Bachelor Thesis Presentation

InvIdenti: Author Disambiguation for Medical Patents

Guide (IIIT-A) : Prof. Dr. U.S. Tiwary
Guide (RWTH Aachen) : PD Dr. Christoph Quix
Enrolment : IIT2012108
Email : iit2012108@iiita.ac.in / alekh@dbis.rwth-aachen.de

Contents

Prof. M. Jarke
Lehrstühl Informatik 5
RWTH Aachen

Introduction and Goals

1. What and Why?

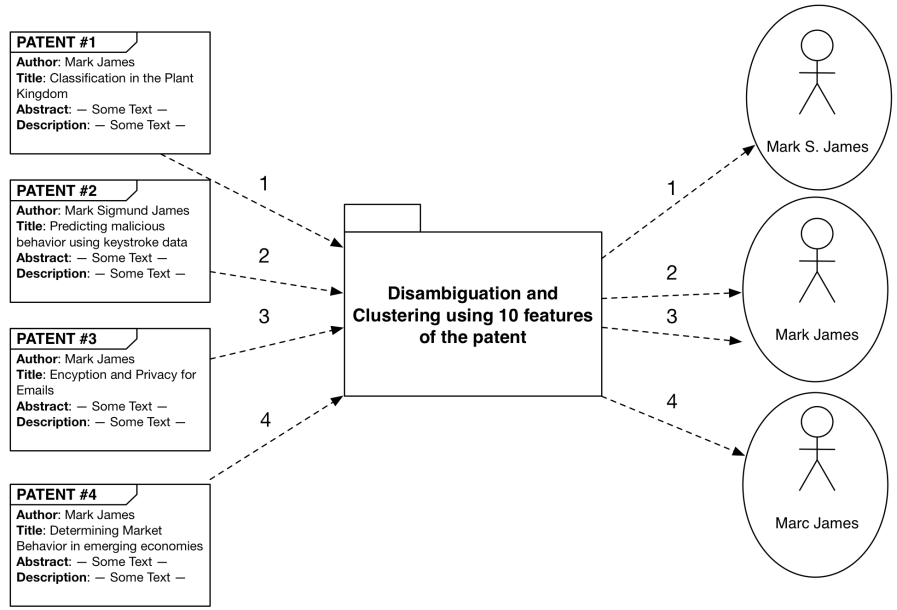
- Author Disambiguation: Distinguish between inventors with same or similar names / competence fields
- Identifying by name has severe limitations
- Spelling errors in patent database introduce ambiguity
- Authors/Inventors may share name and/or expertise area
- Manual Approaches infeasible and not future-proof due to explosion in number of patents

Prof. M. Jarke
Lehrstühl Informatik 5
RWTH Aachen

Introduction and Goals

2. Software Functionality Goals

- Feature Selection : Find good and representative features for the disambiguation task
- Importance Weighting of Features
- Similarity Calculation
- Patent Clustering
- Patent-Publication Matching

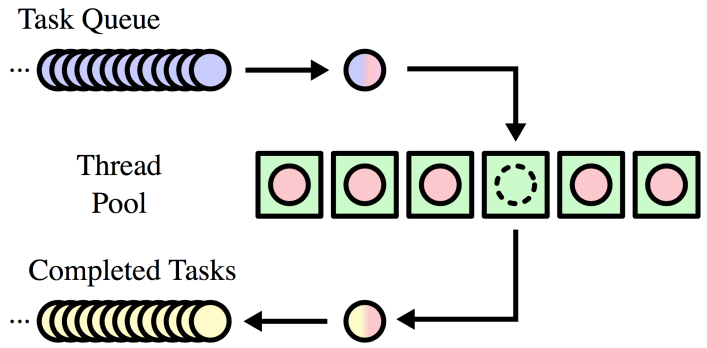


Prof. M. Jarke
Lehrstühl Informatik 5
RWTH Aachen

Introduction and Goals

3. Software Quality Goals

- Software Design and Architecture : Software should conform to S.O.L.I.D principles for code maintainability and possibility of future extension
- Support for Parallelization & Multiprocessor Architecture
- Lucid Documentation for long-term maintainability
  - UML Diagrams
  - JavaDoc™ Documentation
  - Wiki Pages



Prof. M. Jarke
Lehrstühl Informatik 5
RWTH Aachen

Background

1. Project Mi-Mappa

- Complex innovation in medical engineering not possible without collaboration
- Goal is to develop an integrative competence model based on Data Mining Algorithms
- Assignment of patents and medical products to competence fields
- Actors selected based on published texts for a given project
- Use of Ontology Modeling and matching, Data and Text Mining

Prof. M. Jarke
Lehrstühl
Informatik 5
RWTH Aachen

Background

2. Related Work

- **PatentsView Inventor Disambiguation Workshop – Sept. '15**
  Neural Networks, Rule-based methods, Ensemble ML Methods for Inventor Disambiguation

- **[Fleming et al. 2014] Disambiguation and Co-Authorship Networks of the US Patent Inventor Database(1975-2010)**
  Uses a Naïve Bayesian Classifier Technique with Blocking

- **[Maraut et al. 2014] Identifying author-inventors from Spain**
  Computes a global similarity and clusters inventors based on that

Sanchit Alekh

mi-Mapa

Prof. M. Jarke
Lehrstühl Informatik 5
RWTH Aachen

Solution: Outline

भारतीय सूचना प्रौद्योगिकी संस्थान इलाहाबाद | RWTH AACHEN UNIVERSITY

1. Underlying data-structure used is an Inventor-Patent Instance, which stores the metadata as well as textual features
2. An Assortment of 10 features is used, out of which there are 6 metadata and 4 textual features
3. Different Feature Similarity metrics are used for each of the features to compute a weighted similarity matrix between instance pairs
4. Weight Training is done using pre-labelled instances from dataset provided by Fleming et al. using Logistic Regression
5. Hierarchical Clustering and DBSCAN are used to assign inventor-patent instances to clusters with unique inventors

Solution: Flow

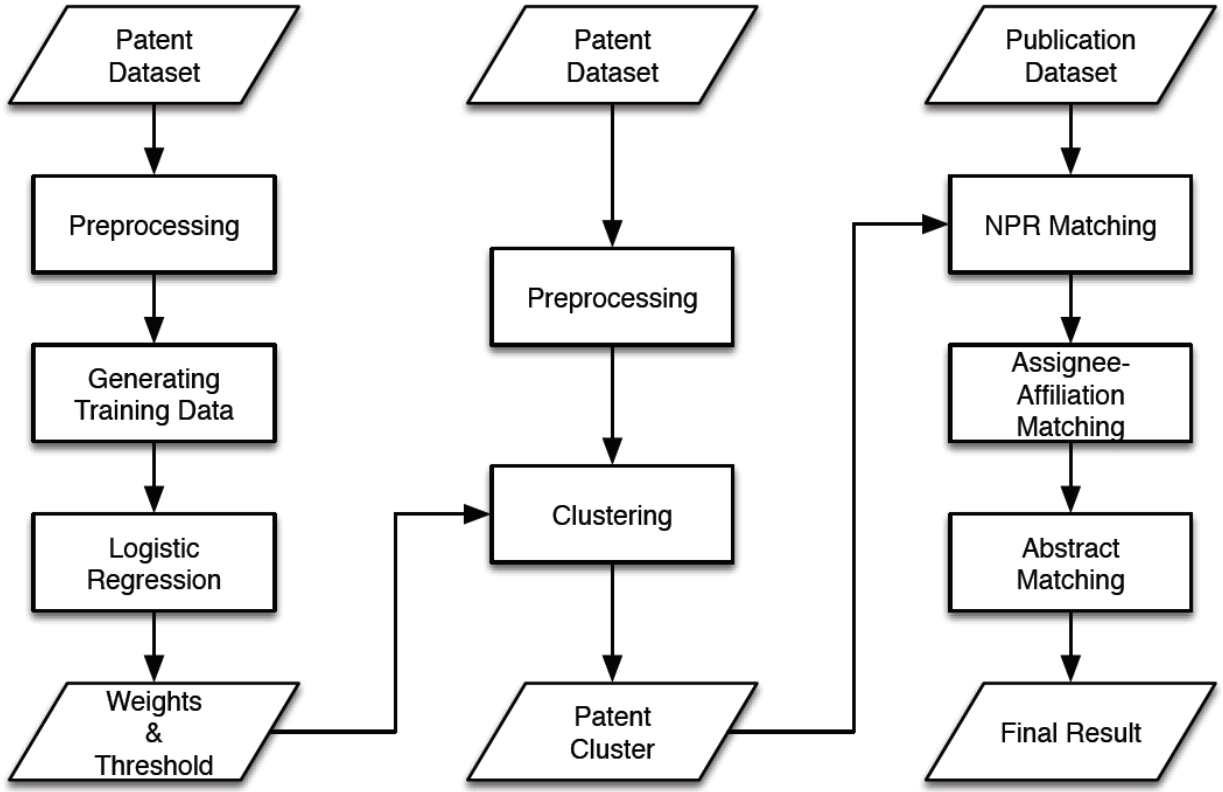


Fig. 9.1 Flowchart of processes involved in InvIdenti

Sanchit Alekh
28 July 2016
Slide 10

Solution: Inventor-Patent Instance

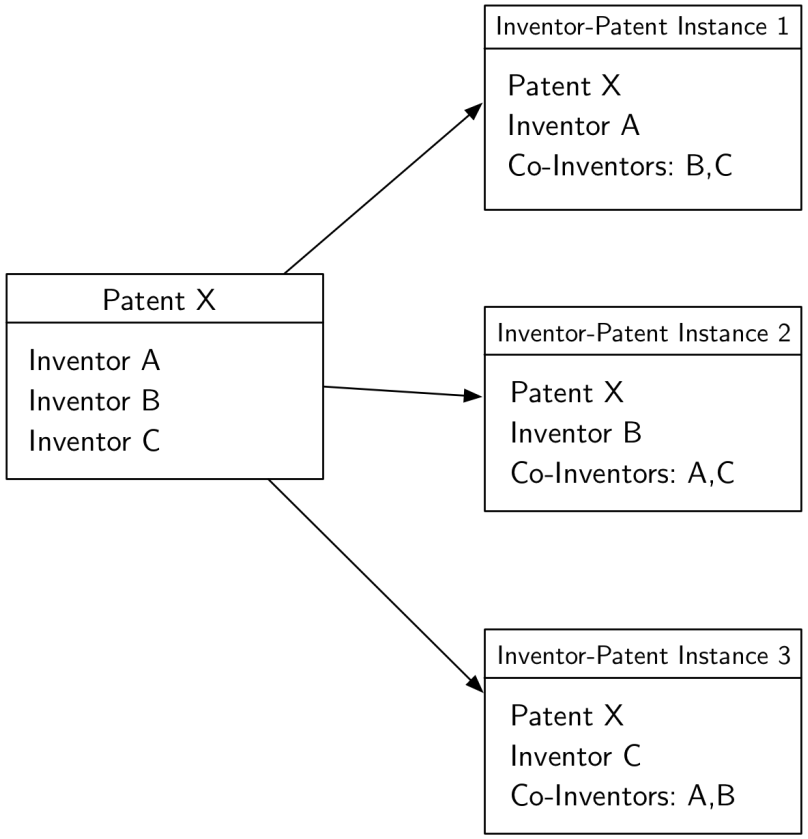


Fig. 10.1 Inventor Patent Instances obtained from Patent X

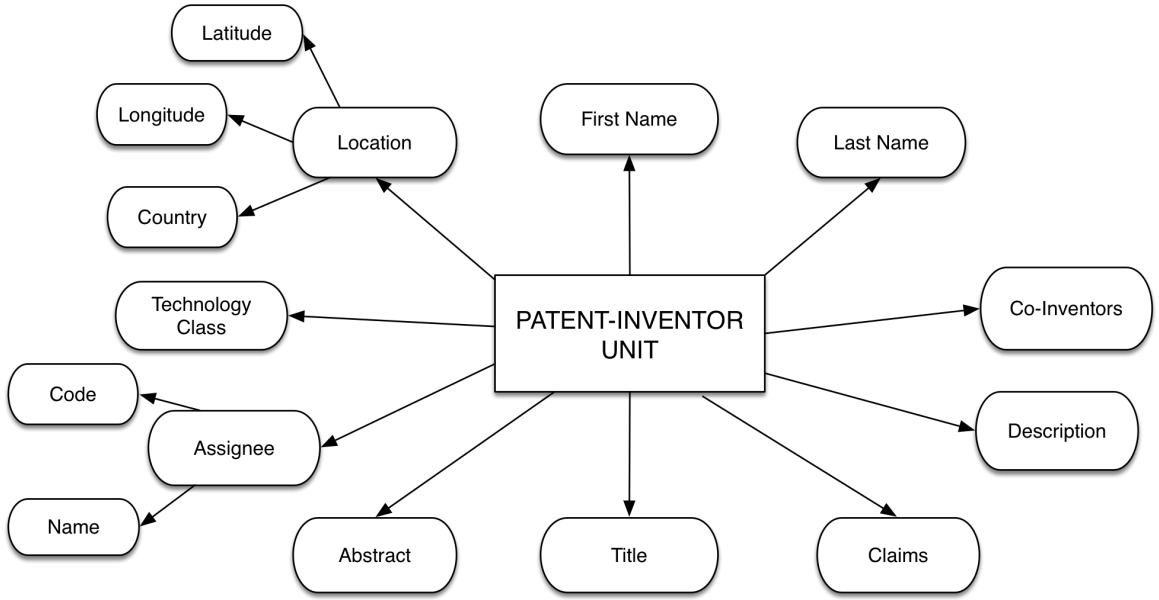


Fig. 10.2 Ten features used to represent the Inventor-Patent Instance

Prof. M. Jarke
Lehrstühl Informatik 5
RWTH Aachen

Solution: Similarity

- **Feature Similarity Techniques**
  1. Name : Levenshtein Distance
  2. Location : Country + Distance (from Latitude & Longitude)
  3. Assignee : Assignee Code + Levenshtein Distance of Ass. Name
  4. Technology Class : Number of shared classes
  5. Co-Inventors : Number of Shared Co-Inventors
  6. Textual Features : Cosine Similarity between Document Vectors

$$Level(x) = \begin{cases} 5 & \text{if } x \leq 5km \\ 4 & \text{if } x \leq 10km \\ 3 & \text{if } x \leq 25km \\ 2 & \text{if } x \leq 50km \\ 1 & \text{otherwise} \end{cases}$$

$$Level(x) = \begin{cases} 6 & \text{if } x \leq 6 \\ x & \text{if } x \in \{1,2,3,4,5\} \end{cases}$$

$$(cosine(U,V)) = \frac{\sum_{i=1}^{n} u_i \cdot v_i}{\sqrt{\sum_{i=1}^{n} {u_i}^2} \cdot \sqrt{\sum_{i=1}^{n} {v_i}^2}}$$

Fig. 11.1 Feature Similarity Calculation for Location, Co-Author and Textual Features

Solution: Similarity

- **Feature Similarity Transformations**
  1. Distance Measures are converted to Similarity Measures
  2. All Similarity values are normalized to fall within range [0,1]

- **Global Similarity**
  1. $S_{global} = \sum_{i=1}^{n} w_i S_i$, where w_i are feature weights and S_i are the normalized similarity values
  2. Threshold : ϵ

- **How to find suitable values for weights and threshold?**
  - ❖ Logistic Regression

Prof. M. Jarke
Lehrstühl Informatik 5
RWTH Aachen

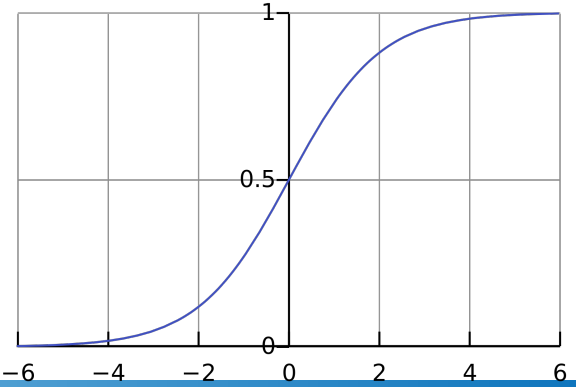
Solution: Logistic Regression

- Maximum Log-likelihood is used to model the Probability $P(Y = 1 \,|X = x)$ based on binary output variable Y $\in \{0,1\}$
- The Logistic (or Logit) Function is used to model this probability as it is bounded in both directions. The equation is:

$$log \frac{P(Y = 1|X = x)}{1 - P(Y = 1|X = x)} = \sum_{i}^{n} x \cdot w^T$$

- On solving for $P(Y = 1 \,|X = x)$, we get the Sigmoid Function

$$P(Y = 1|X = x) = \frac{1}{1 - e^{-x \cdot w^T}}$$



Sanchit Alekh
28 July 2016
Slide 14

Solution: Logistic Regression

- Using Logistic Regression, aim is to train the model on labelled data to obtain weights and threshold
- We can say that there is a match or no match if $\sum_{i=1}^{n} w_i x_i$ is greater than or less than ϵ respectively
- For training, there must be a cost function associated with the sigmoid function. The cost function follows a –ve log form, and is given by:

$$J(w) = -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_w(x^{(i)}) + (1 - y^{(i)}) \log (1 - \log h_w(x^{(i)}))]$$

where,

$$h_w(x) = \frac{1}{1 + e^{-x \cdot w^T}}$$

Prof. M. Jarke
Lehrstühl Informatik 5
RWTH Aachen

Solution: Logistic Regression

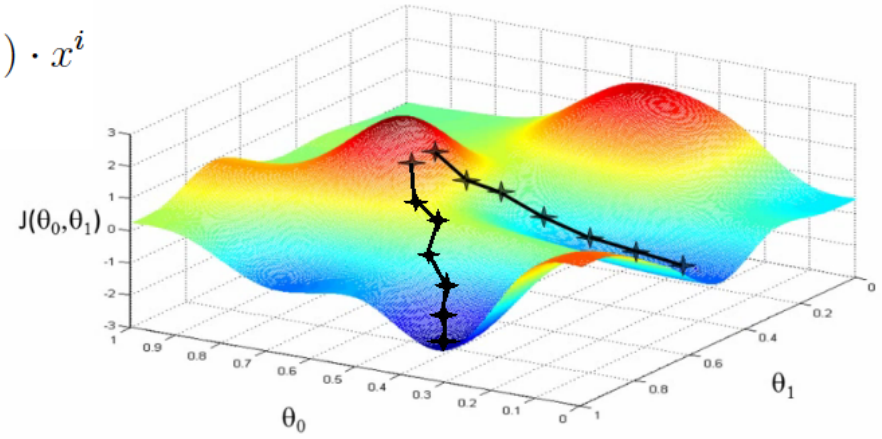
- For training in Logistic Regression, classic Gradient Descent method is used, i.e. error correction is made by a factor of the gradient of the cost function

$$\frac{\partial J(w)}{\partial w_i} = -\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)} - h_w\left(x^{(i)}\right)\right)\cdot x^i$$

- Therefore, the weight update of each parameter after every iteration of Gradient Descent is given by

$$w_i^{t+1} = w_i^t - \frac{\alpha}{m}\sum_{i=1}^{m}\left(y^{(i)} - h_w\left(x^{(i)}\right)\right)\cdot x^i$$

Where α is the learning rate

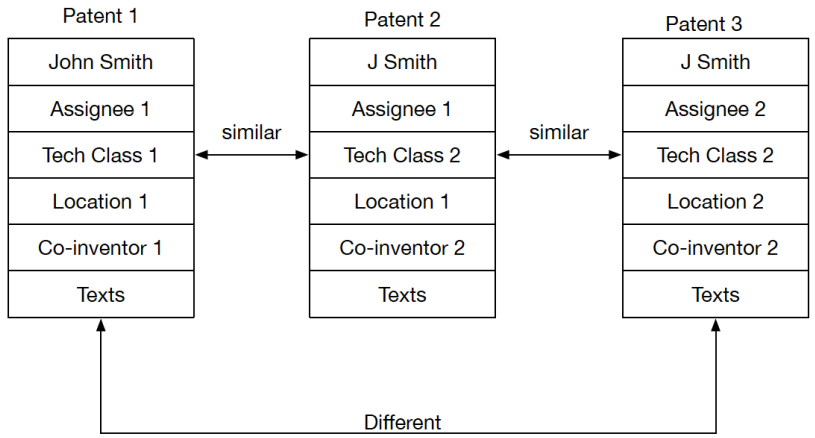


Prof. M. Jarke
Lehrstühl Informatik 5
RWTH Aachen

Solution: Transitivity

- **Simple Binary Classification using Logistic Regression does not yield good results. Why?**
  1. Many inventors cover several expertise areas
  2. Inventors may change their location or organization/university
  3. Logistic Regression often suffers from overfitting.

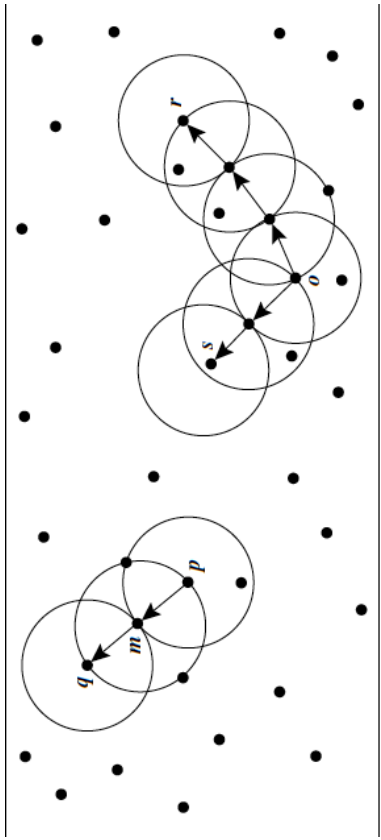
- **To remedy this, we propose that additional property, i.e. Transitivity be fulfilled by patents.**



Prof. M. Jarke
Lehrstühl Informatik 5
RWTH Aachen

Solution: Transitivity

- In InvIdenti, Transitivity is affected by Clustering Algorithms, i.e. Hierarchical Clustering and DBSCAN.
- **In Hierarchical Clustering, the type of linkage method used controls the extent of transitivity**
  1. Single-Linkage : Promotes chaining; best transitivity
  2. Complete-Linkage : Avoids chaining; worst transitivity
  3. Group-Average Linkage : Medium Transitivity
- In DBSCAN, the parameter *MinPts* determines the extent of transitivity. MinPts = 1 guarantees chaining and best transitivity

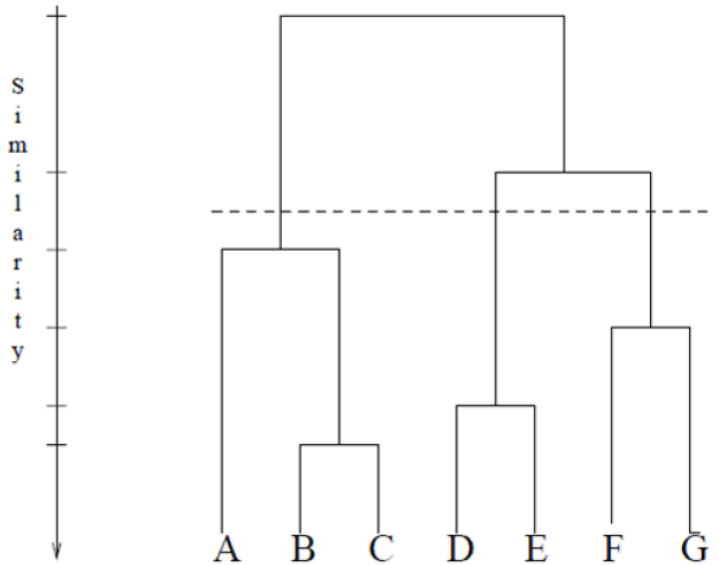


Prof. M. Jarke
Lehrstühl Informatik 5
RWTH Aachen

Sanchit Alekh
28 July 2016
Slide 18

Solution: Hierarchical Clustering

- Hierarchical Agglomerative Clustering starts with each patent in a different cluster, and then merges successfully based on the best similarity values
- The Stopping Criterion used is the threshold obtained from Logistic Regression.
- We employ Single-linkage clustering, which uses the best similarity value between clusters to merge them.
- When cluster similarities are less than the threshold, merge process is stopped

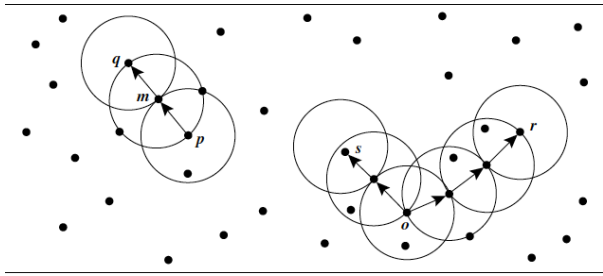


Prof. M. Jarke
Lehrstühl Informatik 5
RWTH Aachen

Solution: DBSCAN

- DBSCAN is the acronym for Density-Based Spatial Clustering of Applications with noise

- **Basic Terms**
  1. Neighbors: $N_0 = \{o_i \in O \mid S_{global}(o_i,o) \geq \epsilon\}$
  2. Core Object: $|N_0| \geq MinPts$
  3. Density-Reachability:
     - $\{o_o,o_1,o_2,\ldots o_{n-1},o_n\}$
     - o_i and o_{i+1} are neighbors, where $0 \leq i \leq n-1$
     - o_i is a core object, where $1 \leq i \leq n-1$

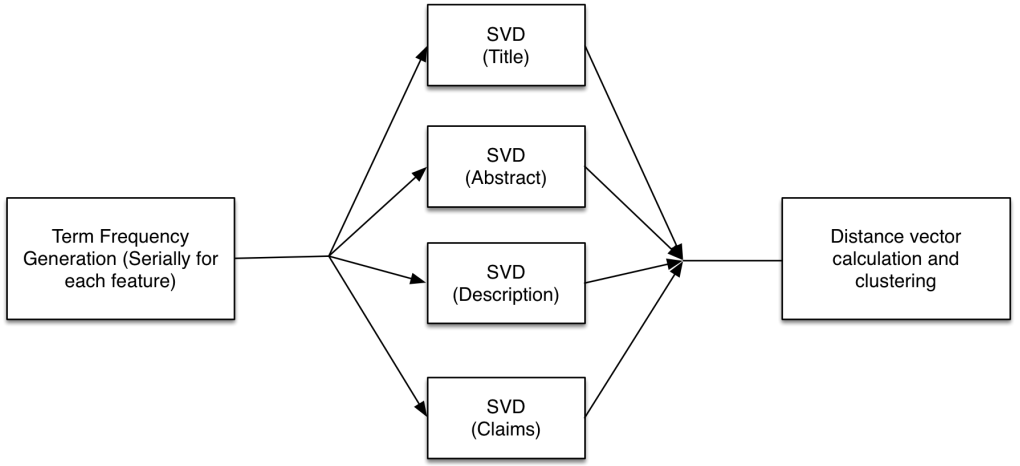


- DBSCAN finds all density-reachable objects for core objects and groups them. All unassigned points are noise

Prof. M. Jarke
Lehrstühl Informatik 5
RWTH Aachen

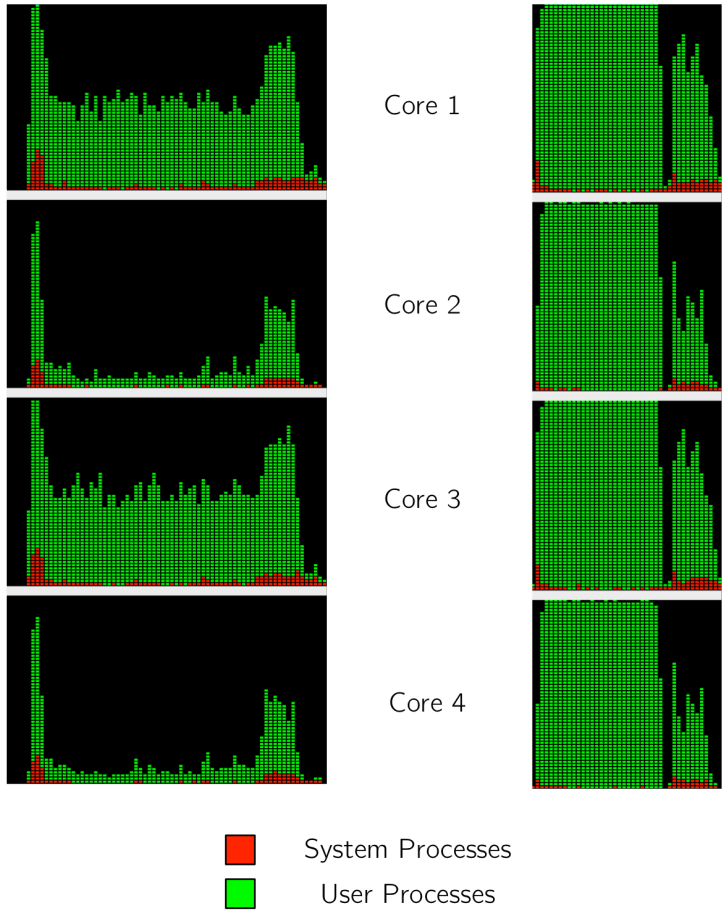
Solution: Performance Improvement

- Multithreading in Java used to significantly speed-up the training and clustering processes, making them up to 50% faster
- SVD, the slowest part of our implementation was designed to run on parallel threads for each feature, increasing CPU utilization to 100% and reducing time slice by 35-45%



- For Similarity Matrix creation, we calculate the number of logical cores of the CPU, and initialize the same number of worker threads to perform the task

Solution: Performance Impact of Multithreading



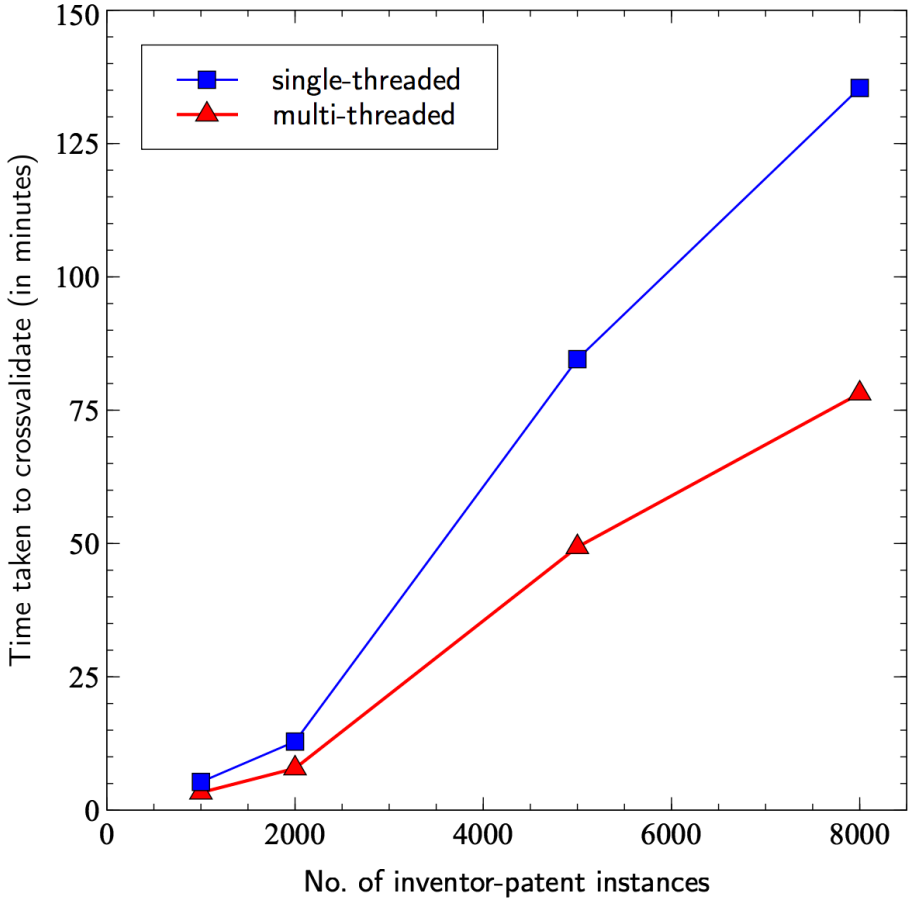


Fig. 21.1 Performance Impact of Multithreaded SVD

Prof. M. Jarke
Lehrstühl Informatik 5
RWTH Aachen

भारतीय सूचना प्रौद्योगिकी संस्थान इलाहाबाद

RWTH AACHEN UNIVERSITY

Evaluation : Datasets

- **Original Datasets Used**
  1. Engineers and Scientists (E&S) dataset, Ivan Png
  2. Inventor-Patent Instance Dataset(1975-2010), Fleming et al.
  3. Benchmark Dataset, Fleming et al.

- **Evaluation Datasets**
  1. Intersection of E&S and Inventor-Patent instance dataset
     - Training Dataset (8000)
     - Testing Dataset (24495)
  2. Fleming et al.'s Benchmark Dataset (1305)

Prof. M. Jarke
Lehrstühl Informatik 5
RWTH Aachen

Evaluation : Metrics

	Same Cluster	Different Clusters
Same ID	True Positive (TP)	False Negative (FN)
Different IDs	False Positive (FP)	True Negative (TN)

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F\text{ - }Measurement = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}}$$

$$Lumping\ Error = \frac{FP}{TP + FN}$$

$$Splitting\ Error = \frac{FN}{TP + FN}$$

Sanchit Alekh

Evaluation: Cross-validation

- K-fold cross-validation with varying sizes of randomly selected subsets from the training dataset
- At each iteration, Logistic Regression calculates the weights and threshold which is consumed by DBSCAN and Hierarchical Clustering for applying transitivity
- Average f-score is taken across the k iterations. Test of stability of the Logistic Regression, along with overfitting avoidance
- **Results of 5-fold cross-validation**: F-score consistently close to 0.99 and standard error is always less than 0.01

Prof. M. Jarke
Lehrstühl Informatik 5
RWTH Aachen

Sanchit Alekh
28 July 2016
Slide 25

Evaluation: 5-fold Cross-validation Results

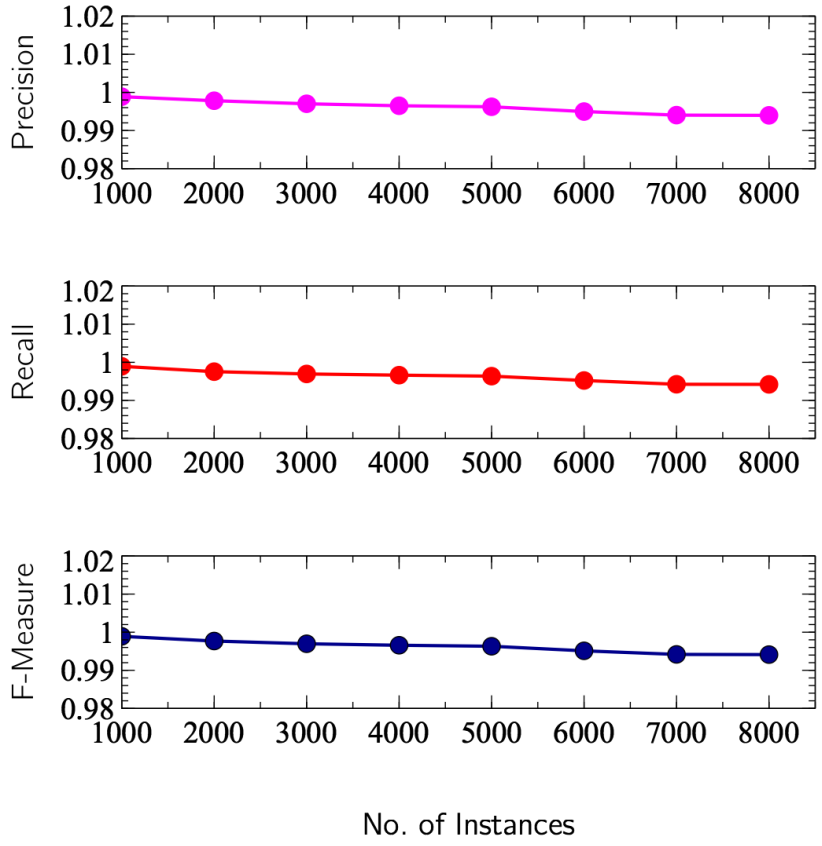


Fig. 25.1 Performance Metrics for 5-fold cross-validation

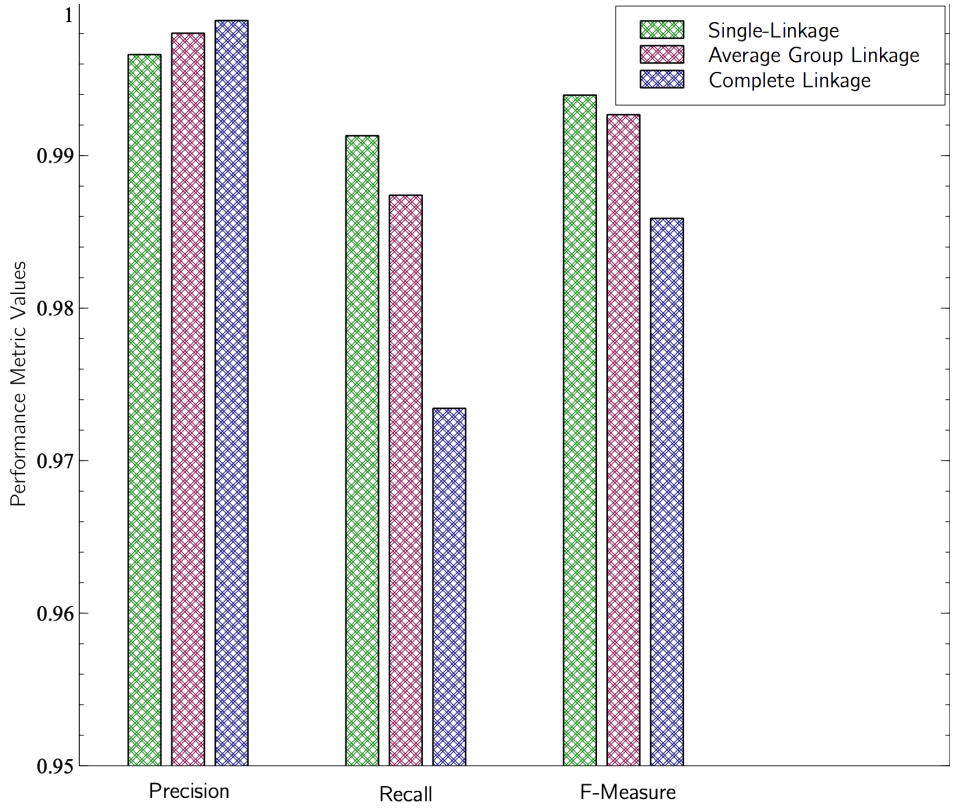


Fig. 25.2 Performance Metrics for different linkage rules in Hierarchical Clustering

Prof. M. Jarke
Lehrstühl Informatik 5
RWTH Aachen

Sanchit Alekh
28 July 2016
Slide 26

Evaluation: 5-fold Cross-validation Results

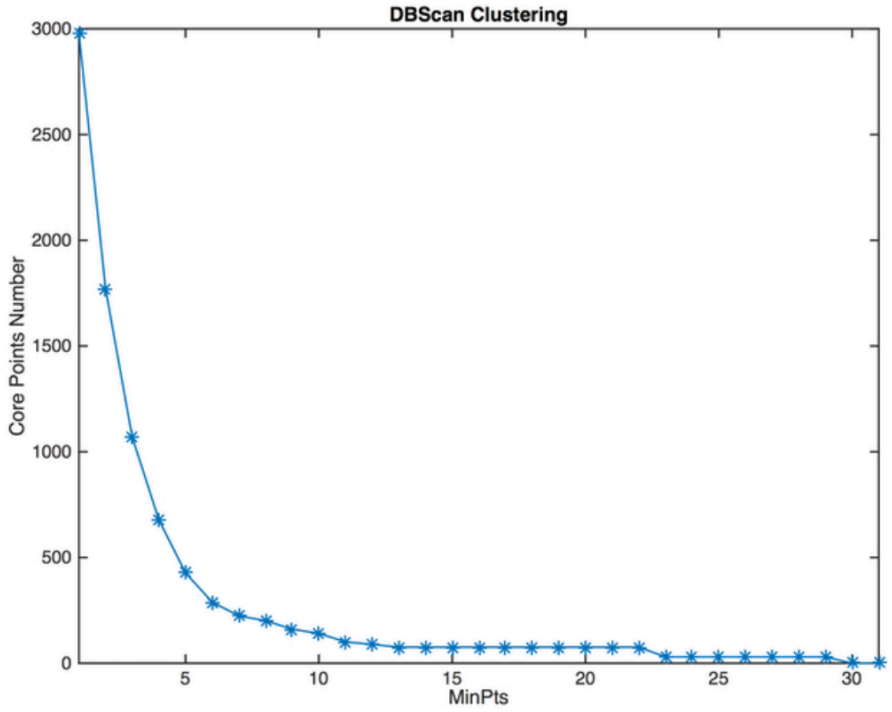


Fig. 26.1 Variation of Core Points with MinPts

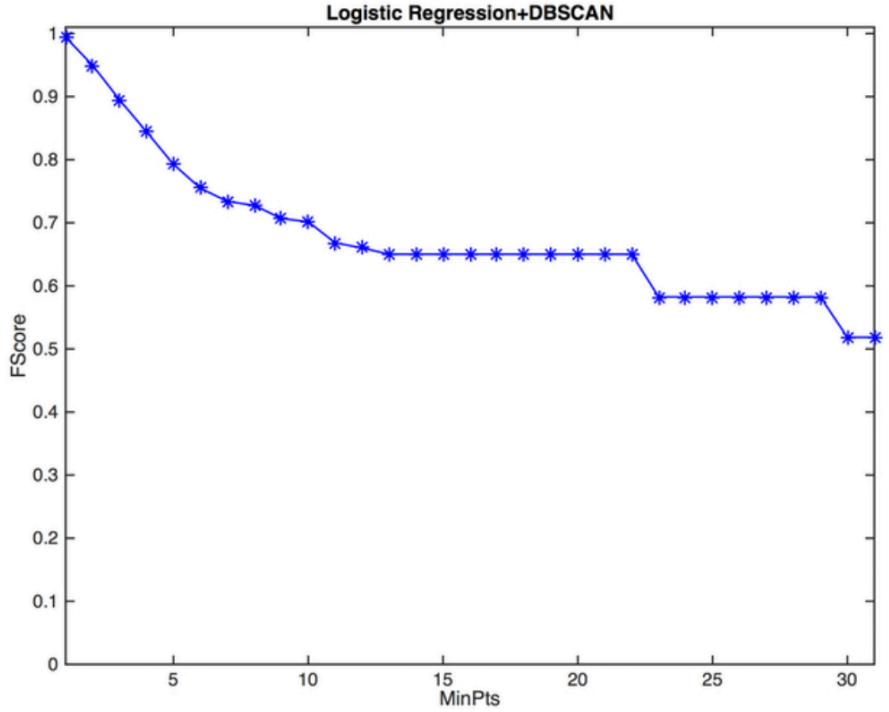


Fig. 26.2 Variation of f-score of with MinPts

Prof. M. Jarke
Lehrstühl Informatik 5
RWTH Aachen

Evaluation: Testing Dataset

- Testing on the full testing dataset. Size of the subset is varied from 2000 to 24495 with an increment of 2000.
- Values of the precision, recall and F-measure are around 0.99 whereas the lumping error and splitting error is less than 0.02.
- The subset with size 2000 has the worst F-measurement of 0.988 and the largest splitting error of 0.0159

	Precision	Recall	F-Measure	Lumping Error	Splitting Error
InvIdenti	0.98522	0.99788	0.99151	0.01497	0.00212
USPTO PatentsView	1.0	0.96616	0.98279	-	-

Fig. 27.1 Performance Metrics for InvIdenti and USPTO PatentsView. Figures don't represent actual comparison as dataset sizes are different

Evaluation: Testing on a Benchmark Dataset

- Benchmark Dataset is provided by Fleming et al. consisting of 95 inventors and 1332 inventor-patent instances
- Their approach uses 'blocking rules' on the basis of which they quote two sets of values for performance analysis.
- InvIdenti was able to achieve a 100% success rate on this benchmark dataset

	F-Measure	Lumping Error	Splitting Error
Fleming et al. (Upper Bound)	0.9744	0.0150	0.0357
Fleming et al. (Lower Bound)	0.9764	0.0150	0.0319
InvIdenti with DBSCAN	1.0	0.0	0.0
InvIdenti with Hierarchical Clustering	1.0	0.0	0.0

Fig 28.1 Testing on benchmark dataset provided by Fleming et al.

Prof. M. Jarke
Lehrstühl Informatik 5
RWTH Aachen

Conclusion

- **InvIdenti** successfully presents an automatic approach for author disambiguation, which is a more reliable approach than manual weighting
- Makes use of both patent metadata and textual properties in its assortment of 10 features
- Uses a novel transitivity approach affected by DBSCAN and Hierarchical Clustering which has proven itself to be efficacious
- Software code is clean, intelligible and upholds S.O.L.I.D principles of software engineering. Lucid code documentation is provided in the form of UML Diagrams, Wiki Pages and JavaDoc™

Sanchit Alekh

Scope for Future Work

- Project can be extended to run on a Distributed Computing architecture such as **Apache Spark**
- Clustering Algorithms can be made multithreaded by using a **Message Passing Interface**
- **MinPts=1** aims to decrease splitting error but if the database becomes massive, it might result in lumping errors
- Inventor-clusters can ultimately be associated with some semantic knowledge in order to aggregate the best pool of experts for a complex medical task **(Mi-Mappa)**

Prof. M. Jarke
Lehrstühl Informatik 5
RWTH Aachen

Other Related Work

- **[Pezzoni et al. 2012] How to kill inventors: Testing the Massacrator algorithm for Inventor Disambiguation**
  Uses 17 criteria to distinguish between the inventors

- **[Kevin et al. 2008] Measuring science-technology interaction using rare inventor-author names**
  Uses rare names to match inventors and authors

- **[Cassiman et al. 2007] Measuring industry-science links through inventor-author relations: A profiling methodology**
  Uses the concept of abstract similarities to match inventors