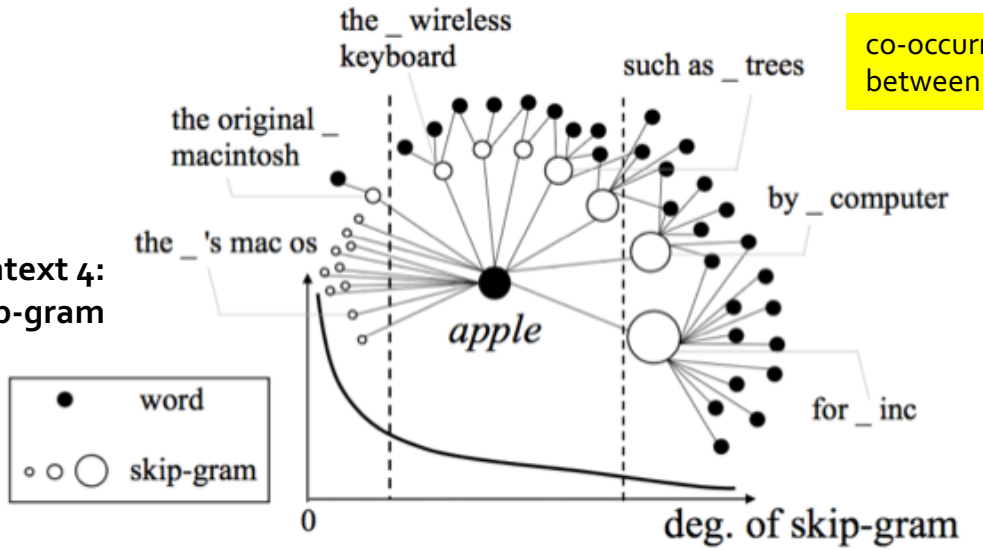
- Find **unstructured** context features for each entity

$$f_{w,s} = \log(1 + X_{w,s})[\log|W| - \log(\sum_{w} X_{w,s})]$$

co-occurrence count between w and s

total number of words in the vocabulary

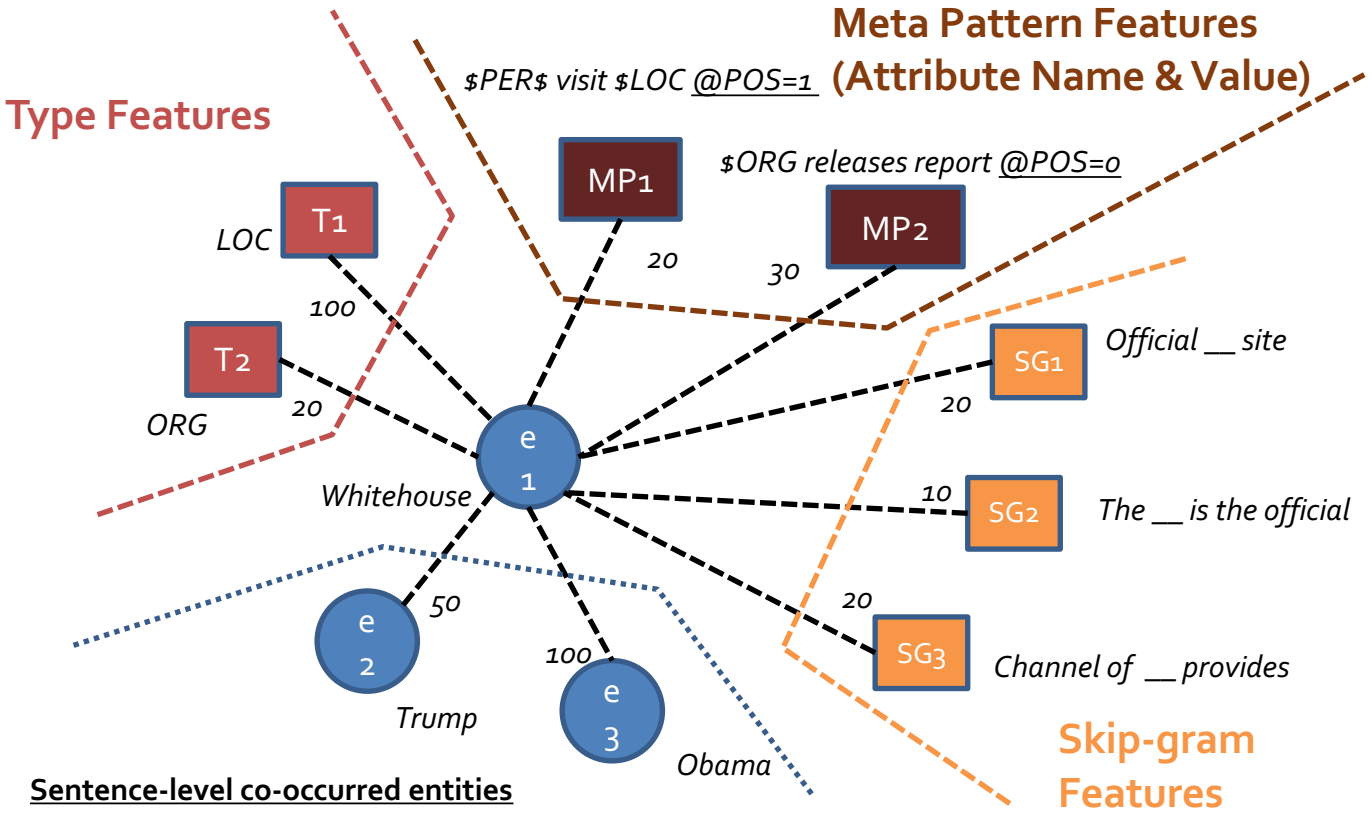
Context 4: Skip-gram



Equivalent to previous bipartite model

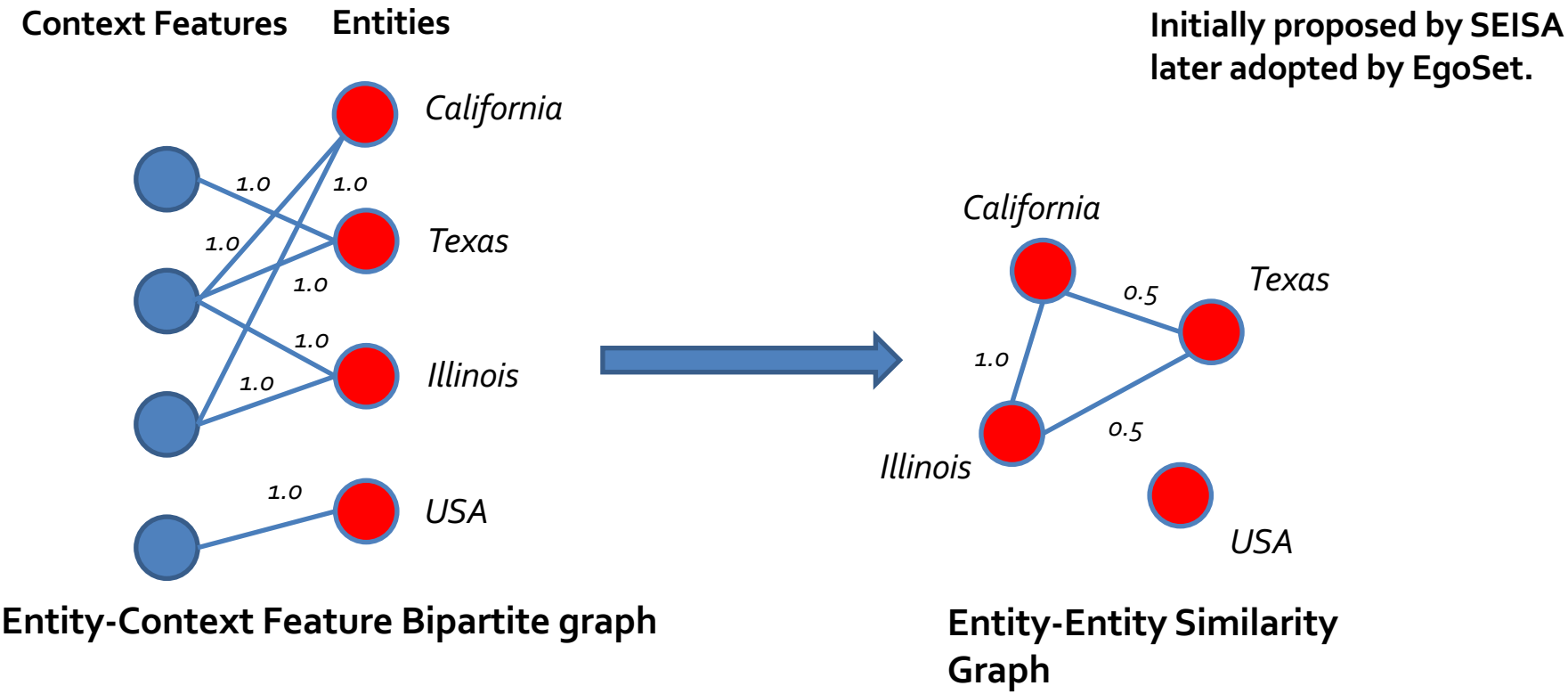
EgoSet: Exploiting Word Ego-networks and User-generated Ontology for Multifaceted Set Expansion, WSDM 2016

Existing Work: Unstructured Context Feature (2)



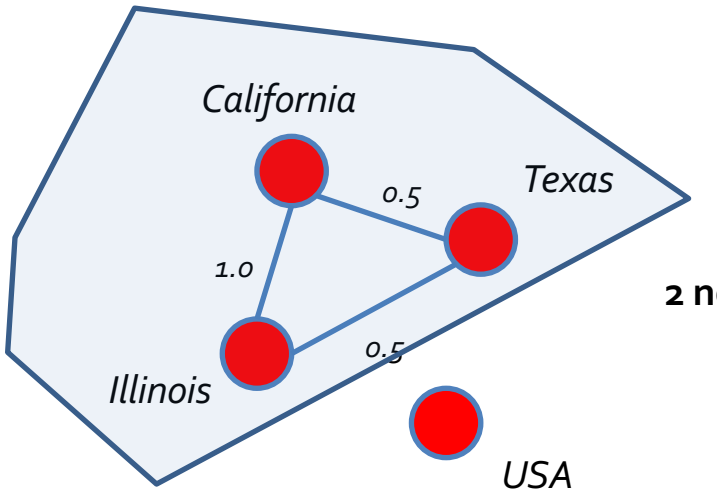
Feature Exploitation

- Use **all context features** to calculate entity-entity similarity graph.



Feature Exploitation: k-NN

- Based entity-entity similarity graph, select k-nearest neighbors of seeds.



2 nearest-neighbor of seed "Texas"

Entity-entity Similarity Graph

Problems of Feature Exploitation

- Incorporate too much noisy features, especially for unstructured context features such as skip-gram.
- Low efficiency for relatively large scale data (~= 100k nodes)

Seeds: {Medicare, Patriot Act, Obamacare}

Super bowl
Cold War
Voting Rights Act
USA Patriot Act
Iowa caucuses
Oscars
World Cups
New Hampshire primary
Academy awards

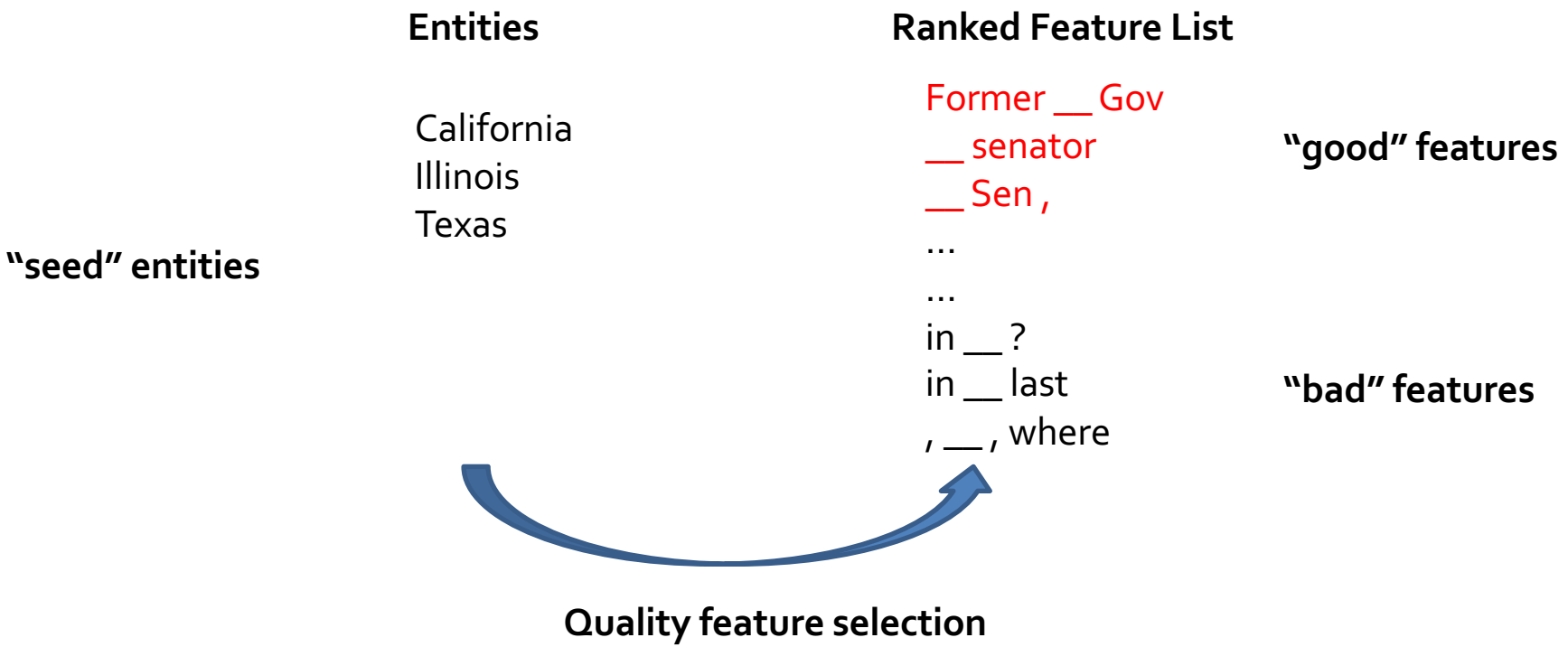
Extracted Entities

be the __ of the
the __ Here 's a
the __ He later
their __ ''
of the __ On
use the __ to
's __ , but the
the __ is over
the __ , getting

Top contributing features

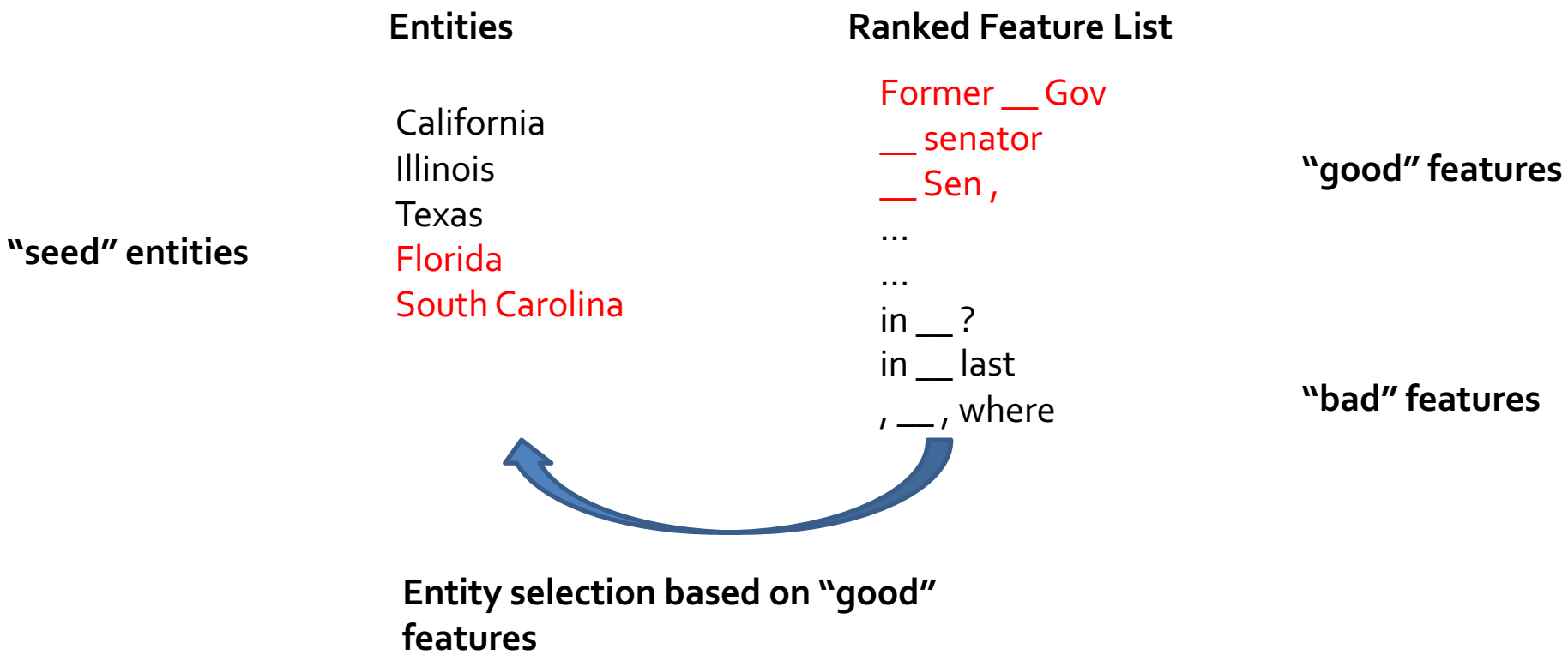
From feature exploitation to feature selection

- Select good features to calculate entity-entity similarity



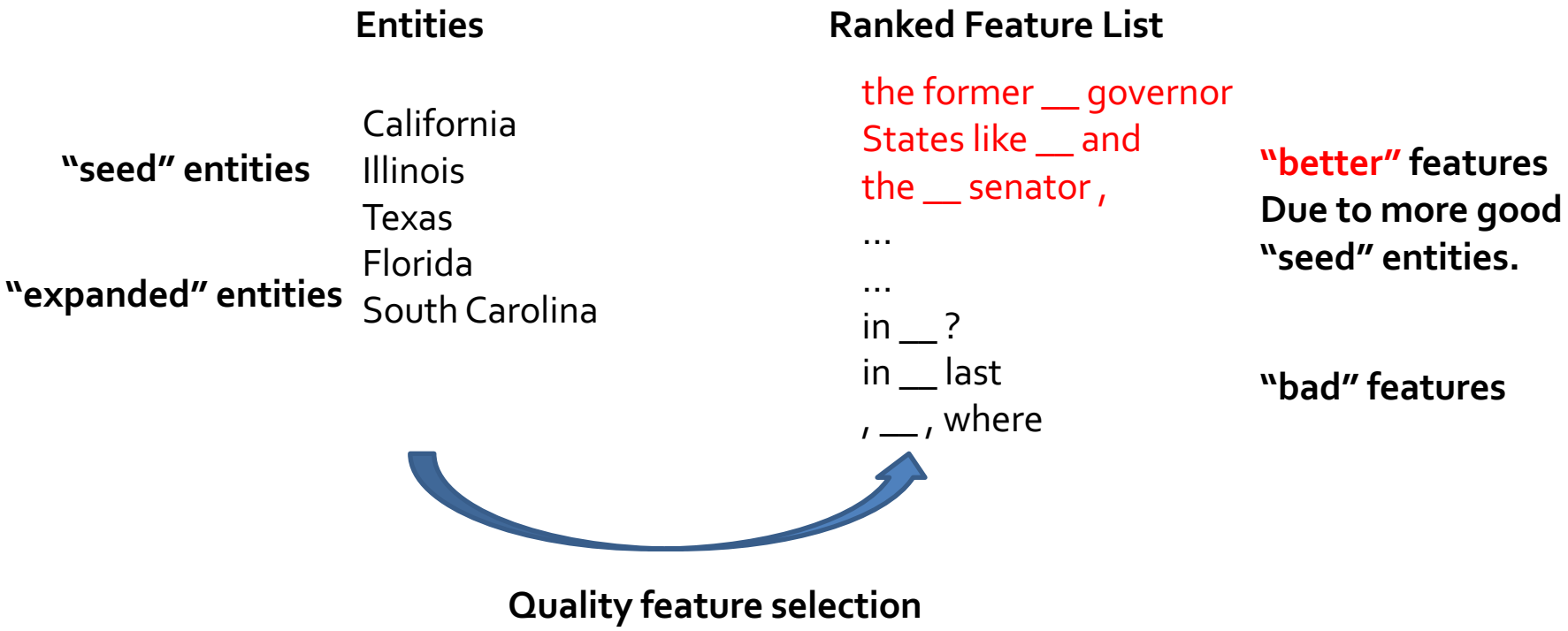
From feature exploitation to feature selection

- Select good features to calculate entity-entity similarity



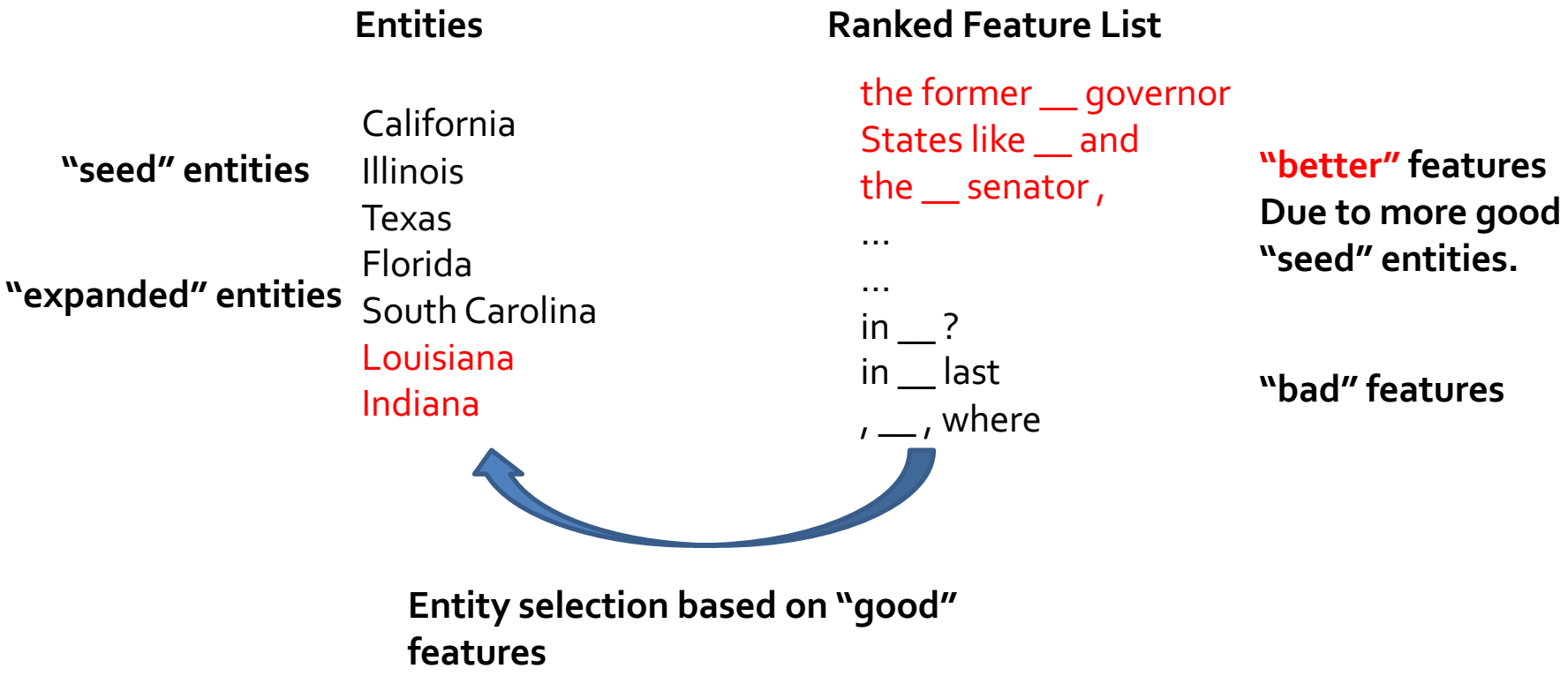
From feature exploitation to feature selection

- Select good features to calculate entity-entity similarity



From feature exploitation to feature selection

- Select good features to calculate entity-entity similarity



From feature exploitation to feature selection

- Initially, expanded entity set = seeds
- In each iteration:
  - Feature selection:
    - Score features based on current expanded entity set from scratch.
    - Select top quality features and extract candidate entities.
  - Entity selection:
    - Score each candidate entity based on its similarity with each entity in current expanded entity set conditioned on quality features.
    - Select top quality entities and add into the expanded entity set.

Entity Selection via ranking-based ensemble

- How to score features and select entities
  - Hard to find the "optimal" scoring function, easy to find "relatively-good" one.
  - Always can find a low quality feature ranked above a high quality feature.

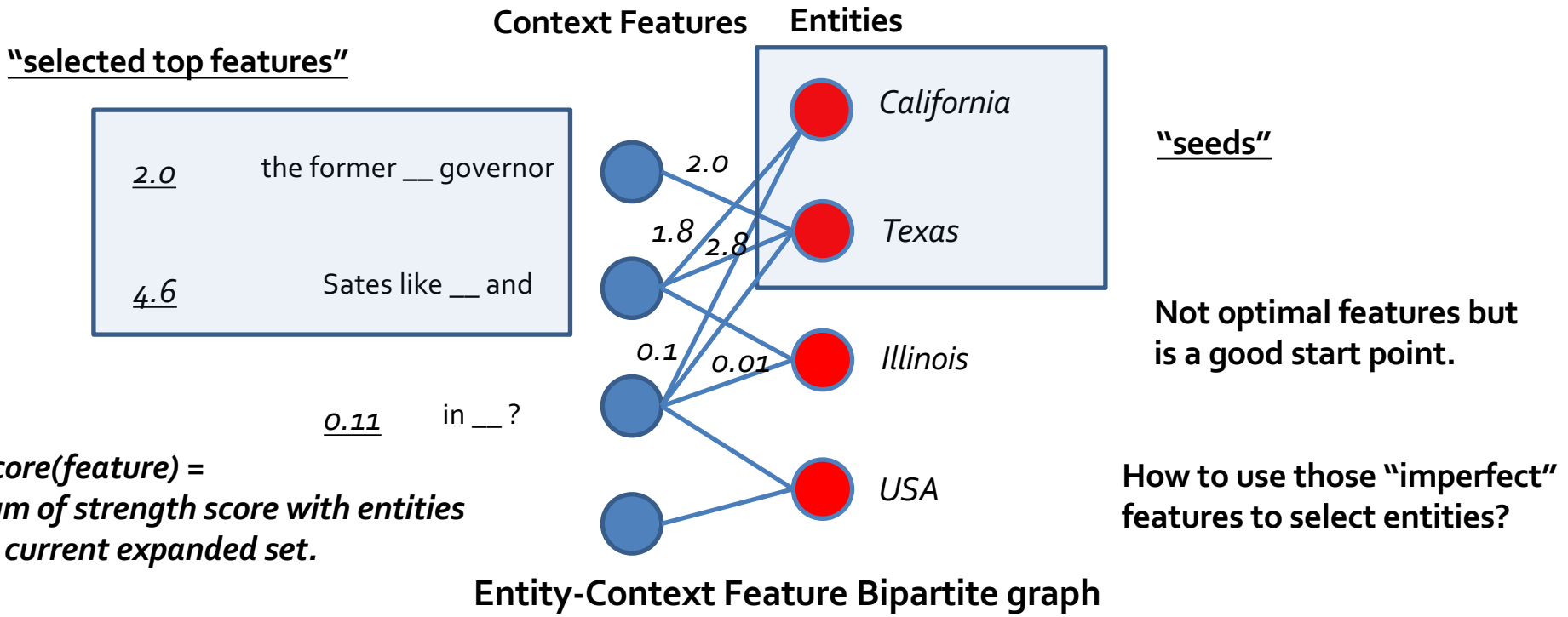
Features	Quality ranking from unknown ground truth	Quality ranking by "Optimal" scoring function	Quality ranking "relatively-good" scoring function
the former __ governor	1	1	2
Sates like __ and	2	2	1
The __ senator	3	3	3
…	…		
in __ ?	10,000	10, 000	9,988
In __ last	10,001	10, 001	10,002
, __ , where	10,002	10, 002	10,000
…	…	…	

Exactly same order with ground truth

Positively correlated with ground truth

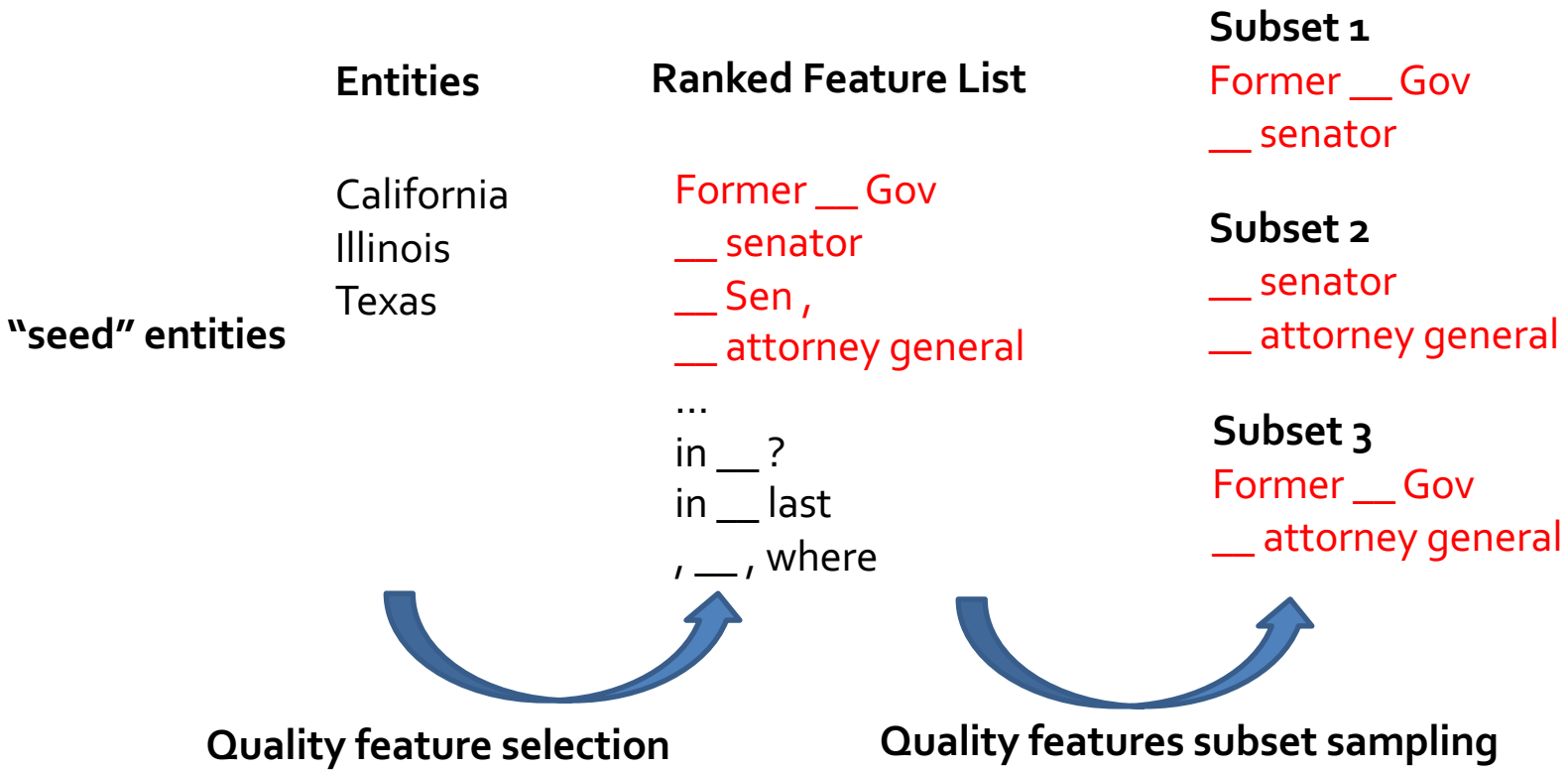
Entity Selection via ranking-based ensemble

- An intuitive "relatively-good" feature scoring function.



Entity Selection via ranking-based ensemble

- Quality feature sampling



Entity Selection via ranking-based ensemble

- Entity selection via ranking-based ensemble.

Subset 1
Former __ Gov
__ senator

Subset 2
__ senator
__ attorney general

Subset 3
Former __ Gov
__ attorney general

Entity selection based on subset features

Expanded set 1
Florida *1/1*
South Carolina *1/2*
Tamil Nadu *1/3*

Expanded set 2
South Carolina *1/1*
North Carolina *1/2*
China Daily *1/3*

Expanded set 3
North Carolina *1/1*
Florida *1/2*
Baja California *1/3*

Ranking-based ensemble

Combined expanded set
Florida *3/2*
South Carolina *3/2*
North Carolina *3/2*
Tamil Nadu *1/3*
China Daily *1/3*
Baja California *1/3*

Entity Selection via ranking-based ensemble

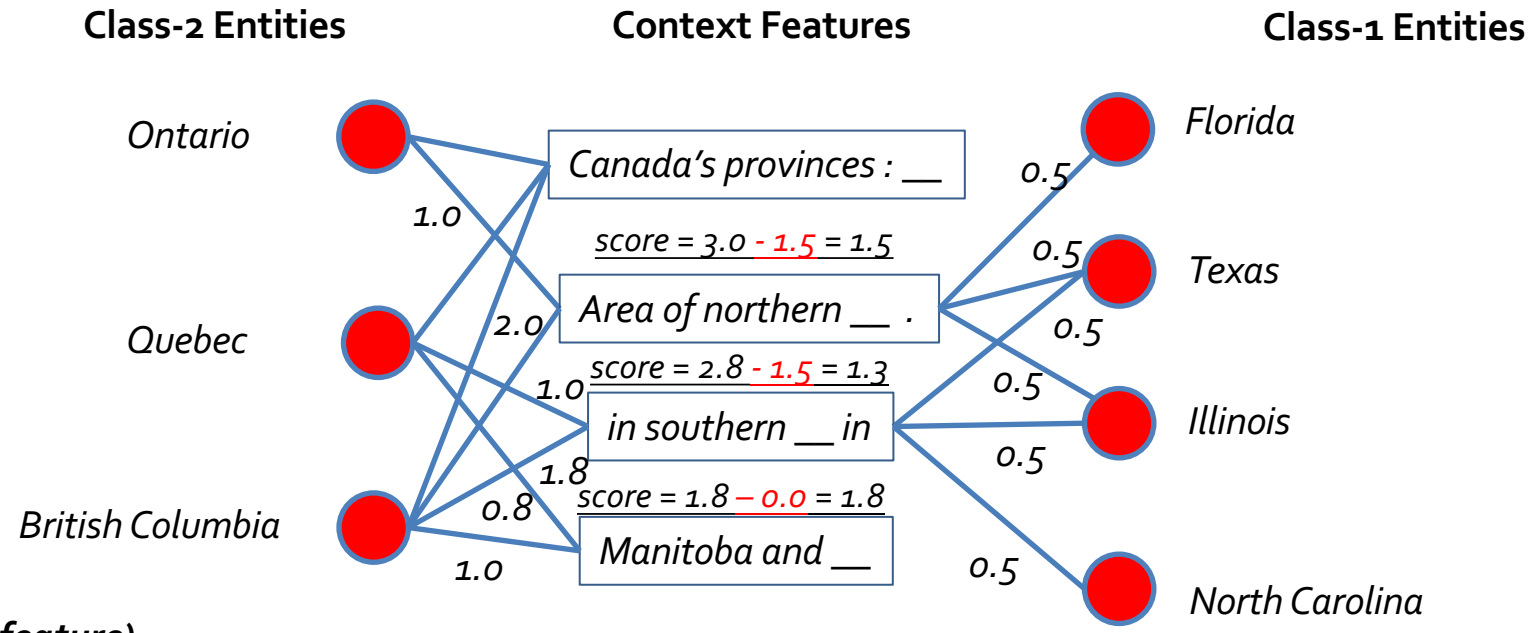
- Initially, expanded entity set = seeds
- In each iteration:
  - Feature selection:
    - Score features based on current expanded entity set from scratch.
    - Select top quality features and extract candidate entities.
  - Entity selection via ranking-based ensemble:
    - Repeat:
      - Sample a subset of top quality features
      - Rank each candidate entity based on its similarity with each entity in current expanded entity set conditioned on sampled subset of quality features.
      - Update ranking-based score for each entity
    - Select top quality entities and add into the expanded entity set.

Further improvement by negative examples

- Cross-set negative examples
  - Assume the multiple sets are mutually exclusive. →Positive examples in class 1 are negative examples for all other classes.
  - Examples:
    - Class 1: {Texas, Illinois, Florida}, Class 2: {Ontario, Quebec, British Columbia}. An US state will not be a Canada Provinces.
    - Class 1: {Google, Apple, IBM}, Class 2: {Stanford, UIUC, UCSB}. A company will not be a university.
  - We ***softly incorporate*** the mutually exclusiveness assumption into the our feature scoring procedure.

Further improvement by negative examples

- Cross-set negative-aware feature scoring:



Score(feature) =
sum of strength score with positive entities minus
sum of strength score with negative entities

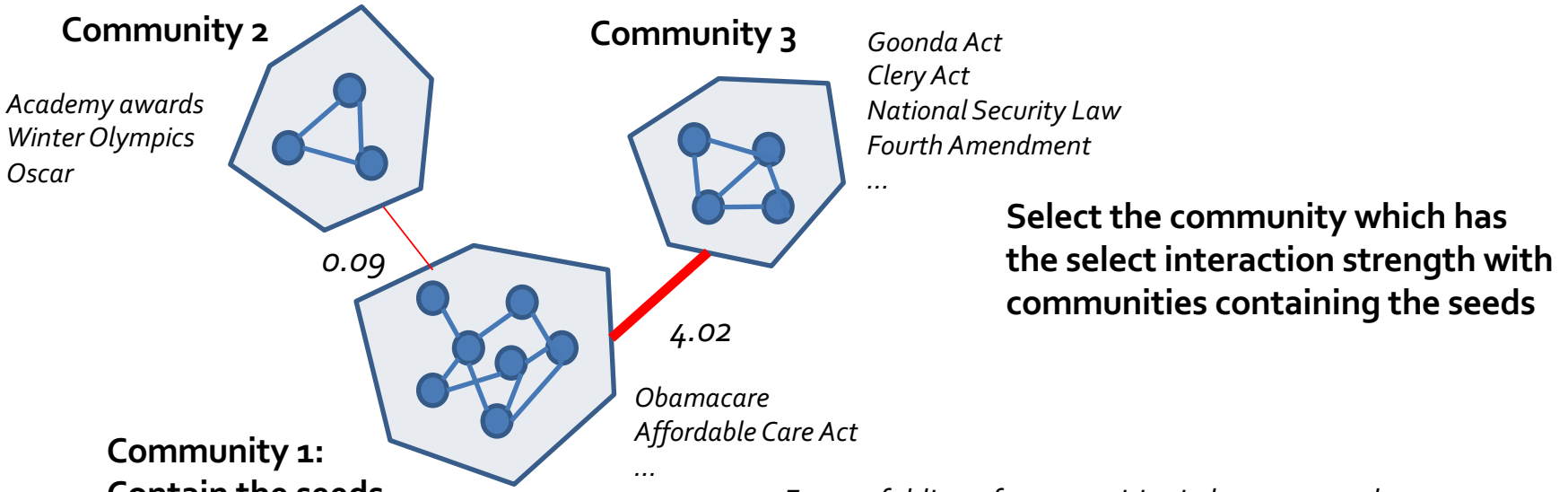
Favor those features matching entities in one unique semantic class

Further improvement by negative examples

- Near-boundary negative examples
  - Sometimes two semantic classes are too remote.
    - For example, Class 1: {Obamacare, Clean Air Act, First Amendment} -> All US laws, Class 2: {Zhejiang, Jiangsu, Guangdong} -> All Chinese provinces.
    - Entities in Class 2 are not "good" negative examples for Class 1.
  - Sometimes it's hard even for human to find good "negative".
    - For example, what are appropriate negative examples for Class 1 "LAW"?
  - We use a data-driven method to find near-boundary negative examples, and feedback to our previous modules.

Further improvement by negative examples

- Near-boundary negative examples
  - Conduct community detection on entity-entity similarity graph.
    - Currently use the Louvian Algorithm.



Fast unfolding of communities in large networks, Journal of Statistical Mechanics: Theory and Experiment 2008

Experimental Results

- APR (Associated Press and Reuter) Results

Law
Seeds: {Medicare, Patriot Act, Affordable Care Act }

Entity in rank 1-9	Entity in rank 10-18
Medicare	Civil Rights Act
Patriot Act	RFRA
Affordable Care Act	Federal Records Act
USA Freedom Act	First Amendment
Voting Rights Act	Geneva Convention
Super bowl	Clean Water Act
Clean Air Act	Fourth Amendment
USA Patriot Act	Antiquities Act
Freedom Act	Anti-terrorism Act

Political Parties
Seeds: {GOP, Democratic, Labour}

Entity in rank 1-9	Entity in rank 10-18
GOP	***White house***
Labour	Republican party
Republicans	Lib Dem
Democratic	SNP
Democrats	PDP
Conservative	US senate
Tory	Greens Party
Likud	Nationalist
Ukip	AAP

Experimental Results

- Wikipedia Results

Diseases
Seeds: {tuberculosis, malaria, cancer}

Entity in rank 1-9	Entity in rank 10-18
Tuberculosis	hypertension
cancer	psoriasis
AIDS	leukemia
Lung cancer	Hypothyroidism
Breast cancers	Multiple sclerosis
Alzheimer's disease	phenylketonuria
malaria	asthma
Heart failure	insomnia
HIV	inflammation

Companies
Seeds: {Facebook, Goldman Sachs, Google}

Entity in rank 1-9	Entity in rank 10-18
Facebook	Credit Lyonnais
Goldman Sachs	Julius Bar
Google	Grameen Bank
myspace	Amazons
Twitter	Metacritic
Youtube	Rotten tomatoes
Citigroup	Intel
Wasserstein Perella	Amazon.com
eBay	Microsoft

Experimental Results

- PubMed Abstracts Results

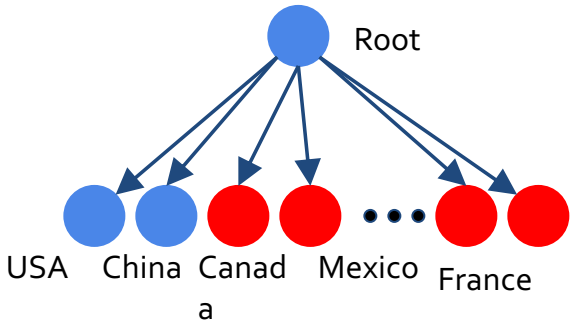
Proteins

Seeds: {hemoglobins, kinases, crp}

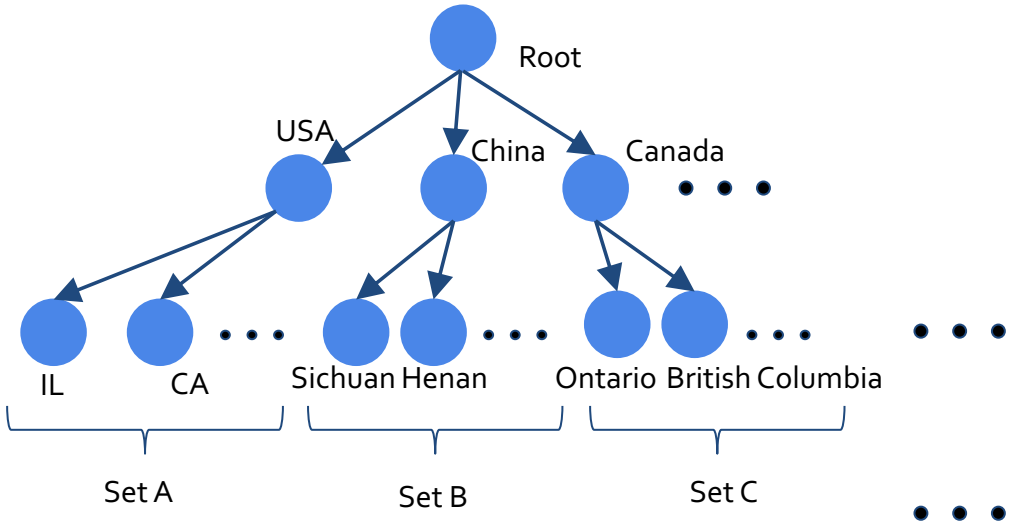
Entity in rank 1-9	Entity in rank 10-18
Hemoglobins	interferons
Kinases	PPAR
crp	transforming growth factor
fibrinogen	PKC
tnf	adiponectin
Tgf	PDGFr
Bnp	IL-5
kinase-MB	GRO
IFN	collagenase

A journey to taxonomy

- Better multi-set expansion results -> Better width expansion
- Reduce error in each level -> Better quality of whole taxonomy
- A small example for Location Taxonomy in Wikipedia



Width expansion



Whole taxonomy induction

References

- A. El-Kishky, Y. Song, C. Wang, C. R. Voss, and J. Han, "Scalable Topical Phrase Mining from Text Corpora", VLDB'15
- J. Liu, J. Shang, C. Wang, X. Ren, J. Han, "Mining Quality Phrases from Massive Text Corpora", SIGMOD'15
- J. Liu, X. Ren, J. Shang, T. Cassidy, C. Voss and J. Han, "Representing Documents via Latent Keyphrase Inference", WWW'16
- X. Ren, A. El-Kishky, C. Wang, F. Tao, C. R. Voss, H. Ji and J. Han, "ClusType: Effective Entity Recognition and Typing by Relation Phrase-Based Clustering", KDD'15
- X. Ren, W. He, M. Qu, C. R. Voss, H. Ji, J. Han, "Label Noise Reduction in Entity Typing by Heterogeneous Partial-Label Embedding", KDD'16
- Y. Sun and J. Han, Mining Heterogeneous Information Networks: Principles and Methodologies, Morgan & Claypool Publishers, 2012
- J. Tang, M. Qu, M. Zhang, Q. Mei, Large-scale Information Network Embedding, WWW'15
- C. Wang and J. Han, Mining Latent Entity Structures, Morgan & Claypool, 2015

References (cont.)

- C. Apte and S. Weiss. Data mining with decision trees and decision rules. Future Generation Computer Systems, 13, 1997
- P. K. Chan and S. J. Stolfo. Learning arbiter and combiner trees from partitioned data for scaling machine learning. KDD'95
- A. J. Dobson. An Introduction to Generalized Linear Models. Chapman & Hall, 1990.
- R. O. Duda, P. E. Hart, and D. G. Stork. Pattern Classification, 2ed. John Wiley, 2001
- U. M. Fayyad. Branching on attribute values in decision tree generation. AAAI'94.
- Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. J. Computer and System Sciences, 1997.
- J. Gehrke, R. Ramakrishnan, and V. Ganti. Rainforest: A framework for fast decision tree construction of large datasets. VLDB'98.
- J. Gehrke, V. Gant, R. Ramakrishnan, and W.-Y. Loh, BOAT -- Optimistic Decision Tree Construction. SIGMOD'99.
- T. Hastie, R. Tibshirani, and J. Friedman. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer-Verlag, 2001.
- T.-S. Lim, W.-Y. Loh, and Y.-S. Shih. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. Machine Learning, 2000

References (cont.)

- J. Magidson. The Chaid approach to segmentation modeling: Chi-squared automatic interaction detection. In R. P. Bagozzi, editor, Advanced Methods of Marketing Research, Blackwell Business, 1994
- M. Mehta, R. Agrawal, and J. Rissanen. SLIQ : A fast scalable classifier for data mining. EDBT'96
- T. M. Mitchell. Machine Learning. McGraw Hill, 1997
- S. K. Murthy, Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey, Data Mining and Knowledge Discovery 2(4): 345-389, 1998
- J. R. Quinlan. Induction of decision trees. Machine Learning, 1:81-106, 1986.
- J. R. Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufmann, 1993.
- J. R. Quinlan. Bagging, boosting, and c4.5. AAAI'96.
- R. Rastogi and K. Shim. **Public: A decision tree classifier that integrates building and pruning**. VLDB'98
- J. Shafer, R. Agrawal, and M. Mehta. **SPRINT : A scalable parallel classifier for data mining**. VLDB'96
- J. W. Shavlik and T. G. Dietterich. **Readings in Machine Learning**. Morgan Kaufmann, 1990
- P. Tan, M. Steinbach, and V. Kumar. **Introduction to Data Mining**. Addison Wesley, 2005
- S. M. Weiss and C. A. Kulikowski. **Computer Systems that Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems**. Morgan Kaufman, 1991
- S. M. Weiss and N. Indurkhya. **Predictive Data Mining**. Morgan Kaufmann, 1997
- I. H. Witten and E. Frank. **Data Mining: Practical Machine Learning Tools and Techniques**, 2ed. Morgan Kaufmann, 2005