Project 4: Who Is Who -- Entity Resolution

Paper 4: Main Idea

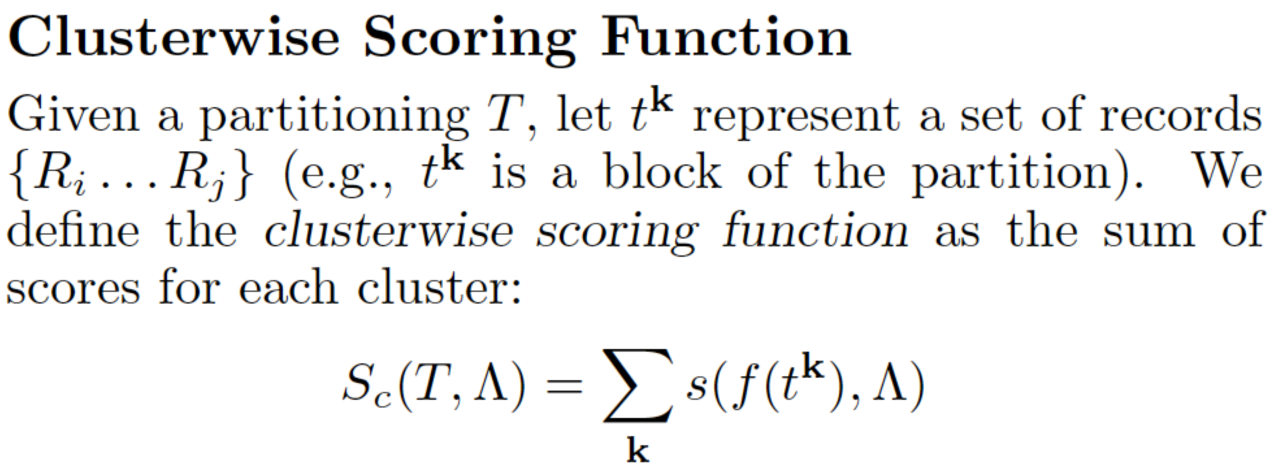
- Create two kinds of document term matrix
  - Each author as a corpus
  - All author as a corpus (more robust)
- Topic - based Bayesian model: LDA
  - Use the topic modeling package to form 10 latent topics
  - Generate theta: distribution for each records over 10 topics
- Hierarchical Clustering
  - Use those theta values (probability matrix)
  - Form distance matrix

Some differences

- Didn’t use the similarity matrix of author name
  - Already have same representation for every name

- Choose different height parameter (the dataset is different and we contain less words in each record)
  - Use cross validation define epsilon = 0.15

Paper 5: Main Idea



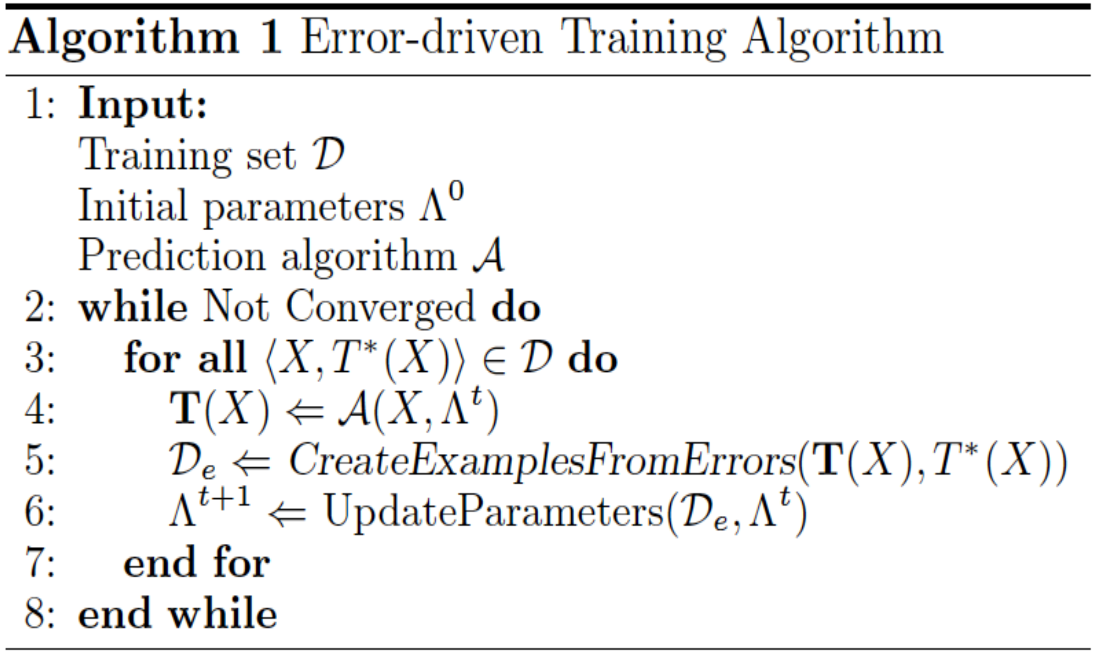
Clusterwise Scoring Function

Given a partitioning T, let $t^{\mathbf{k}}$ represent a set of records $\{R_i \ldots R_j\}$ (e.g., $t^{\mathbf{k}}$ is a block of the partition). We define the *clusterwise scoring function* as the sum of scores for each cluster:

$$S_c(T, \Lambda) = \sum_{\mathbf{k}} s(f(t^{\mathbf{k}}), \Lambda)$$

- f function: column mean of each group’s feature matrix
- s function: linearly combine f function together using different parameters
- S function: sum up the score of each group

Main Algorithm

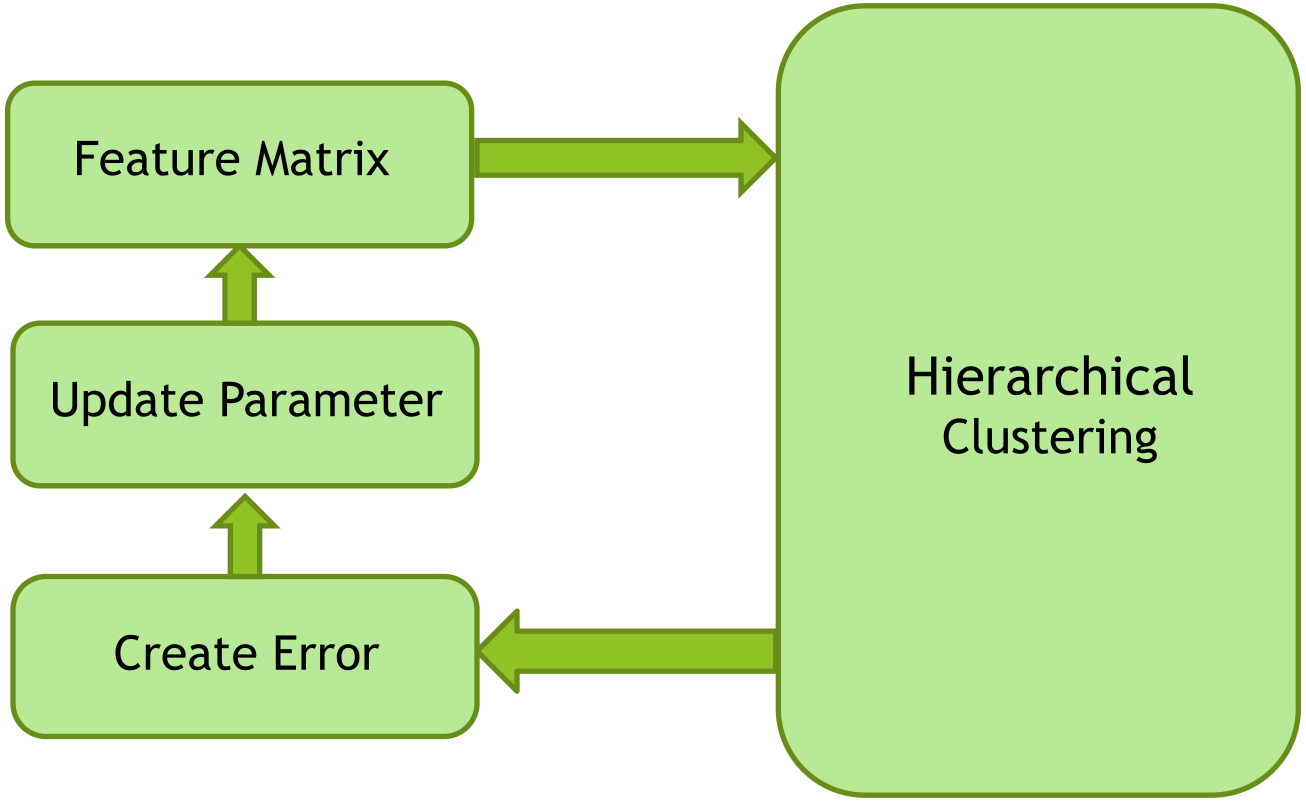


Algorithm 1 Error-driven Training Algorithm

```
Input:
    Training set $\mathcal{D}$
    Initial parameters $\Lambda^0$
    Prediction algorithm $\mathcal{A}$
while Not Converged do
    for all $\langle X, T^*(X) \rangle \in \mathcal{D}$ do
        $\mathbf{T}(X) \Leftarrow \mathcal{A}(X, \Lambda^t)$
        $\mathcal{D}_e \Leftarrow \mathit{CreateExamplesFromErrors}(\mathbf{T}(X), T^*(X))$
        $\Lambda^{t+1} \Leftarrow \text{UpdateParameters}(\mathcal{D}_e, \Lambda^t)$
    end for
end while
```

- Hierarchical Clustering
- Extract features of co-author, titles and journals by using the TF-IDF method

Process



Feature Matrix

Update Parameter

Create Error

Hierarchical Clustering

Create Example From Errors

- Step1: Find the first mistake made by the clustering (finding the number of unique label in one of clusters is larger than one)

- Step2: Randomly re-divide records based on the mistake. The divisions are called as neighbors of this mistake.

- Step3: (1) Find the neighbor reach the largest score function. (2) Find the neighbor reach the largest precision. If they are different, we will update the parameters. (The score function is not good enough)

Update Parameter

$$\Lambda^{t+1} = \underset{\Lambda}{\operatorname{argmin}} ||\Lambda^t - \Lambda||^2 \text{ s.t.}$$

$$S(N^*(T), \Lambda) - S(\hat{N}(T), \Lambda) \geq 1$$

$$S(\hat{N}, \Lambda) < \tau$$

- Make better neighbor have a higher score
- Change to the parameters should be minimal
- Using r package with can solve optimization problem with linear constraints

Evaluation

	paper 4 (combine)	paper 5(estimate)
f1	0.25	----
recall	0.25	----
precision	0.29	----
accuracy	0.81	> 0.81
time	< 4h	> 24h

Thank you for listening!