# Application of Dempster-Shafer theory to ensemble classification and user preferences


Van-Nam HUYNH

Japan Advanced Institute of Science and Technology
1-1 Asahidai, Nomi, Ishikawa, 923-1292 Japan
Email: huynh@jaist.ac.jp
http://www.jaist.ac.jp/~huynh/


$6^{th}$ School on Belief Functions and Their Applications
October 27-31, 2023, Ishikawa, Japan

# Purpose of This Talk

- After a brief recall of fundamental concepts of the Dempster-Shafer theory of evidence (DST), the issue of conflict in evidence combination is revisited.
- To briefly examine how DST could be applied in ensemble classification and recommendation systems.
- An integrated approach that combines machine learning (ML) techniques and DST for user preference modeling is discussed.

# Outline

# Outline

# Introduction

## Dempster-Shafer Theory

- Providing a general mechanism for representing and reasoning with uncertain information.
- Providing a proper way of quantifying ignorance and therefore a suitable framework for handling incomplete uncertain information.
- Providing a powerful tool for combining evidence from distinct sources of information.

## Reference


- A. Dempster, Upper and lower probabilities induced by a multi-valued mapping, *Ann. Math. Stat.* **38** (1967) 325–339.
- G. Shafer, *A Mathematical Theory of Evidence* (Princeton University Press, Princeton, 1976).

# Basic Notions

1. *Frame of Discernment*: a finite set $\Theta$ of mutually exclusive and exhaustive hypotheses, and $X$ is a variable on $\Theta$

2. *Mass function* (Basic Probability Assignment, BPA): $m : 2^\Theta \to [0, 1]$ verifying

   (i) $m(\varnothing) = 0$, and (ii) $\sum_{A \in 2^\Theta} m(A) = 1$.

- A subset $A$ of $\Theta$ such that $m(A) > 0$ is called a focal element of $m$.
- A mass function $m$ is often used to model a piece of evidence about variable $X$.
- The quantity $m(A)$ can be interpreted as a measure of the belief that is committed exactly to the proposition "$X \in A$".

# Examples

## 100 Marble Example:

A bag of 100 marbles: 30 red; 70 blue and yellow but the exact proportion of blue and yellow is not known.

- The frame of discernment: $\Theta = \{red, blue, yellow\}$
- The information that there are exactly 30 red marbles provides support in degree of 0.3 for $\{red\}$.
- The information that there are 70 blue and yellow marbles does not provide any positive support for either $\{blue\}$ or $\{yellow\}$, but does provide support in degree of 0.7 for $\{blue, yellow\}$.
- This can be modeled by the mass function: $m(\{red\}) = 0.3, m(\{blue, yellow\}) = 0.7$, and $m(A) = 0$ for any $A \in 2^{\Theta} \setminus \{\{red\}, \{blue, yellow\}\}$

# Outline

# Evidential Functions

## Belief Function

- Definition: $Bel_m(A) = \sum_{\varnothing \neq B \subseteq A} m(B)$ – the credibility of $A$
- Interpretation: total degree of justified belief in $A$.

## Plausibility Function

- Definition: $Pl_m(A) = \sum_{B \cap A \neq \varnothing} m(B)$ – the plausibility of $A$
- Interpretation: the degree to which the evidence fails to refute $A$.

# Discounting Operation

- Discounting allows us to take into account meta-knowledge about the reliability of a source of information.
- Assume that we have:
  - $m$ is a mass function provided by a source of information $S$.
  - Meta-knowledge: probability that "the source $S$ is reliable" is $\alpha$.
- Then, discounting $m$ at a discount rate of $(1-\alpha)$ yields the following mass function (denoted by $m^\alpha$):

$$m^\alpha(A) = \begin{cases} \alpha \times m(A), & \text{if } A \neq \Theta; \\ \alpha \times m(\Theta) + (1-\alpha), & \text{if } A = \Theta \end{cases}$$

Note: $m^1 = m$; and $m^0 = m_\Theta$.

# Dempster's Rule of Combination

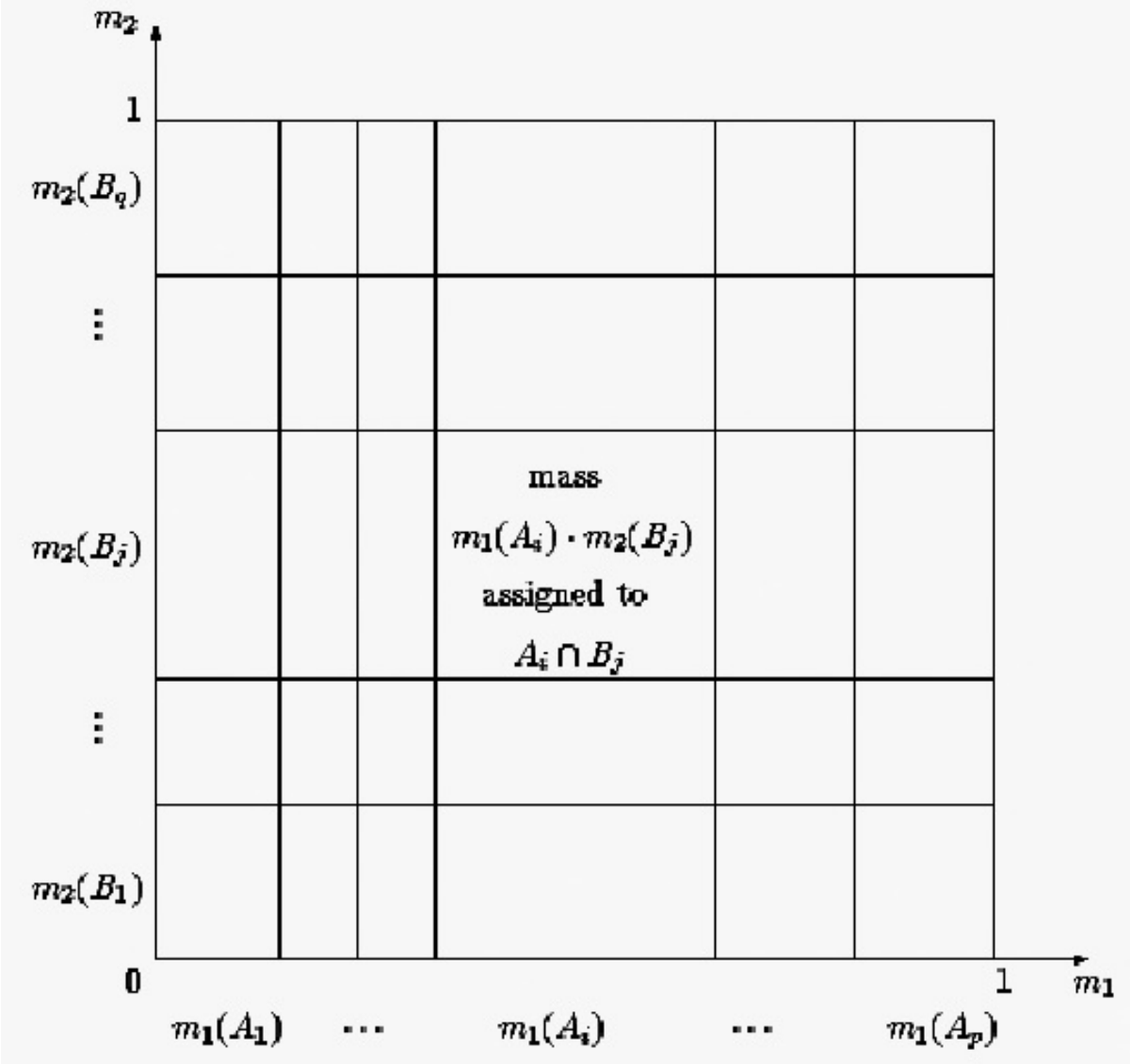
- Let $m_1$ and $m_2$ be two mass functions on $\Theta$ induced by two distinct sources of information.
- Dempster's rule of combination:

$$(m_1 \oplus m_2)(A) = \begin{cases} 0, & \text{if } A = \varnothing \\ \frac{1}{1-\kappa} \sum\limits_{B \cap C = A} m_1(B) \times m_2(C), & \text{if } A \neq \varnothing \end{cases}$$

where $\kappa = \sum\limits_{B \cap C = \varnothing} m_1(B) \times m_2(C)$ – degree of conflict.

- Properties:
  - Commutative, associative.
  - Neutral element – $m_\Theta$ (represents total ignorance): $m_\Theta \oplus m = m$.

# Dempster's Rule of Combination



# Remark on Dempster's Rule for Evidence Combination

- Criticisms on the counterintuitive results of applying Dempster's combination rule to conflicting beliefs soon emerged since its inception.
- In Dempster's rule of combination, the combined mass assigned to the empty set considered as the conflict is distributed proportionally to the other masses.
- Zadeh (1984) presented an example where Dempster's rule of combination produces unsatisfactory results.
- Since then, many alternatives have been proposed in the literature.
- The study of combination rules in DS theory when evidence is in conflict remains an interesting topic, especially in data/information fusion applications.

# Zadeh's Example

- One doctor believes a patient has either *meningitis* – with a probability of 0.99, or a *brain tumor* – with a probability of only 0.01.
- A second doctor believes the patient suffers from *concussion* – with a probability of 0.99, and also believes the patient has a *brain tumor* – with a probability of only 0.01.

Combining these two pieces of evidence with Dempster's rule yields

$$m_{\oplus}(\text{brain tumor}) = Bel_{\oplus}(\text{brain tumor}) = 1$$

✘ This result implies complete support for the diagnosis of a *brain tumor*, which both doctors believed very unlikely.

➪ Many alternative rules of combination have been developed.

# Remarks

- Many other suggestions have been made, creating a "jungle" of combination rules.
- Most of these works usually began with analyzing some counterintuitive examples when applying existing combination rules, and then proposed new ones which would give more reasonable results to these particular situations.
- This approach may only yield solutions being good locally, and consequently, it is difficult to be theoretically justified.

# Outline

# $m_\oplus(\varnothing)$ as Conflict?

**[Liu, AI 2006]**

Liu (2006) argued that value $m_\oplus(\varnothing)$ cannot be used as a measure of conflict between two bodies of evidence but only represents the mass of uncommitted belief as a result of combination.

Example – Two identical mass functions

Let us consider two **identical** mass functions $m_1 = m_2$ on $\Theta = \{\theta_i\}_{i=1}^5$:

- $m_1(\theta_i) = m_2(\theta_i) = 0.2$ for $i = 1, \ldots, 5$
- Then, $m_\oplus(\varnothing) = 0.8$, which is quite high whilst it appears the total absence of conflict as two mass functions are identical.

**Remark:**

More generally, we always get $m_\oplus(\varnothing) > 0$ with two identical mass function whenever their focal elements defines a partition of the frame.

# Liu's Criteria for Conflict

**[Liu, AI 2006]**

Two mass functions $m_1$ and $m_2$ are said to be in conflict if and only if

$$m_{\oplus}(\varnothing) > \epsilon \quad \text{and} \quad \mathrm{difBetP}(m_1, m_2) > \epsilon$$

where $\epsilon \in [0, 1]$ is a threshold of conflict tolerance and $\mathrm{difBetP}(m_1, m_2)$ is defined by

$$\mathrm{difBetP}(m_1, m_2) = \max_{A \subseteq \Theta}(|BetP_{m_1}(A) - BetP_{m_2}(A)|)$$

and called the **distance between betting commitments** of the two mass functions.

☞ For a comprehensive analysis of combination rules and conflict management, see [P. Smets, *Information Fusion* **8** (2007)].

# Liu's Criteria for Conflict

**[Liu, AI 2006]**

Example: Consider the following pair of mass functions on the same frame $\Theta = \{\theta_i | i = 1, \ldots, 7\}$

$$m_1(\{\theta_1, \theta_2, \theta_3, \theta_4\}) = 1; \text{ and } m_2(\{\theta_4, \theta_5, \theta_6, \theta_7\}) = 1$$

Then, $m_\oplus(\varnothing) = 0$, i.e, these mass functions are not in conflict at all. However, using the second criterion we easily get:

$$\mathrm{difBetP}(m_1, m_2) = 0.75$$

Note that $m_1$ and $m_2$ have assigned, by definition, the total mass exactly to $\{\theta_1, \theta_2, \theta_3, \theta_4\}$ and $\{\theta_4, \theta_5, \theta_6, \theta_7\}$, respectively, and to none of the proper subsets of them. So intuitively these two mass functions are partly in conflict. Such a partial conflict does not be judged by means of $m_\oplus(\varnothing)$ but $\mathrm{difBetP}(m_1, m_2)$ as shown above.

# Distance Between two Mass Functions

- Let $\mathcal{B}_1 = (\mathcal{F}_{m_1}, m_1)$ and $\mathcal{B}_2 = (\mathcal{F}_{m_2}, m_2)$ be two bodies of evidence on the same frame $\Theta$.
- Denote $\mathrm{dif}_{\mathcal{F}}(m_1, m_2)$ the **symmetric difference** between two families of focal elements $\mathcal{F}_{m_1}$ and $\mathcal{F}_{m_2}$, i.e.,

$$\mathrm{dif}_{\mathcal{F}}(m_1, m_2) = (\mathcal{F}_{m_1} \setminus \mathcal{F}_{m_2}) \cup (\mathcal{F}_{m_2} \setminus \mathcal{F}_{m_1})$$

# Difference Between two BoEs

- If $\mathrm{dif}_{\mathcal{F}}(m_1, m_2) = \mathcal{F}_{m_1} \cup \mathcal{F}_{m_2}$, and $A \cap B = \varnothing$ for any $A \in \mathcal{F}_{m_1}$ and $B \in \mathcal{F}_{m_2}$, then $m_{\oplus}(\varnothing) = 1$ – fully conflict.

- If $\mathrm{dif}_{\mathcal{F}}(m_1, m_2) = \varnothing$ and $d(m_1, m_2) > 0$, then qualitatively two sources are not in conflict but having different preferences in distributing their masses to focal elements.

⇨ How different between two sources in realization of the question of where the true hypothesis lies.

# Difference Between two BoEs

- Liu's criterion of using $\mathrm{difBetP}(m_1, m_2)$ is somewhat weaker than using the direct distance of $d(m_1, m_2)$.

Example: consider again the following pair of mass functions:

$$m_1(\{\theta_1, \theta_2, \theta_3, \theta_4\}) = 1; \text{ and } m_2(\{\theta_4, \theta_5, \theta_6, \theta_7\}) = 1$$

Then, we have $d(m_1, m_2) = 1$ whilst $\mathrm{difBetP}(m_1, m_2) = 0.75$.

- In addition, if $m_1 = m_2$ we have $\mathrm{difBetP}(m_1, m_2) = 0$ but the reverse does not hold in general.

# Quantifying Conflict

- We have argued that only a part of value $m_\oplus(\varnothing)$ should be used to quantify a conflict qualitatively stemming from $\mathrm{dif}_\mathcal{F}(m_1, m_2)$.
- Let

$$m_\oplus^{\mathrm{comb}}(\varnothing) = \sum_{A,B \in \mathcal{F}_1 \cap \mathcal{F}_2, A \cap B = \varnothing} m_1(A) m_2(B)$$

- Clearly, $m_\oplus^{\mathrm{comb}}(\varnothing)$ is a part of $m_\oplus(\varnothing)$ and intuitively representing the mass of uncommitted belief as a result of combination rather than a conflict.
- Therefore, the conflict is properly represented by the remainder of $m_\oplus(\varnothing)$, i.e.

$$m_\oplus(\varnothing) - m_\oplus^{\mathrm{comb}}(\varnothing) \triangleq m_\oplus^{\mathrm{conf}}(\varnothing)$$

# Quantifying Conflict

**Remark**

With this formulation of conflict, the fact used to question the validity of Dempster's rule that two identical probability measures are always conflicting becomes inappropriate.

**Example**

Consider again two identical mass functions on $\Theta = \{\theta_i | i = 1 \ldots 5\}$: $m_1(\theta_i) = m_2(\theta_i) = 0.2$ for $i = 1, \ldots, 5$. Then we get $m_{\oplus}^{\text{comb}}(\varnothing) = 0.8$ and $m_{\oplus}^{\text{conf}}(\varnothing) = 0$, and hence no conflict appears between the two at all.

- Generally, we always get $m_{\oplus}^{\text{conf}}(\varnothing) = 0$ whenever two mass functions being combined are identical.

# Quantifying Conflict

## Zadeh's example revisited

Consider two mass functions $m_1$ and $m_2$ defined on $\Theta = \{a, b, c\}$ as:

- $m_1(a) = 0.99$, $m_1(b) = 0.01$
- $m_2(c) = 0.99$, $m_2(b) = 0.01$
- Then we get $m_{\oplus}^{\text{conf}}(\varnothing) = 0.98$, which accurately reflects a very high conflict between the two sources of evidence.

## Remark

With such a high conflict but still assuming both sources are fully reliable to proceed with directly applying Demspter's rule on them (to get 'unsatisfactory' results) seems irrational.

# Outline

# A Solution to Conflict

- According to Smets' two-level view of evidence (Smets, 1994), to make decisions based on evidence, beliefs encoding evidence must be transformed into probabilities using the so-called **pignistic transformation**.
- Guided by this view, we propose to discount a mass function involving in combination based upon **how sure in its decision** when it is used alone for decision making.
- More particularly, we provide a method for defining discount rates of mass functions being combined using the entropy of their corresponding pignistic probability functions.

# A Solution to Conflict

## Ambiguity Measure:

- Let $m_1$ and $m_2$ be two mass functions on the frame $\Theta$ and $BetP_{m_1}$ and $BetP_{m_2}$ be **pignistic probability functions** of $m_1$ and $m_2$, respectively.
- For $i = 1, 2$, we denote

$$H(m_i) = -\sum_{\theta \in \Theta} BetP_{m_i}(\theta) \log_2(BetP_{m_i}(\theta))$$

  the Shannon entropy expression of pignistic probability distribution $BetP_{m_i}$.

- This measure has been used in Jousselme *et al* (2006) as an ambiguity measure of belief functions.
- Clearly, $H(m_i) \in [0, \log_2(|\Theta|)]$.

# A Solution to Conflict

**Entropy-based Discount Rate:**

- The discount rate of mass function $m_i$ $(i = 1, 2)$, denoted $\delta(m_i)$, is defined by

$$\delta(m_i) = \frac{H(m_i)}{\log_2(|\Theta|)}$$

- That is, the higher uncertainty (in its decision) a source of evidence is, the higher discount rate it is applied.

**General Discounting and Combination Rule:**

$$m_{\oplus} = m_1^{(1-\delta(m_1))} \oplus m_2^{(1-\delta(m_2))}$$

where $\oplus$ is a combination operator in general and $m_i^{(1-\delta(m_i))}$ is the discounted mass function obtaining from $m_i$.

# Outline

# DS Theory and Its Applications

- DS theory has been theoretically well studied and widely applied to such areas of application as
  - Classification, Identification, Recognition
  - Decision Making, Expert Systems
  - Fault Detection and Failure Diagnosis
  - Image Processing, Medical Applications
  - Risk and Reliability
  - Robotics, Multiple Sensors
  - Signal Processing
  - Etc.

# DS theory's Application in ML

- During the last decades, DS theory has been actively applied in Machine Learning (ML) for

  ✓ developing so-called **evidential ML methods** [Denoeux, 1995; Zouhal & Denoeux, 1998; Denoeux & Masson, 2004; Lian *et al*, 2015; Li *et al*, 2018; Tong *et al*, 2021].

  ✓ **combining multiple classifiers** (ensemble learning) [Xu *et al*., 1992; Rogova, 1994; Al-Ani & Deriche, 2002; Quost *et al*, 2007; Huynh *et al*., 2010; Bi, 2012; Wang *et al*, 2020; Fu *et al*, 2021], and

  ✓ **recommendation systems** [Wickramarathne *et al*, 2009; Wickramarathne *et al* 2011; Nguyen & Huynh, 2014; 2015; 2017]; Nguyen *et al*, 2017; 2020].

# Outline

# Classifier Combination

As observed in studies of machine learning systems:

- the set of patterns misclassified by different classification systems would **not necessarily overlap**.
- different classifiers potentially offer complementary information about patterns to be classified.

**Remark:** The observation highly motivated the interest in combining classifiers during the last two decades (Kittler et al., IEEE PAMI 1998).

**Combination Scenarios:**

- All classifiers use the same representation of the input
- Each classifier uses its own representation of the input

# Classifier Combination

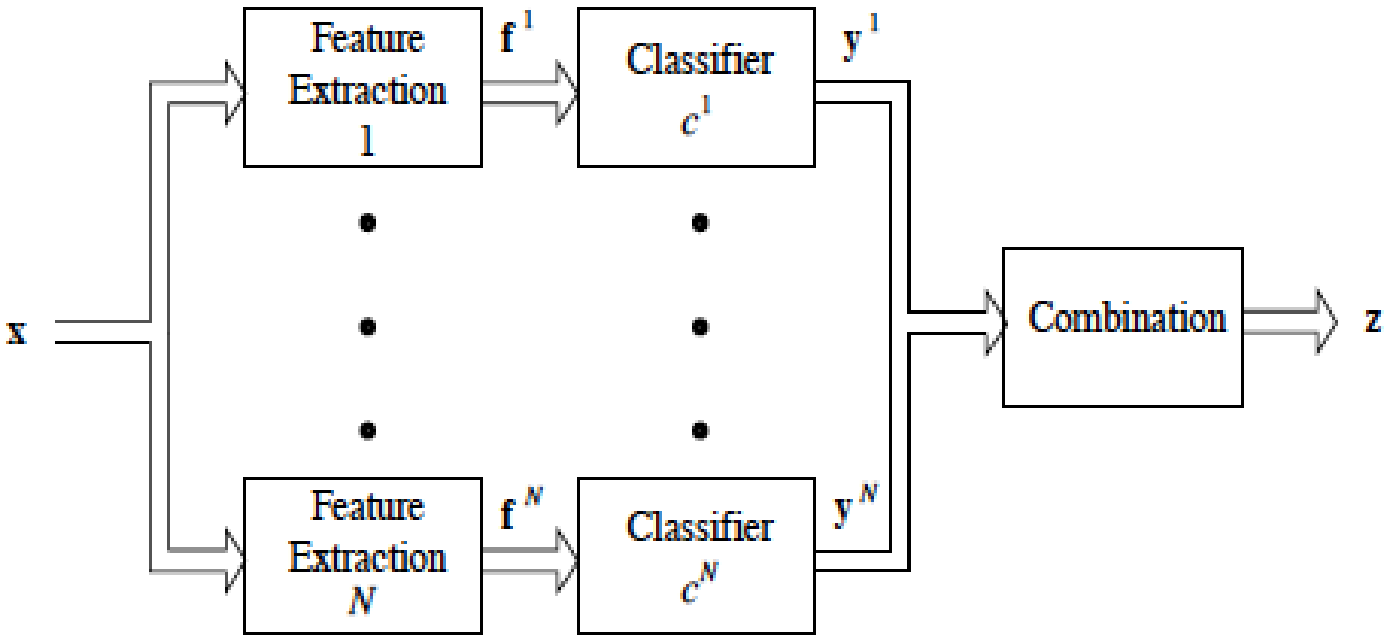



**Figure:** Classifier fusion using different feature sets (Al-Ani & Deriche, 2002)

# DS theory in Classifier Combination

- Application of DS theory to classifier combination has received attention since early 1990s.
- In the context of single-class classification problem, the frame of discernment is often modeled by the set of all possible classes used to assign to an input pattern.
- Given an input pattern, each individual classifier produces an output considered as a source of information serving for classification of the input pattern.
- These sources of information from all classifiers participating in the combination process will be combined to make the final decision on the classification.

# DS theory in Classifier Combination

- Let $\mathcal{C} = \{c_1, c_2, \ldots, c_M\}$ be the set of classes – the frame of discernment of the problem.
- Assume that we have $R$ classifiers: $\{\psi_1, \ldots, \psi_R\}$.
- For an input $\mathbf{x}$, each classifier $\psi_i$ produces an output $\psi_i(\mathbf{x})$ defined as

$$\psi_i(\mathbf{x}) = [s_{i1}, \ldots, s_{iM}]$$

where $s_{ij}$ indicates the degree of **confidence or support** in saying that "the pattern $\mathbf{x}$ is assigned to class $c_j$ according to classifier $\psi_i$."

- ➪ Note that $s_{ij}$ can be a binary value or a continuous numeric value and its semantic interpretation depends on what type of learning algorithm used to build $\psi_i$.

# Xu's Combination Method

- Each individual classifier produces a crisp decision on classifying an input $\mathbf{x}$, which is used as the evidence come from the corresponding classifier.
- Then this evidence is associated with prior knowledge defined in terms of performance indexes of the classifier to define its corresponding mass function.
- Performance indexes of a classifier are defined by recognition, substitution and rejection rates obtained by testing the classifier on a test sample set.

✓ Reference: Xu et al., Several methods for combining multiple classifiers and their applications in handwritten character recognition. *IEEE Trans. SMC* **22** (1992).

# Xu's Combination Method

- Let the **recognition rate** and **substitution rate** of $\psi_i$ be $\epsilon_r^i$ and $\epsilon_s^i$ (usually $\epsilon_r^i + \epsilon_s^i < 1$, due to the rejection action), respectively
- The mass function $m_i$ from $\psi_i(\mathbf{x})$ is defined by
  1. If $\psi_i$ rejected $\mathbf{x}$, i.e. $\psi_i(\mathbf{x}) = [0, \ldots, 0]$, $m_i$ has only a focal element $\mathcal{C}$ with $m_i(\mathcal{C}) = 1$.
  2. If $\psi_i(\mathbf{x}) = [0, \ldots, 0, s_{ij} = 1, 0, \ldots, 0]$, then $m_i(\{c_j\}) = \epsilon_r^i$, $m_i(\neg\{c_j\}) = \epsilon_s^i$, where $\neg\{c_j\} = \mathcal{C} \setminus \{c_j\}$, and $m_i(\mathcal{C}) = 1 - \epsilon_r^i - \epsilon_s^i$.
- In a similar way one can obtain all $m_i$ $(i = 1, \ldots, R)$ from $R$ classifiers $\psi_i$ $(i = 1, \ldots, R)$.
- Then Dempster's rule is applied to combine these $m_i$'s to obtain a combined $m = m_1 \oplus \ldots \oplus m_R$, which is used to make the final decision on the classification of $\mathbf{x}$.

# Rogova's Combination Method

- Used a proximity measure between a **reference vector** of each class and a classifier's output vector.
- The reference vector is the **mean vector** $\mu_j^i$ of the output set of each classifier $\psi_i$ for each class $c_j$.
- Then, for any input pattern $\mathbf{x}$, the proximity measures $d_j^i = \phi(\mu_j^i, \psi_i(\mathbf{x}))$, $j = 1, \ldots, M$, are transformed into the following mass functions:

$$\begin{aligned} m_j^i(\{c_j\}) &= d_j^i, \quad m_j^i(\mathcal{C}) = 1 - d_j^i \\ m_{\neg j}^i(\neg\{c_j\}) &= 1 - \prod_{k \neq j}(1 - d_k^i), \quad m_{\neg j}^i(\mathcal{C}) = \prod_{k \neq j}(1 - d_k^i) \end{aligned}$$

which together constitute the knowledge about $c_j$ from $\psi_i$.

# Rogova's Combination Method

- Hence, these $m^i_j$ and $m^i_{\neg j}$ are combined to define the evidence from classifier $\psi_i$ on classifying $\mathbf{x}$ as $m^i = m^i_j \oplus m^i_{\neg j}$:

$$m^i(\{c_j\}) = \frac{d^i_j \prod_{k \neq j}(1 - d^i_k)}{1 - d^i_j[1 - \prod_{k \neq j}(1 - d^i_k)]}$$

$$m^i(\neg\{c_j\}) = \frac{(1 - d^i_j)[1 - \prod_{k \neq j}(1 - d^i_k)]}{1 - d^i_j[1 - \prod_{k \neq j}(1 - d^i_k)]}$$

$$m^i(\mathcal{C}) = \frac{\prod_{k}(1 - d^i_k)}{1 - d^i_j[1 - \prod_{k \neq j}(1 - d^i_k)]}$$

# Rogova's Combination Method

- Finally, all evidences from all classifiers are combined using Dempster's rule to obtain an overall mass function for making the final decision on the classification for $\mathbf{x}$.

✓ Reference: Rogova, Combining the results of several neural network classifiers. *Neural Networks* **7** (1994).

# Al-Ani & Deriche's Combination Method

- The distance between the output classification vector provided by each single classifier and a reference vector is used to estimate mass functions.
- These mass functions are then combined using Dempster's rule to obtain a new output vector that represents the combined confidence in each class label.
- However, instead of defining a reference vector as the mean vector of the output set of a classifier for a class as in Rogova's work, it is measured such that the mean square error (MSE) between the new output vector obtained after combination and the target vector of a training data set is minimized.

⇨ This interestingly makes their combination algorithm **trainable**.

# Al-Ani & Deriche's Combination Method

- Given an input $\mathbf{x}$, the mass function $m_i$ derived from classifier $\psi_i$ is defined as follows:

$$m_i(\{c_j\}) = \frac{d_i^j}{\sum_{k=1}^{M} d_i^k + g_i}$$

$$m_i(\mathcal{C}) = \frac{g_i}{\sum_{k=1}^{M} d_i^k + g_i}$$

where $d_i^j = \exp(-\|\mathbf{v}_j^i - \psi_i(\mathbf{x})\|^2)$, $\mathbf{v}_j^i$ is a reference vector and $g_i$ is an ignorance coefficient of $\psi_i$.

- Both $\mathbf{v}_j^i$ and $g_i$ are estimated via the minimized MSE learning process.

✓ Reference: Al-Ani & Deriche, A new technique for combining multiple classifiers using the Dempster-Shafer theory of evidence, *Journal of Artificial Intelligence Research* **17** (2002).

# Bell's Combination Method

- A new method for representing and combining outputs from different classifiers for text categorization.
- Different from all the above mentioned methods, Bell et al. (2005) directly used outputs of individual classifiers to define the so-called 2-points focused mass functions.
- Given an input $\mathbf{x}$, the output $\psi_i(\mathbf{x})$ from classifier $\psi_i$ is normalized:

$$p_i(c_j) = \frac{s_{ij}}{\sum_{k=1}^{M} s_{ik}}, \text{ for } j = 1, \ldots, M$$

- Then the collection $\{p_i(c_j)\}_{j=1}^{M}$ is arranged so that

$$p_i(c_{i_1}) \geq p_i(c_{i_2}) \geq \ldots \geq p_i(c_{i_M})$$

# Bell's Combination Method

- The mass function $m_i$ induced from $\psi_i$ on the classification of $\mathbf{x}$:

$$\begin{aligned} m_i(\{c_{i_1}\}) &= p_i(\{c_{i_1}\}) \\ m_i(\{c_{i_2}\}) &= p_i(\{c_{i_2}\}) \\ m_i(\mathcal{C}) &= 1 - m_i(\{c_{i_1}\}) - m_i(\{c_{i_2}\}) \end{aligned}$$

- This mass function is called the 2-points focused mass function and the set $\{\{c_{i_1}\}, \{c_{i_2}\}, \mathcal{C}\}$ is referred to as a **triplet**.
- These 2-points focused mass functions are then combined using Dempster's rule to obtain an overall mass function for making the final classification decision.

✓ Reference: Bell, Guan & Bi, On combining classifiers mass functions for text categorization, *IEEE Trans. KDE* **17** (2005).
Y. Bi, The impact of diversity on the accuracy of evidential classifier ensembles. *Int. J. Approx. Reasoning* **53** (2012).

# Discounting+Combination Method

- Built Naive Bayes classifiers corresponding to distinct representations of the input.
- Then weighted them by their accuracies obtained by testing with a test sample set, where weighting is modeled by the discounting operator.
- Finally, discounted mass functions are combined to obtain the final mass function which is used for making the classification decision.

✓ Reference: Le, Huynh, Shimazu & Nakamori, Combining classifiers for word sense disambiguation based on Dempster-Shafer theory and OWA operators, *Data & Knowledge Engineering* **63** (2007).

# Discounting+Combination Method

- Let $\mathbf{f}_i$ be the $i$-th representation of an input $\mathbf{x}$ and classifier $\psi_i$ building on $\mathbf{f}_i$ produces a posterior probability distribution $P(\cdot|\mathbf{f}_i)$ on $\mathcal{C}$.
- Assume that $\alpha_i$ is the weight of $\psi_i$ defined by its accuracy.
- Then the piece of evidence represented by $P(\cdot|\mathbf{f}_i)$ is discounted at a discount rate of $(1-\alpha_i)$, resulting in a mass function $m_i$ defined by

$$\begin{aligned} m_i(\{c_j\}) &= \alpha_i \times P(c_j|\mathbf{f}_i), \text{ for } j = 1, \ldots, M \\ m_i(\mathcal{C}) &= 1 - \alpha_i \end{aligned}$$

- These discounted mass functions are then combined using either Dempster's rule or averaging operator.

# Remarks

- This method of weighting clearly focuses on only the strength of individual classifiers, which is defined by testing them on the designed sample data set.
- Therefore it does not be influenced by an input pattern under classification.
- However, the information quality of soft decisions or outputs provided by individual classifiers might vary from pattern to pattern.

⇨ The general discounting and combination strategy for solving conflict discussed above has been applied to classifier combination.

# Revised Discounting+Combination Method

- Let us denote $m_i(\cdot|\mathbf{x})$ the probability distribution $\psi_i(\mathbf{x})$ on $C$, i.e. $m_i(c_j|\mathbf{x}) = s_{ij}(\mathbf{x})$.
- The weight associated with $\psi_i$ regarding the classification of $\mathbf{x}$ is defined by

$$w_i(\mathbf{x}) = 1 - \frac{H(m_i(\cdot|\mathbf{x}))}{\log(M)}$$

  where $H$ is Shannon entropy expression of the probability distribution $m_i(\cdot|\mathbf{x})$.
- **Note:** This definition of a classifier weight essentially depends on the input $\mathbf{x}$ under consideration, then the weight of an individual classifier can vary differently from pattern to pattern depending on how ambiguity associated with its decision on the classification of a particular pattern.

# Revised Discounting+Combination Method

- Then, an overall mass function $m(\cdot|\mathbf{x})$ can be formulated in the general form of the following:

$$m(\cdot|\mathbf{x}) = \bigoplus_{i=1}^{R} (w_i(\mathbf{x}) \otimes m_i(\cdot|\mathbf{x}))$$

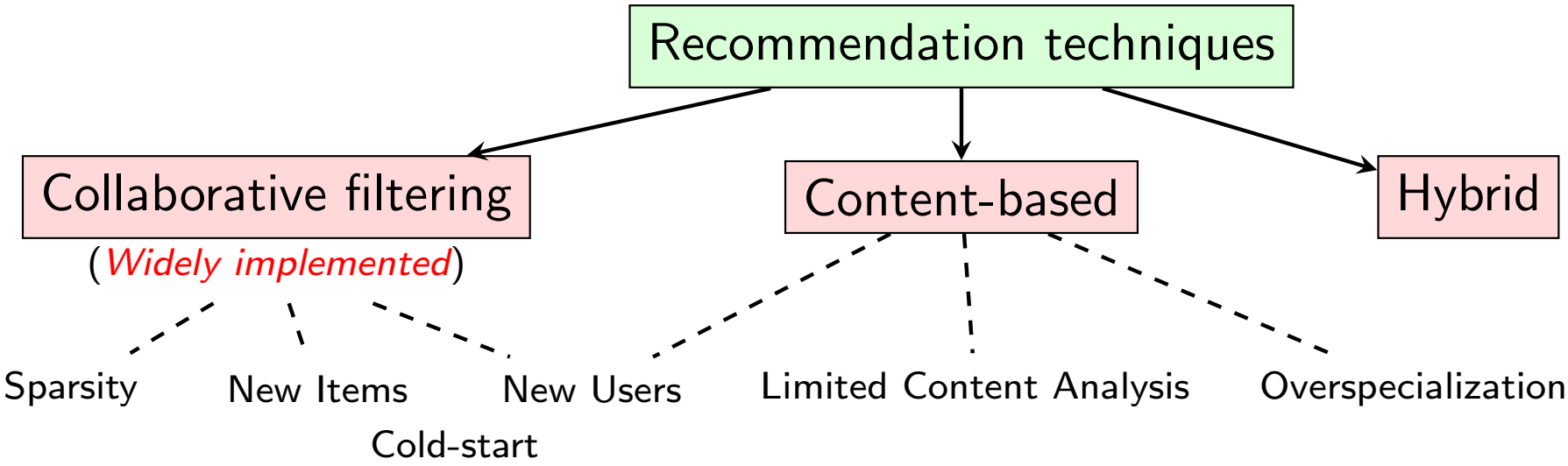
  where $\otimes$ is the discounting operator and $\oplus$ is a combination operator in general.
- Under such a general formulation, using different combination operators in DS theory we can obtain different decision rules for the classification of $\mathbf{x}$.

✓ Reference: Huynh, Nguyen & Le, Adaptively entropy-based weighting classifiers in combination using Dempster-Shafer theory for word sense disambiguation, *Computer Speech and Language* **24** (2010).

# Outline

# Recommender Systems

- RSs were introduced in 1990s
- Classification of RSs [Adomavicius & Tuzhilin, *IEEE Trans. Knowl. Data Eng.*, 2005]




- Most RSs allow users to express their preferences as hard ratings

# Hard Ratings

- A **hard rating** is known as a single value in the rating domain
  - A rating domain containing 5 elements $\Theta = \{1, 2, 3, 4, 5\}$
  - A hard rating can be $\theta = 3$


- Each hard rating may encode qualitative, subjective, and imperfect information inside
- In some cases, hard ratings may be not suitable
  - Rated: user $U_1 \leftarrow 3$; user $U_2 \leftarrow 4$
  - How to represent the preference of user $U_3$, who may partly agree with both $U_1$ and $U_2$ or somehow in between them?

# Soft Ratings

- Being used for the purpose of capturing and modeling qualitative, subjective, and imperfect information
- A soft rating is known as a subset of a rating domain
- Using soft ratings is considered as a more realistic and flexible way to represent user preferences
  - Rated: $U_1 \leftarrow \{3\}$; $U_2 \leftarrow \{4\}$
  - Representing the preference of $U_3$: $(\{3, 4\}, 1.0)$ or $\{(\{3\}, 0.3), (\{4\}, 0.7)\}$
- RSs offering soft ratings were developed based on Dempster-Shafer theory.

# Soft Ratings

- For example, in a RS with a rating domain $\Theta = \{1, 2, 3, 4, 5\}$
- Possible answers for a rating request would be:

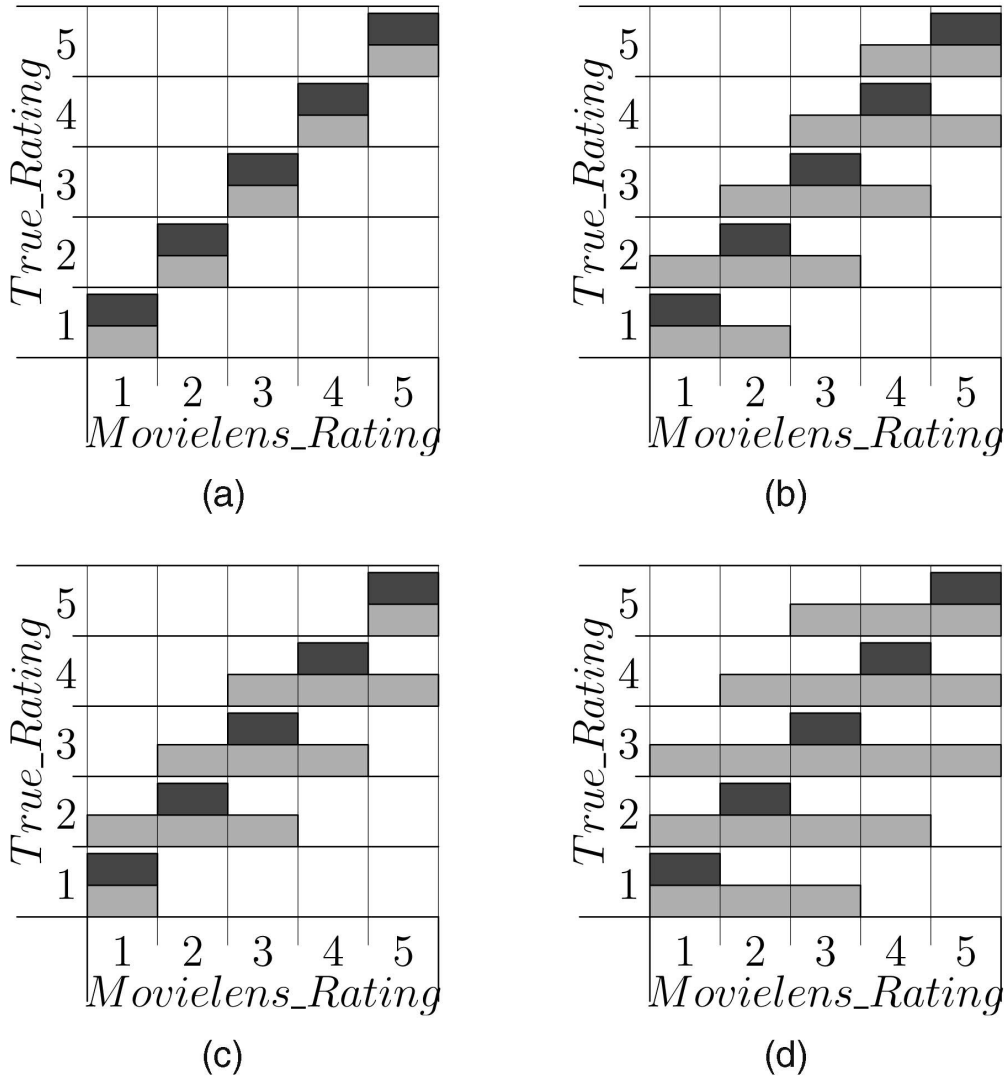
**(1)** *I'll rate it as 4 and I am sure about it* (precise, certain)
**(2)** *I'll rate it as 4 and I am 90% sure about it* (precise, uncertain)
**(3)** *I'll rate it at least 4 and I am sure about it* (imprecise, certain)
**(4)** *I'll rate it at least 4 and I am 90% sure about it* (imprecise,uncertain)
**(5)** *I'll not rate it now* (ignorance)

- The corresponding soft ratings

**(1)** $r_1(\{4\}) = 1.0$
**(2)** $r_2(\{4\}) = 0.9; r_2(\Theta) = 0.1$
**(3)** $r_3(\{4, 5\}) = 1.0$
**(4)** $r_4(\{4, 5\}) = 0.9; r_4(\Theta) = 0.1$
**(5)** $r_5(\Theta) = 1.0$

# Soft Ratings

**[Wickramarathne et al., IEEE Trans. Knowl. Data Eng., 2011]**



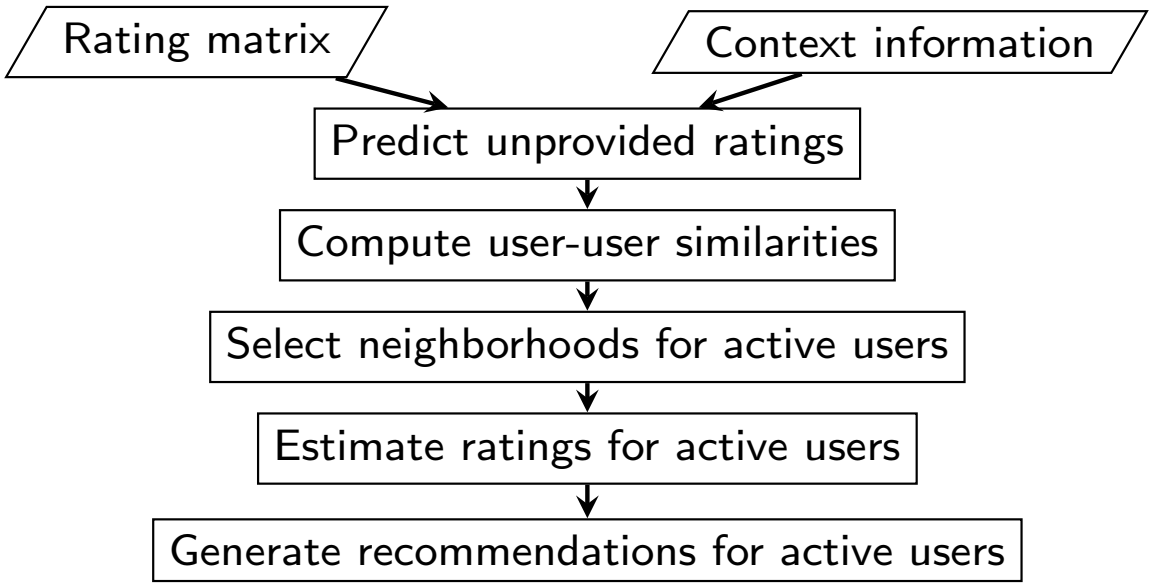
# CoFiDS: A Belief-Theoretic Approach for Automated Collaborative Filtering

**[Wickramarathne et al., IEEE Trans. Knowl. Data Eng., 2011]**

- User preferences modeling based on DS-theoretic framework
- Incorporation of contextual information for the prediction of unrated items to overcome the sparsity problem
- User-user similarity based on the distance between user-BoEs (users' bodies of evidence)
- User neighborhood determined using the $K$-nearest neighbor (KNN) strategy
- Collaborative filtering based recommendation

# CoFiDS: A Belief-Theoretic Approach for Automated Collaborative Filtering

**[Wickramarathne et al., IEEE Trans. Knowl. Data Eng., 2011]**




- Predicted ratings are considered the same as provided ones
- Could not predict all unprovided ratings
- The cold-start problem has not been discussed

# Integrating Social Networks into DST-Based RSs

## Using Community Context Information

- Social Networks


*image source: www.123rf.com*

- Communities

*image source: wordpress.com*

# Integrating Social Networks into DST-Based RSs

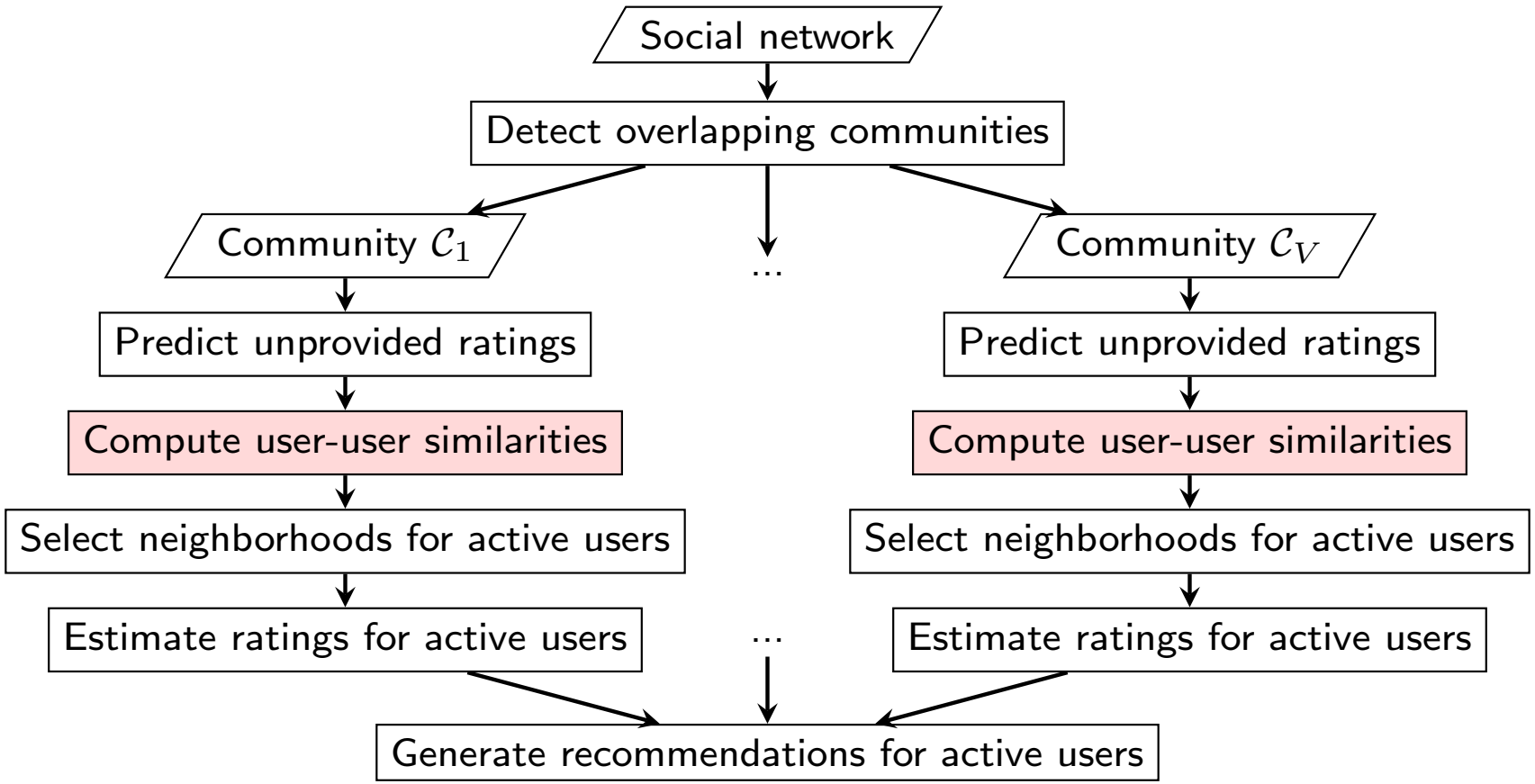
## Using Community Context Information

- A new approach to overcome the sparsity problem by using community context information.
- A new method for computing user-user similarities, in which provided ratings are weighted more important than predicted ratings.

✓ Reference: [Nguyen & Huynh, PRICAI 2014]; [Nguyen & Huynh, ECSQARU 2015]; [Nguyen *et al.*, IEEE Trans. SMC, 2020]

# Integrating Social Networks into DST-Based RSs

## Using Community Context Information



# Integrating Social Networks into DST-Based RSs

## Using Community Context Information

Data Sets & Assessment Methods

- MovieLens
  - Rating domain $\Theta = \{1, 2, 3, 4, 5\}$
  - 100,000 hard ratings, 943 users, 1682 movies
- Flixster
  - Rating domain $\Theta = \{0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0\}$
  - 535,013 hard ratings, 3827 users, 49410 friend relationships, 1210 movies
- Assessment methods
  - $MAE$, $Precision$, $Recall$, $F_\beta$ [Herlocker et al., *ACM Trans. Inf. Syst.*, 2004]
  - $DS\text{-}Precision$, $DS\text{-}Recall$ [Hewawasam et al., IEEE Trans. Syst. Man Cybern., 2007]
  - $DS\text{-}MAE$ , $DS\text{-}F_\beta$ [Wickramarathne et al., IEEE Trans. Knowl. Data Eng., 2011]
- CoFiDS was selected as a baseline

# Integrating Social Networks into DST-Based RSs

## Using Community Context Information

Comparative results for MovieLens

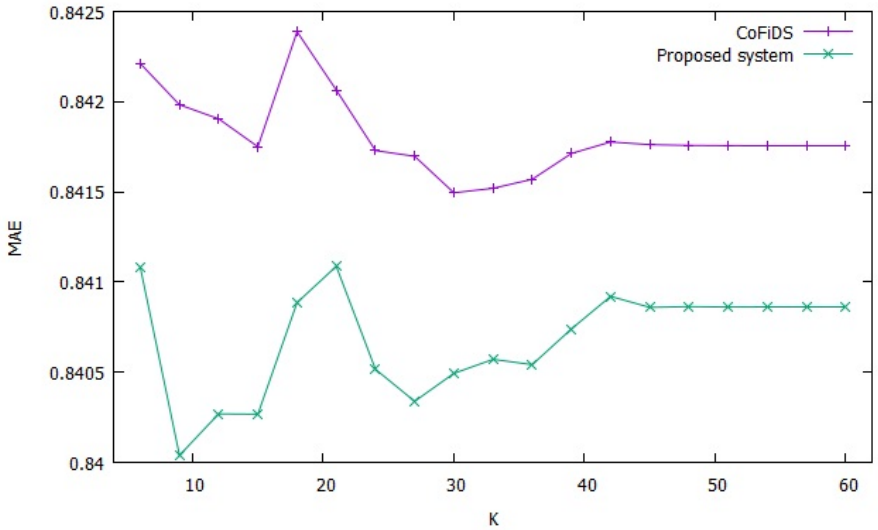



**Figure:** Overall $MAE$ versus $K$

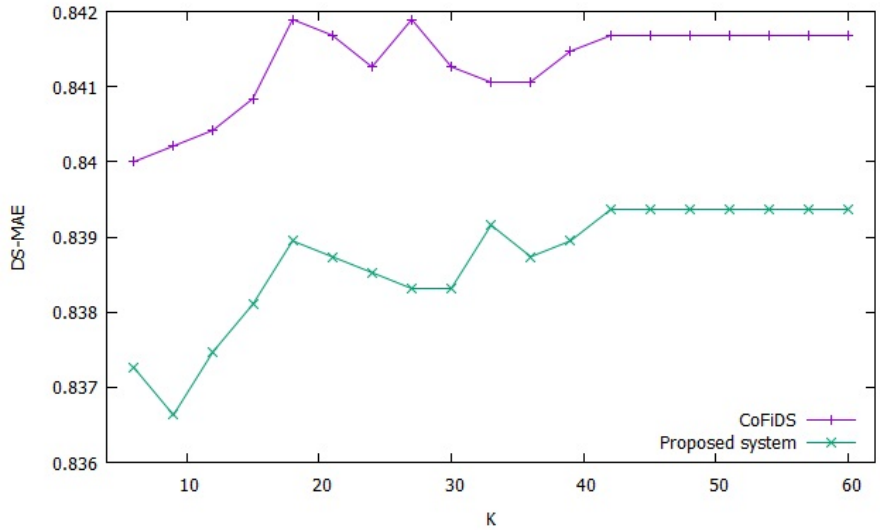



**Figure:** Overall $DS\text{-}MAE$ versus $K$

# Integrating Social Networks into DST-Based RSs

## Using Community Context Information

Comparative results for Flixster

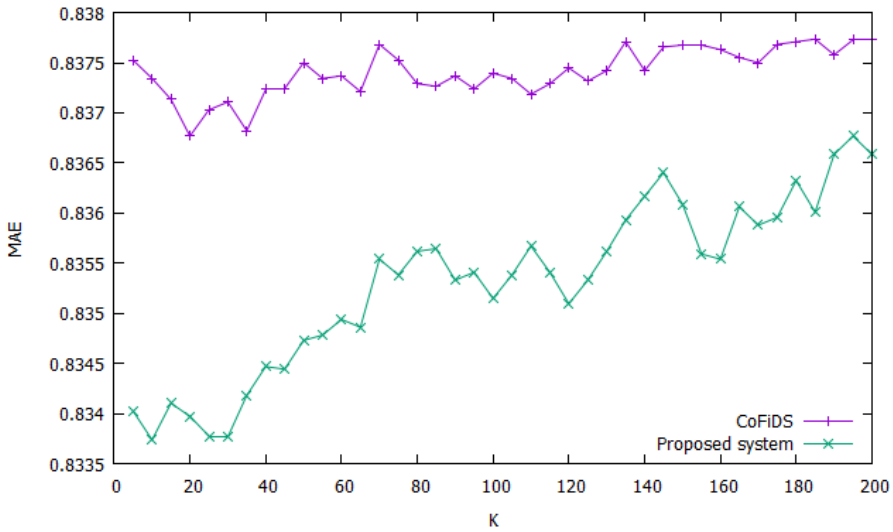



**Figure:** Overall $MAE$ versus $K$

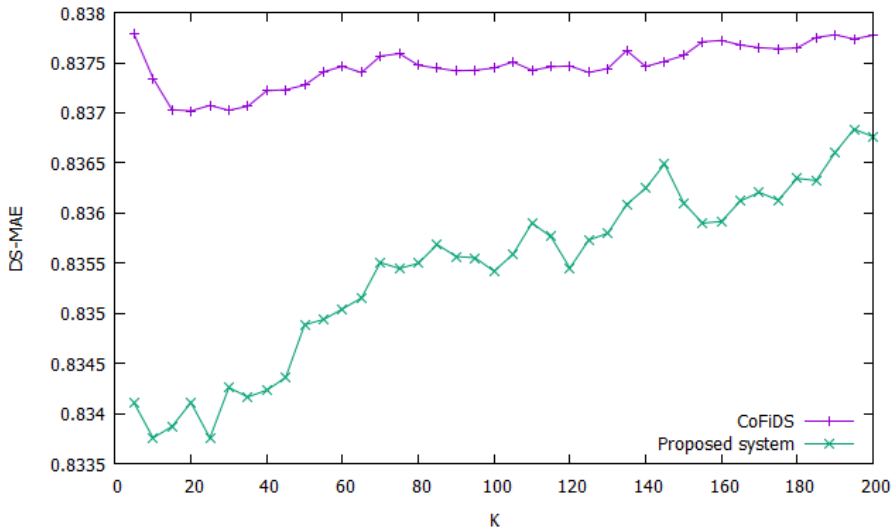
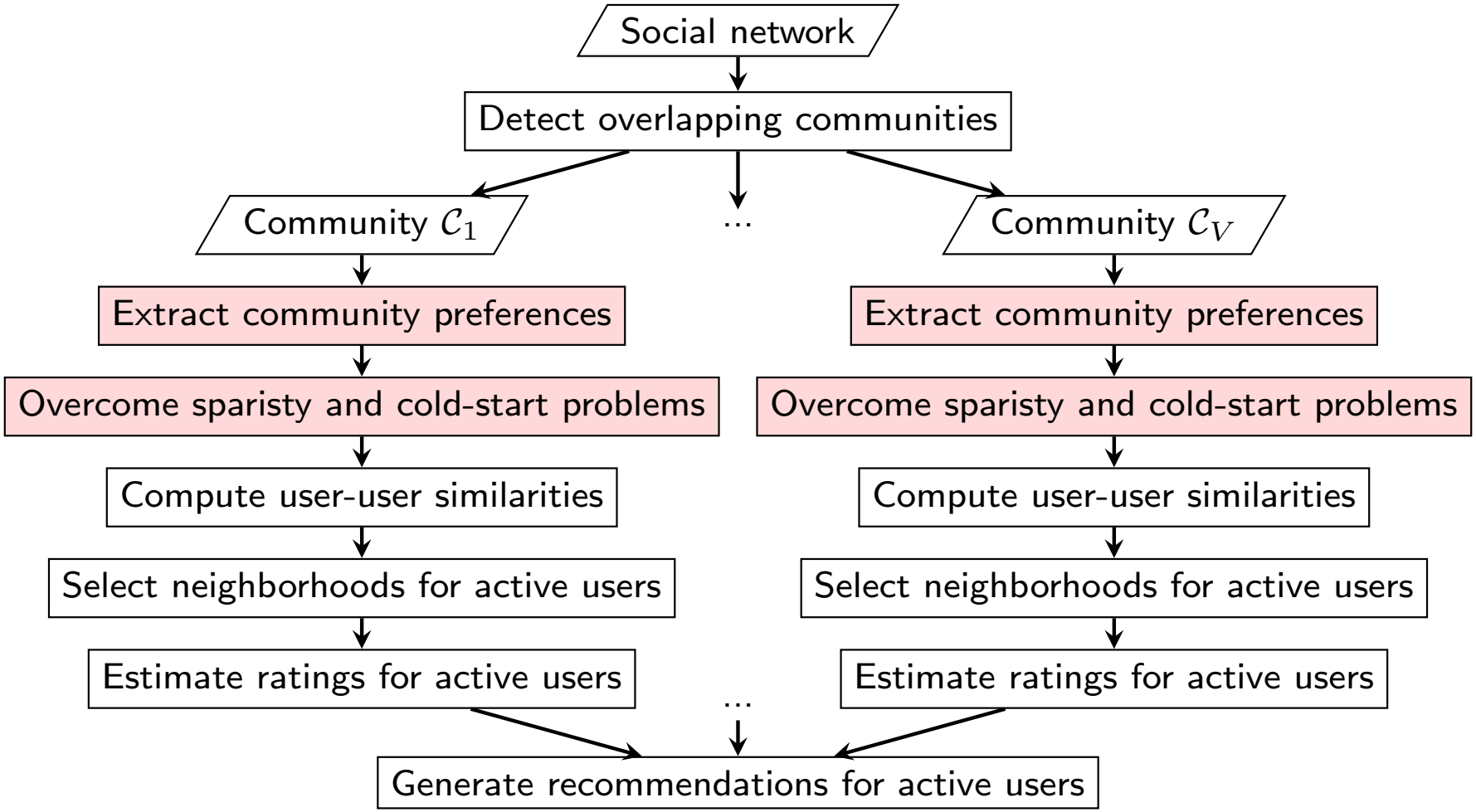



**Figure:** Overall $DS\text{-}MAE$ versus $K$

# Integrating Social Networks into DST-Based RSs

## Exploiting Community Preferences



# Integrating Social Networks into DST-Based RSs

## Exploiting Community Preferences

- Overcoming the sparsity problem
  - Generating unprovided ratings by using the extracted community preferences
- Overcoming the cold-start problem: new items
  - Generating all unprovided ratings on new item $o_{k'}$
  - If item $o_{k'}$ belongs to group $g_{p,q}$ then community preference on group $g_{p,q}$ is considered to be community preference on this item regarding group $g_{p,q}$

# Integrating Social Networks into DST-Based RSs

## Exploiting Community Preferences

- Overcoming the cold-start problem: new users
  - Generating all unprovided ratings regarding new user $u_{i'}$
  - If user $u_{i'}$ is interested in group $g_{p,q}$ then community preference on item $o_k$ regarding group $g_{p,q}$ is considered as preference of user $u_{i'}$ on item $o_k$ regarding group $g_{p,q}$
  - If information about the groups in which user $u_{i'}$ is interested is not available, community preference on item $o_k$ is considered as preference of this user on item $o_k$

✓ Reference: [Nguyen & Huynh, CSoNet 2016]; [Nguyen *et al.*, Electronic Commerce Research and Applications, 2017]

# Information Fusion in DST-Based RSs

- Ratings are represented as mass functions
- Tasks of combining mass functions are executed frequently
- Dempster's rule is mainly used for evidence combination
- With Dempster's rule, combined results usually contain many focal elements (FEs) with very low probabilities and a few FEs with high probabilities
- The FEs with very low probabilities can lead to time consuming and unsatisfactory results in case of combining highly conflicting mass functions

# Our Recent Work

- ML paired with DS theory for user profile modelling

  ✓ Inferring user preferences from short texts generated by users on microblogging platforms such as Facebook and Twitter

  ✓ ML techniques are utilized for concept learning and then DS theory is applied for reasoning and fusion to effectively infer user preferences.

  ✓ Two scenarios of the user profiling problem are considered: static profile (unchanged over time) and dynamic profile (changed over time)

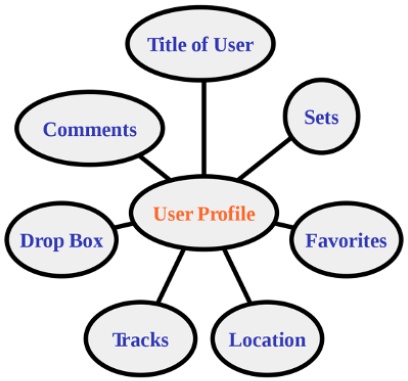
  ✓ The effectiveness and practicality of the developed methods are demonstrated by experiments on short text datasets in comparison with baseline models.

# Outline

# Outline

# User Profiling Problem

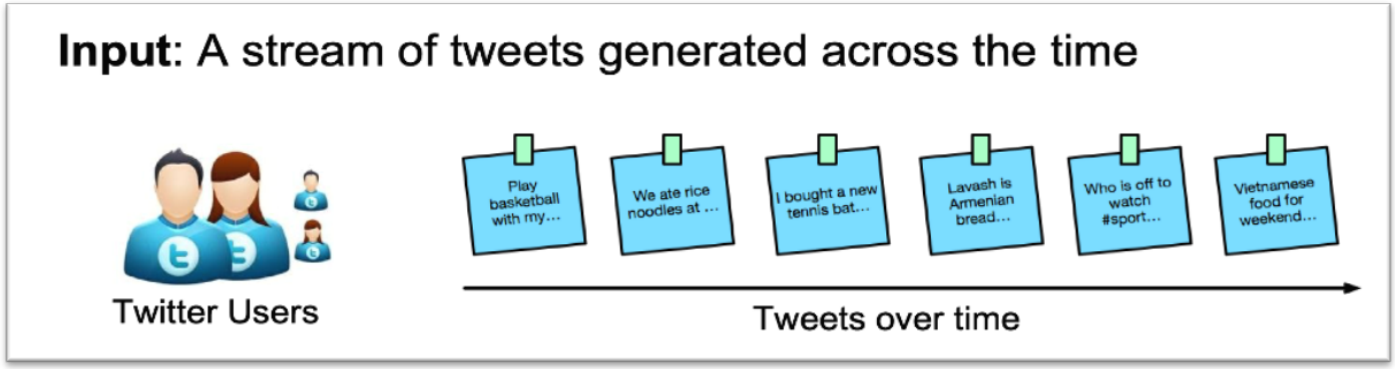
- A (general) "user profile" is a record of personal data associated with a specific user [Wiki].
- The user/expert profile of one person is a record of skills of that person plus a description of her/his network ("social profile") [Balog & de Rijke, 2007].

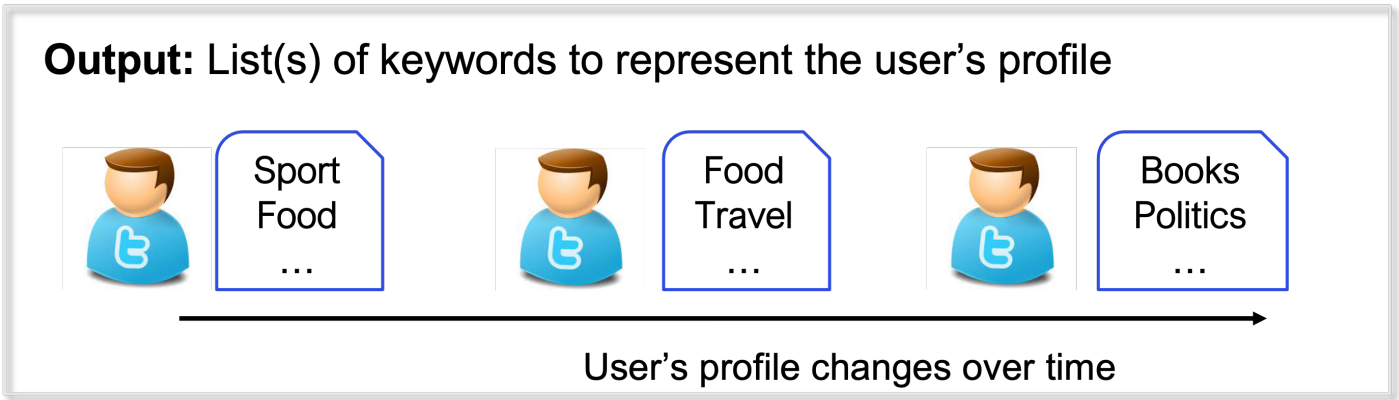



| Attributes | Value | Value Patterns |
|---|---|---|
| Title of User | Name | Letters and Numbers |
| Location | City | Letters (City, State, Country) |
| Tracks | Track File & Date | MP3Audio File; Days |
| Drop Box | Track | MP3Audio File |
| Favorites | Track | MP3Audio File |
| Comments | Words about Songs | Letters; User's Thoughts |
| Sets (Albums) | Track | MP3Audio File |

⇨ The problem of **user profiling** aims at identifying the list(s) of keywords for each user from user's corpus that represents user's expertise or preferences. [Balog *et al.*, 2007; Liang, 2018]

# User Profiling Problem




- Static scenario: [Steyvers *et al.*, KDD 2004; Rosen *et al.*, arXiv.org 2012]
- Dynamic scenario: [Liang *et al.*, KDD 2018; Liang, AAAI 2018]

# User Profiling: Common Challenges

1. The data sparsity problem of short texts

2. User preferences dynamically change over time, and amount of texts within a specific timespan is often limited

3. Data may come in different modes (e.g., images, texts, reactions) from multiple sources (e.g., user may simultaneously have multiple accounts like Facebook, Twitter, Instagram, and WeChat)

4. Social network users create lots of short documents – how to extract and combine useful information from these documents to identify user preferences is still a challenging research problem.

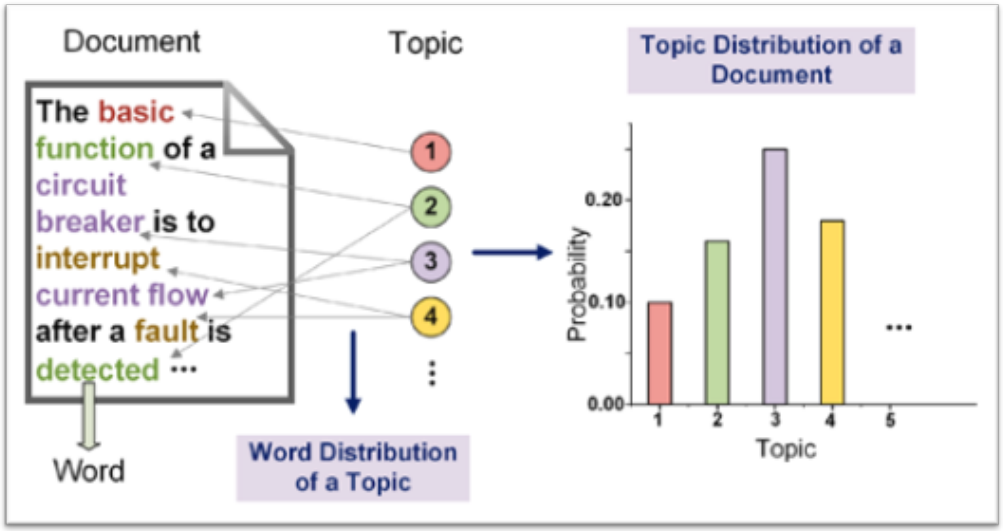
# User Profiling: An Overview

- User profiling has received attention since the launch of the expert finding task at Text REtrieval Conference (TREC) Enterprise Track 2005 [Craswell, de Vries & Soboroff, 2005]
- Most previous work on user profiling worked with collection of static, long documents, and hence posited that users' profile does not change over time [Balog and de Rijke, 2007; Balog *et al.*, 2012]
- Dynamic expertise profiling was introduced in [Rybak *et al.*, 2014] and further studied in [Fang & Godavarthy 2014]. However, these work still worked with a set of long documents.

# User Profiling: An Overview

- Recently, with the rapid growth of social media use, the problem of user profiling in the context of streams of short texts has been actively studied [Deitrick *et al.*, 2012; Estival *et al.*, 2007; Green & Sheppard, 2013; Li *et al.*, 2014; Liang *et al.*, 2018; Liang, 2018]
- User profiles identified from short texts collected from social media are primarily focused for specific applications such as
  - ✓ detecting basic demographic information [Bergsma & Durme, 2013; Preotiuc-Pietro *et al.*, 2015]
  - ✓ inducing the geographical location [Rahimi *et al.*, 2015; Han *et al.*, 2013]
  - ✓ inferring user preferences in politics and their intentions on voting [Cohen & Ruths, 2013; Volkova *et al.*, 2014; Lampos *et al.*, 2013]

# Topic Modelling-based User Profiling




## Observations

- Document corpus = Collection of texts made by users
- Topics that user references = Latent variables

**Problem:** Given a collection of texts created by user.

**Objective:**

- Infer the hidden topics that user is interested in, and
- Extract the top keywords within each topic

# Some Remarks

- Most previous studies were based on frequency for estimating the weight of terms in user vocabularies. However, this approach is not efficient due to the sparsity problem of short texts.
- Frequency-based approach also faces difficulty in capturing "new" topics that first appear in user corpus at a specific time.
- Amount of input data within a specific time interval is limited and it causes difficulties for the inference process.
- Previous approaches are not flexible enough to deal with user data that come from multiple sources (e.g., Twitter, Facebook, LinkedIn, etc.) and in different formats (e.g., texts, photos, reactions, etc.)

# Outline of the Proposed Approach

- We propose an integrated approach that combines advanced ML techniques with DS theory to tackle the aforementioned issues in user profile learning.

  ✓ ML techniques are utilized for **concept learning** that determines the frames of discernment for the target problem of extracting top-$n$ keywords for user profile.

  ✓ Mass functions are determined via **maximum a posterior estimation** (MAP) for text data and then combined using Dempster's rule for inferring users' keyword distributions.

# Outline

# Problem Formulation

The problem is to identify top-$n$ keywords from users' documents for their profiles. Formally, define a function $f$ such that:

$$\begin{aligned} f : \langle \mathcal{U}, \mathcal{D} \rangle &\longrightarrow \mathcal{W} \\ \langle u_i, D_i \rangle &\longmapsto w_i \end{aligned}$$

- $\mathcal{U} = \{u_1, u_2, ..., u_m\}$ – the set of users
- $\mathcal{D} = \{D_1, D_2, ..., D_m\}$ is the set of corpora, each corpus $D_i$ consists of all short documents created by user $u_i$
- $\mathcal{W} = \{w_1, w_2, ..., w_m\}$ is the set of users' profiles with $w_i$ being the list of top-$n$ keywords extracted from $u_i$'s corpus.

# Outline of the Framework

- An integrated framework based on DS theory of evidence, word embedding, and $k$-means clustering for addressing the user profiling problem in the static context.

- Particularly, it consists of three main phases:

  1. Learning abstract concepts at multiple levels of abstraction from user corpora
  2. Evidence modelling and combination for inferring users' keyword distributions
  3. Extracting user profiles based on users' keyword distributions

# Outline of the Framework

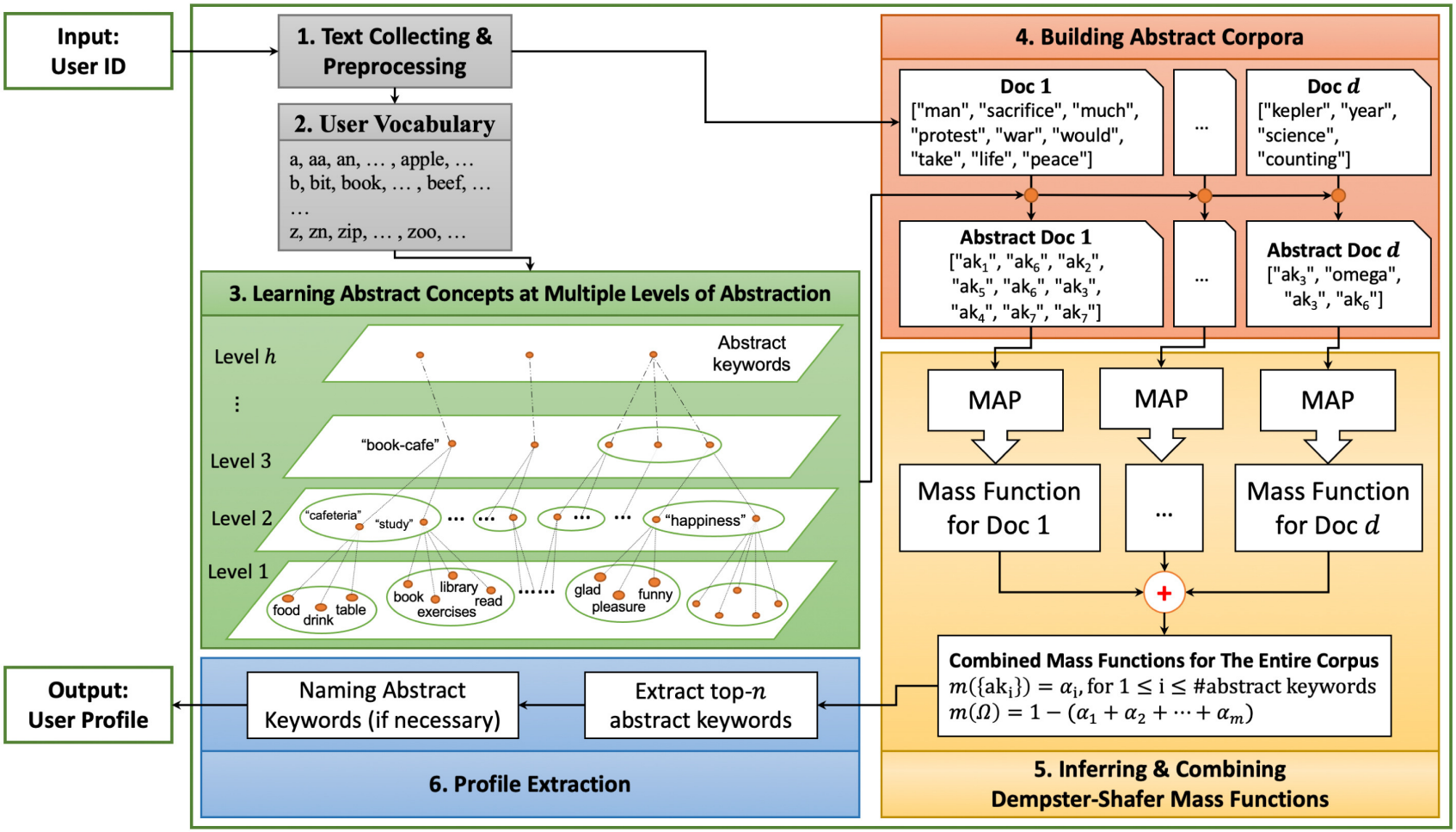
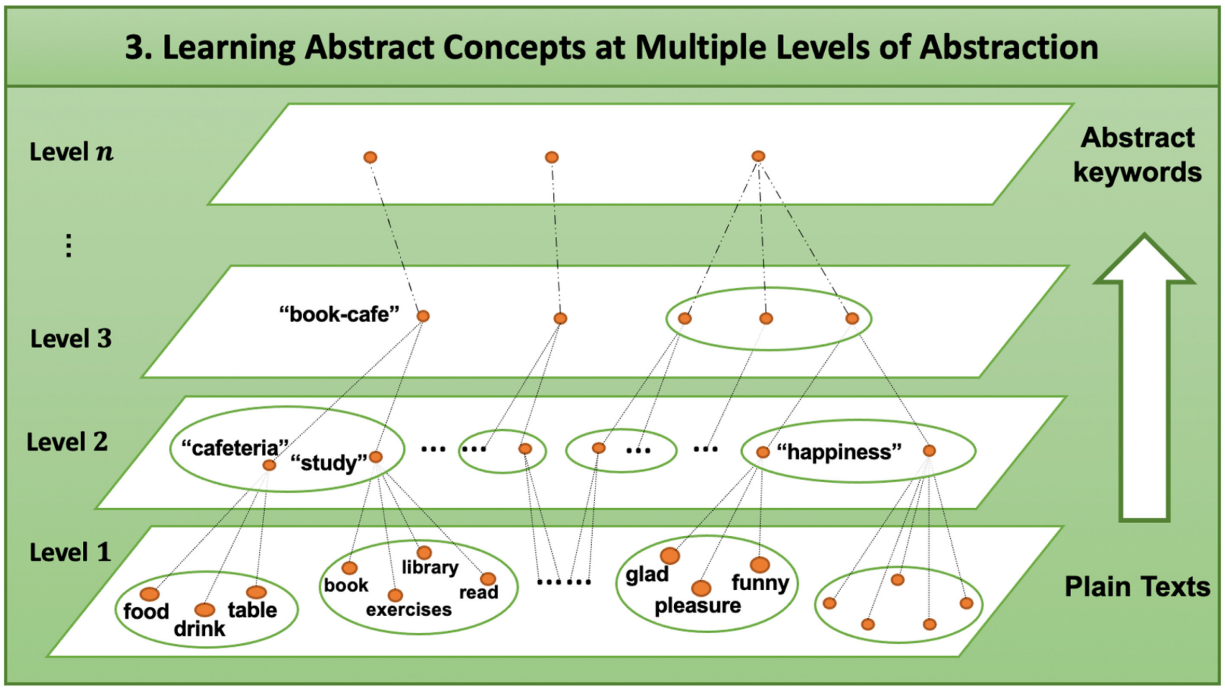



**Figure:** An Evidential Reasoning Based Framework for User Profiling

JAIST
JAPAN ADVANCED INSTITUTE OF SCIENCE AND TECHNOLOGY

# Learning Abstract Concepts

- Text preprocessing: tokenization, normalization, and noise removal
- Converting words to word vectors by a pretrained word embedding model (GloVe [Pennington *et al.*, 2014])
- $k$-means clustering is used for learning a hierarchy of abstract concepts



# Building Abstract Corpora

**Algorithm** : Building Abstract Corpus for A Given User.

**Input:** text corpus $\mathcal{U}$, user dictionary, distant function $d$, threshold $\epsilon$
**Output:** abstract corpus $\mathcal{U}'$ for the given user

```
for each document d ∈ U do
    for each word w_i ∈ d do
        get word vector of w_i
        calculate all similarities sim(w_i, ak_j), ∀j
        if all sim ≥ ε then
            replaced w_i by 'omega' in d
        else
            replaced w_i by ak_j such that sim(w_i, ak_j) is largest
Return abstract corpus U'
```

- Replacing a word $w_i$ in document $d$ by its nearest abstract keyword among abstract keywords learned from user's vocabulary according a distance function.

# Building Abstract Corpora

**Algorithm** : Building Abstract Corpus for A Given User.

**Input:** text corpus $\mathcal{U}$, user dictionary, distant function $d$, threshold $\epsilon$
**Output:** abstract corpus $\mathcal{U}'$ for the given user

```
for each document d ∈ U do
    for each word w_i ∈ d do
        get word vector of w_i
        calculate all similarities sim(w_i, ak_j), ∀j
        if all sim ≥ ε then
            replaced w_i by 'omega' in d
        else
            replaced w_i by ak_j such that sim(w_i, ak_j) is largest
Return abstract corpus U'
```

- 'omega' is a special abstract keyword added for representing *total ignorance*.

# Maximum a Posterior Estimation

- Let $\mathcal{V} = \{ak_1, ak_2, ..., ak_V, ak_{V+1} = \text{'omega'}\}$ – the set of all abstract keywords at a specific level in the hierarchical structure.
- Consider a given set $\mathcal{W} = \{w_1, w_2, ..., w_N\}$ of $N$ independent, identically distributed (i.i.d.) draws from a multinomial distribution on $\mathcal{V}$.
- In this case, $\mathcal{W}$ is considered as a document $d$ created by a user, e.g., a tweet on Twitter or a status on Facebook.

# Maximum a Posterior Estimation

- The likelihood of these drawings in the document is computed by

$$L(\vec{p}|\vec{w}) = p(\mathcal{W}|\vec{p}) = \prod_{i=1}^{N}\prod_{t=1}^{V+1} p_t^{[w_i=ak_t]} = \prod_{t=1}^{V+1} p_t^{n_t} \quad (1)$$

$$\sum_{t=1}^{V+1} n_t = N \text{ and } \sum_{t=1}^{V+1} p_t = 1 \quad (2)$$

✓ $n_t$ is the number of times abstract keyword $ak_t$ was observed as a word in the document $d$ (i.e., $\mathcal{W}$).

✓ Abstract keywords in $\mathcal{V}$ are assumed to follow a multinomial distribution, denoted as $Mult(ak_t \in \mathcal{V}|\vec{p})$, where $\vec{p}$ is the probability that an abstract keyword $ak_t$ is observed as a word $w_i$ in a given document.

# Maximum a Posterior Estimation

- Bayes rule is then applied to infer the posterior distribution as

$$p\left(\vec{p}|\mathcal{W},\vec{\alpha}\right) = \frac{\prod_{n=1}^{N} p(w_n|\vec{p})p(\vec{p}|\vec{\alpha})}{\int_{\mathcal{P}} \prod_{n=1}^{N} p(w_n|\vec{p})p(\vec{p}|\vec{\alpha})\, d\vec{p}} \tag{3}$$

where

$$\vec{p} \sim Dir(\vec{p}|\vec{\alpha}) = \frac{\Gamma\left(\sum_{t}^{V+1} \alpha_t\right)}{\prod_{t=1}^{V+1} \Gamma(\alpha_t)} \prod_{t=1}^{V+1} p_t^{\alpha_i - 1} \tag{4}$$

and $\vec{\alpha}$ is a concentration parameter vector which each element $\alpha_i$ corresponds to $p_i$ in $\vec{p}$.

# Maximum a Posterior Estimation

- Because the denominator of (3) is just a normalization factor, maximum a posterior estimation in (3) leads to an optimization problem defined by (5)

$$\arg\max_{\vec{p}} \quad \prod_{t=1}^{V+1} p_t^{n_t} \times \frac{\Gamma\left(\sum_{t=1}^{V+1} \alpha_t\right)}{\prod_{t=1}^{V+1} \Gamma(\alpha_t)} \prod_{t=1}^{V+1} p_t^{\alpha_i - 1} \tag{5a}$$

$$= \arg\max_{\vec{p}} \quad \prod_{t=1}^{V+1} p_t^{n_t + \alpha_t - 1} \times \frac{\Gamma\left(\sum_{t=1}^{V+1} \alpha_t\right)}{\prod_{t=1}^{V+1} \Gamma(\alpha_t)} \tag{5b}$$

$$\text{subject to} \quad \sum_{t=1}^{V+1} p_t = 1 \tag{5c}$$

# Evidence Modelling for Document

- Solving this constraint optimization problem by Lagrange multiplier method gives us the following solution

$$p_t = \frac{n_t + \alpha_t - 1}{\sum_{t'=1}^{V+1} (n_{t'} + \alpha_{t'} - 1)}, \ \forall t \in [1, V+1] \tag{6}$$

- Now, applying (6) to define the mass function associated with document $d$ in the corpus as follows

$$m_d\big(\{ak_t\}\big) = \frac{(\#\text{times } ak_t \text{ appears in } d) + \alpha_t - 1}{(\#\text{words in } d) + \sum_{t=1}^{V+1} \alpha_t - (V+1)}, \tag{7}$$

$$m_d(\Omega) = \frac{(\#\text{times 'omega' appears in } d) + \alpha_{omega} - 1}{(\#\text{words in } d) + \sum_{t=1}^{V+1} \alpha_t - (V+1)}. \tag{8}$$

where $\{ak_t\} \subseteq \Omega = \big\{ak_1, ak_2, ..., ak_V\big\}, \ \forall t \in \big[1, V\big]$

# Evidence Combination for Keyword Distribution

- For user $u_i$, each document $d$ in the user's corpus $D_i$ is considered as a piece of evidence represented by $m_d$ for inferring the user's profile
- Dempster's rule is then used for combining $m_d$'s for all $d \in D_i$ to obtain the overall mass function $m_i$ for user $u_i$

$$m_i = \bigoplus_{d \in D_i} m_d$$

- Finally, the mass function $m_i$ for the entire user corpus is used to induce the keyword distribution via the pignistic transformation for the user's profile.

# Overall Process of the Proposed Framework

**Algorithm** : The entire process of our framework: An evidential reasoning approach for user profiling using short texts

**Input**: a positive integer $k$, text corpus, distance function $d$
**Output**: top-$k$ abstract/actual keywords

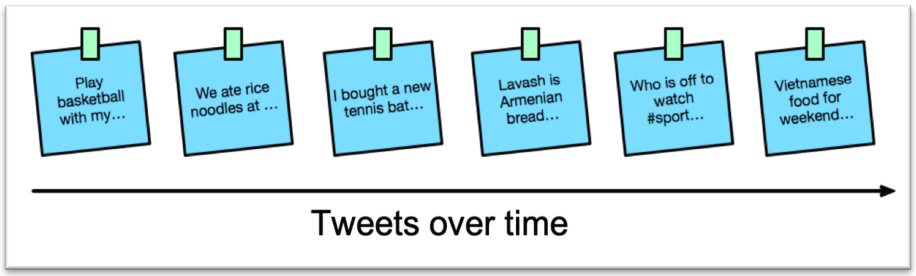
**Phase 1:**
  Get word vector of words via pretrained GloVe model
  Learn and represent abstract concepts at multiple levels of abstractions
**Phase 2:**
  Build abstract corpus: replace individual words by its nearest centroid
  Infer mass function for each document via (18), (19)
  Combine all mass functions via (23), (24), (25), (26)
**Phase 3:**
  Compute pignistic probability for all singletons in the focal set via (9)
  Sort pignistic probabilities in descending order
  Pick up top-$k$ abstract keywords with highest pignistic probability, called set $S$
  Name the abstract keywords if necessary, called $S'$
Return $S/S'$ as user profile
**End Algorithm.**

# Experimental Results

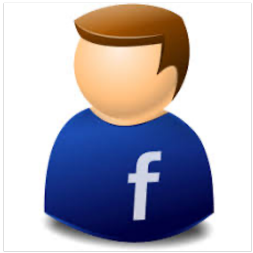
## Datasets: Twitter [Liang, 2018] and Facebook

Twitter Dataset

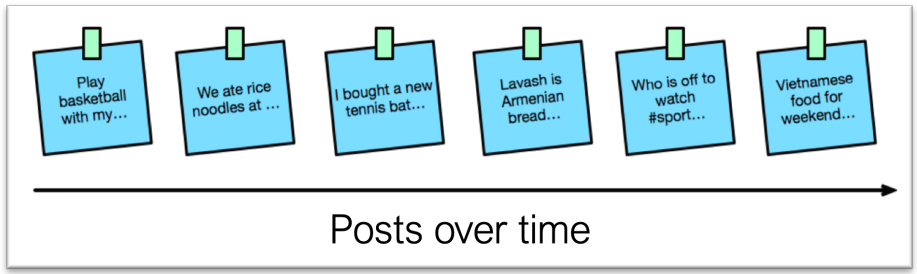

#Users: 1189
#Tweets: 3219
#Tweets' length: 12 words




Facebook Dataset



#Users: 1259
#Posts: 620
#Posts' length: 36 words



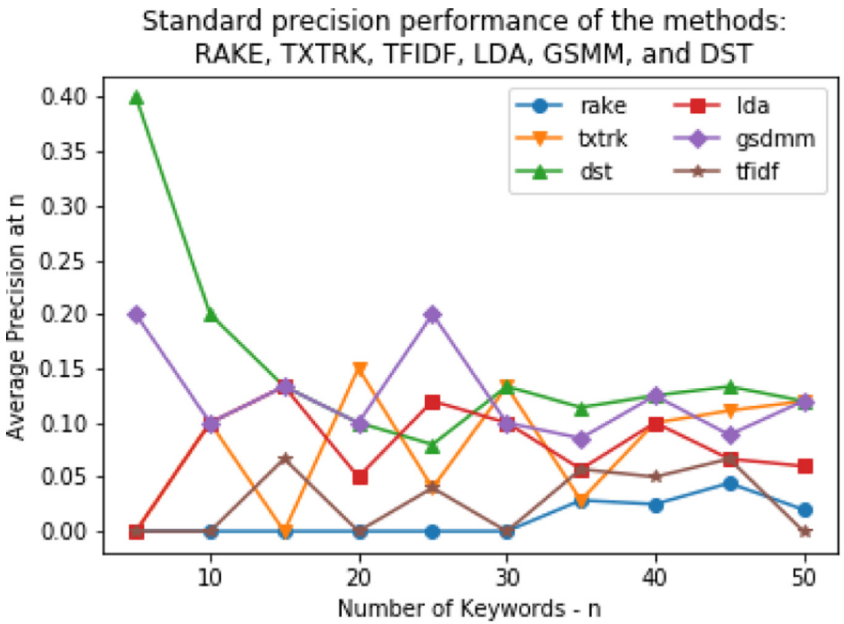
# Experimental Results

## Criteria (CR) & Evaluation Metrics

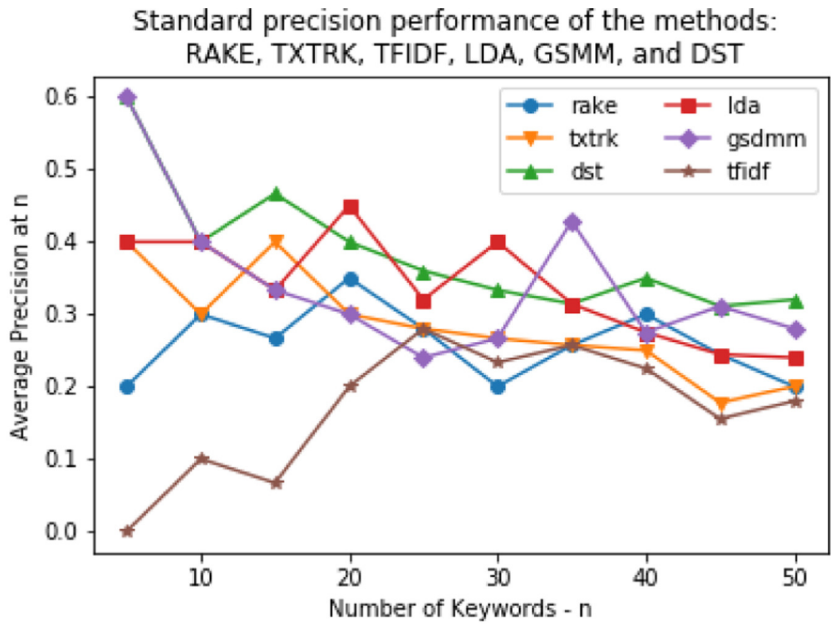
- CR1: How is the performance of the proposed framework and baseline models in inferring actual keywords using short texts? ⇒ **Standard Precision**
- CR2: How is the performance of the proposed framework and baseline models in capturing the conceptual abstractions shared between words that reflect users' preferences in their profiles? ⇒ **Semantic Precision**
- CR3: What is the impact of the size of word vectors on the overall performance of profiles derived by different models? ⇒ **Word Vector Sizes**: 25, 50, 100, and 200
- CR4: How is the time complexity of the proposed method in comparison with baselines when applying to practical datasets? ⇒ **Runtime**

# Experimental Results

## Standard precision



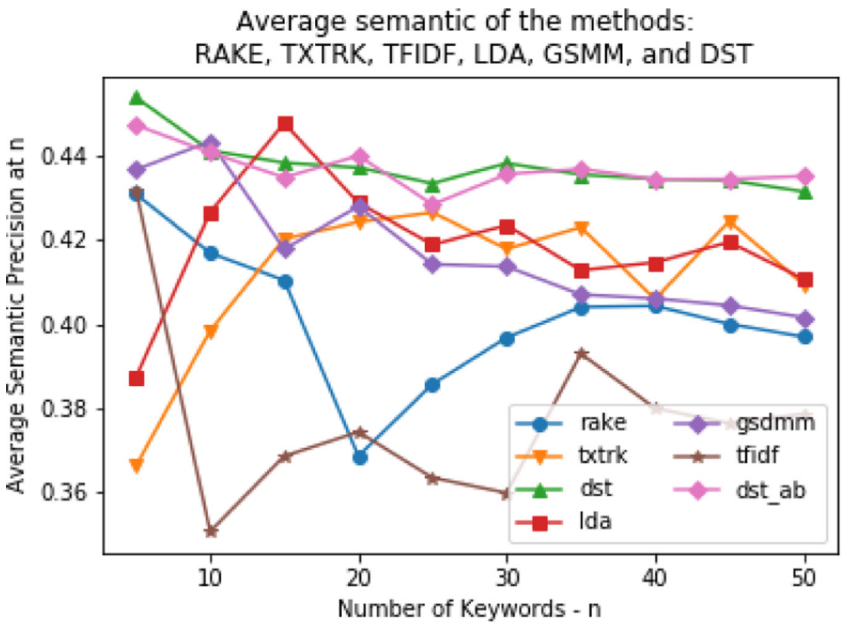

(a) Twitter data



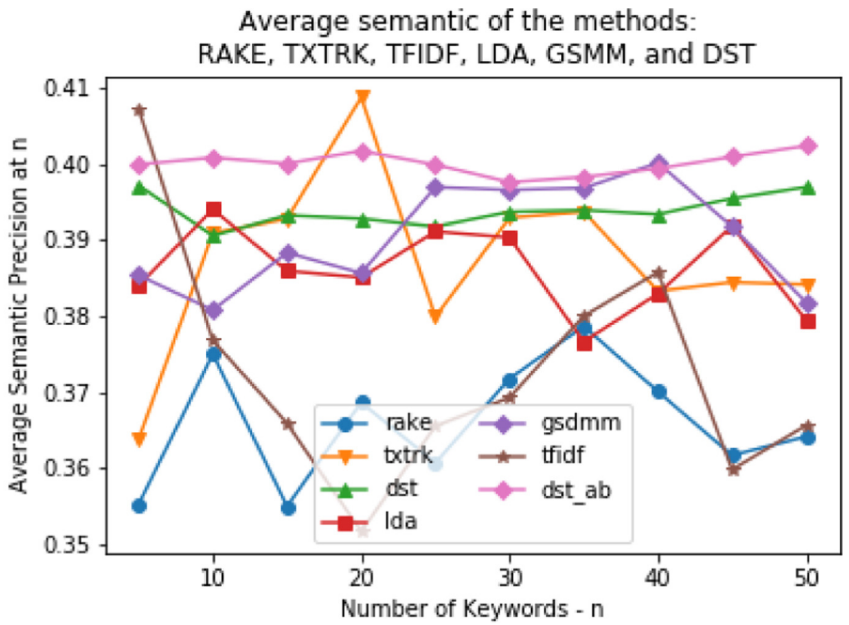

(b) Facebook data

# Experimental Results

## Semantic precision



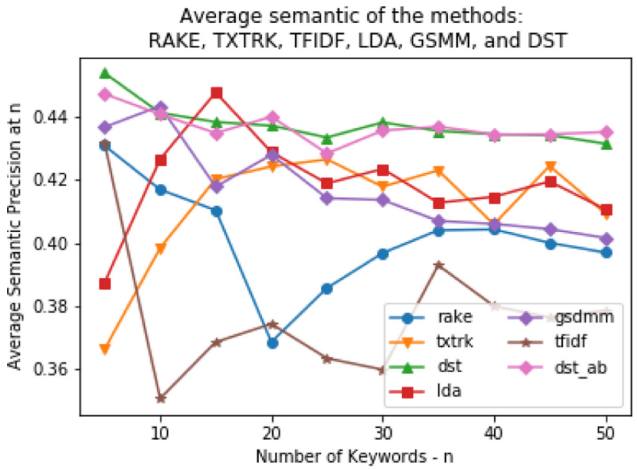

(a) Twitter data



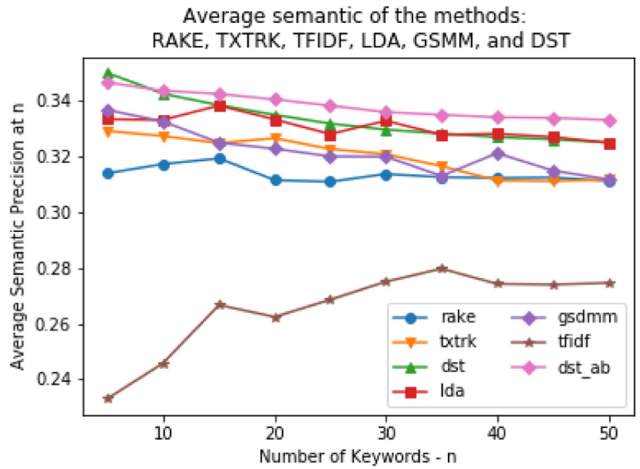

(b) Facebook data

# Experimental Results

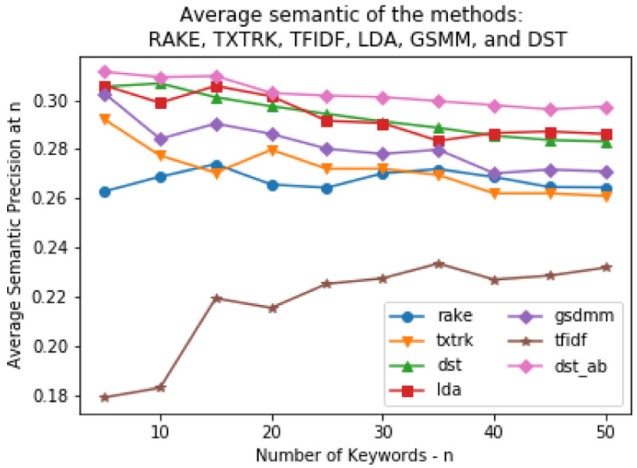
## Semantic precision at different word vector sizes – Twitter dataset



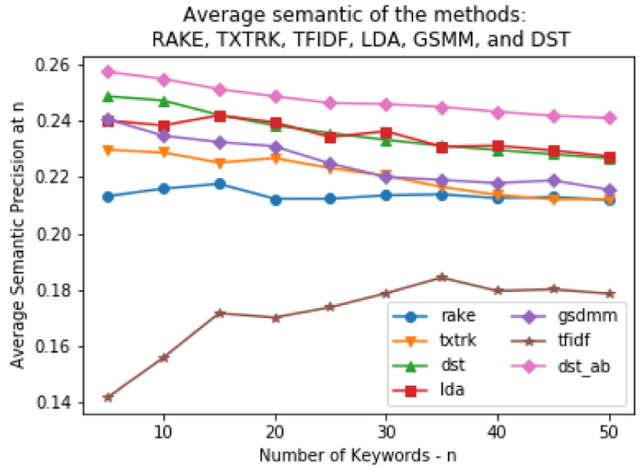

(a) Size of Word Vector = 25




(b) Size of Word Vector = 50



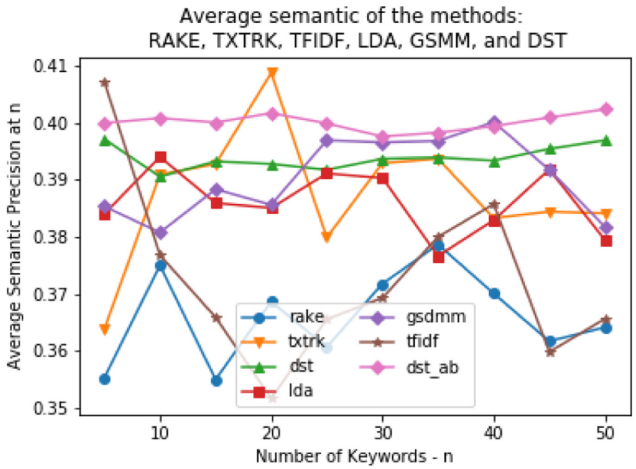

(c) Size of Word Vector = 100



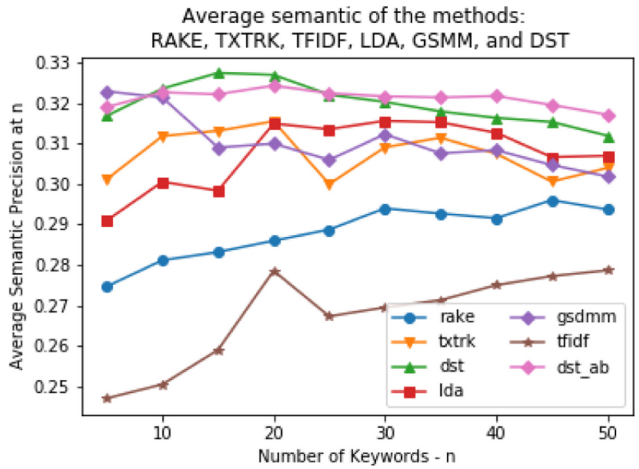

(d) Size of Word Vector = 200

# Experimental Results

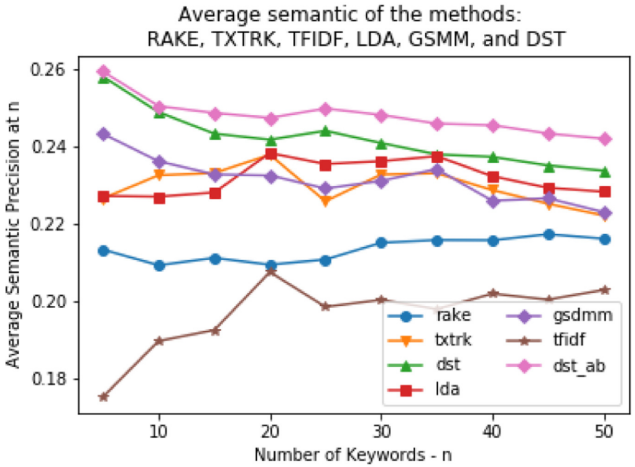
## Semantic precision at different word vector sizes – Facebook dataset



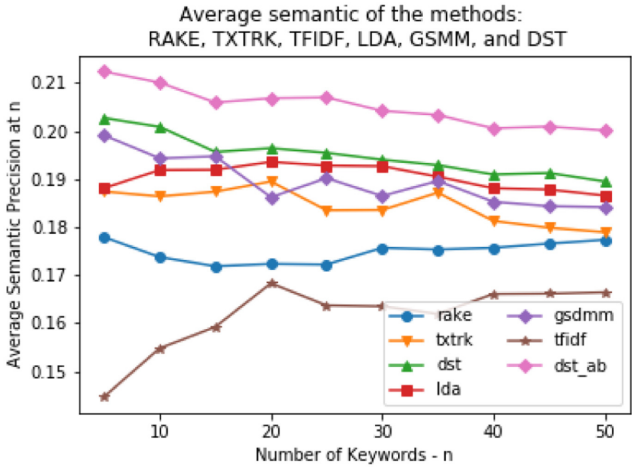

(a) Size of Word Vector = 25




(b) Size of Word Vector = 50



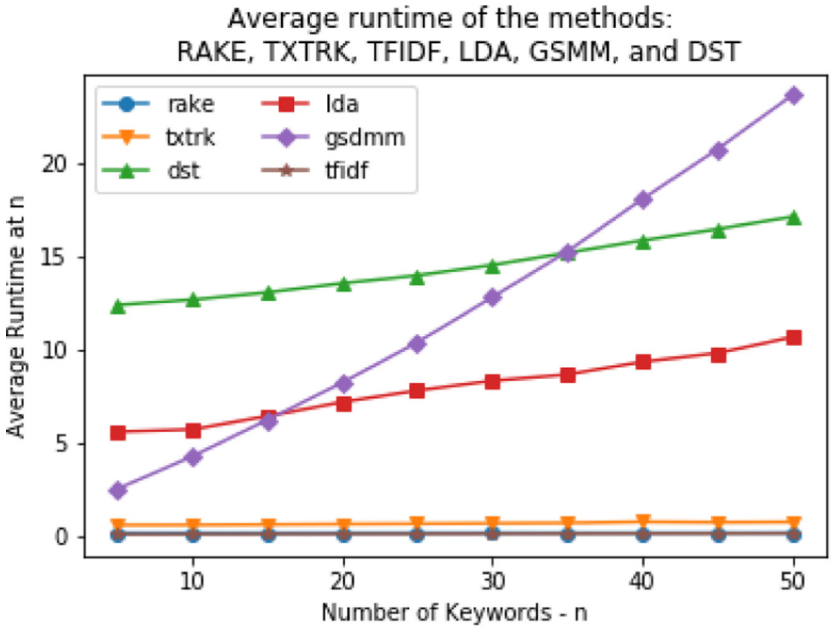

(c) Size of Word Vector = 100



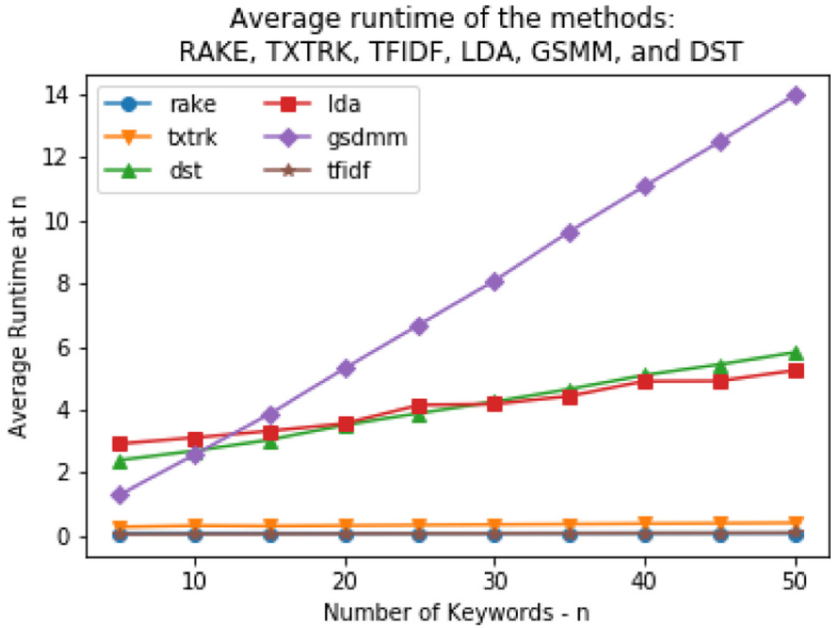

(d) Size of Word Vector = 200

# Experimental Results

## Runtime




(a) Twitter data




(b) Facebook data

# Outline

# Dynamic User Profiling

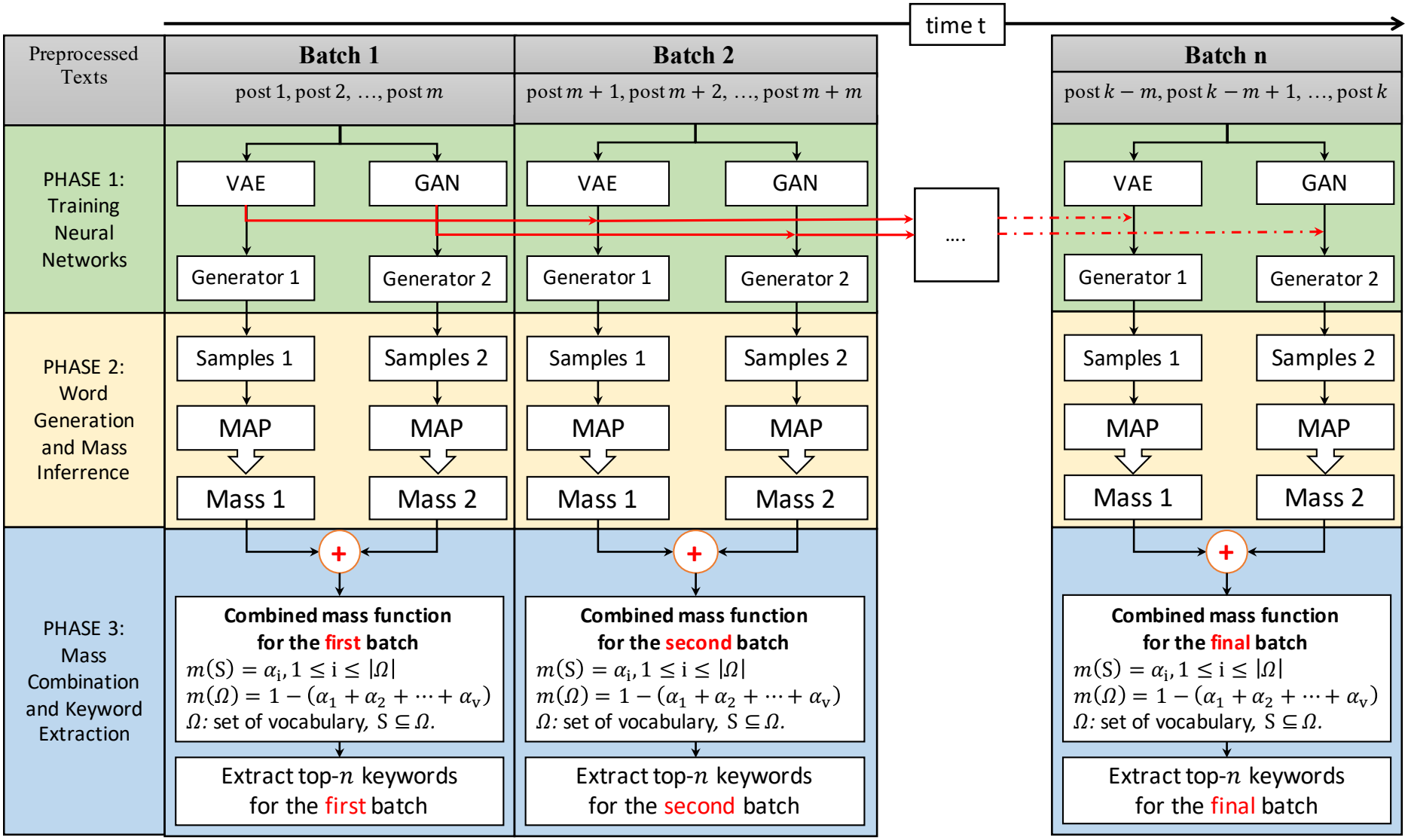
- An integrated framework based on deep neural networks and DS theory for identifying user profiles over time in the context of streams of short texts.

- Particularly, it consists of three main phases:

  1. Learning the latent space of user texts using two deep neural networks (i.e., Variational AutoEncoder (VAE) and Generative Adversarial Network (GAN))
  2. Word generation and evidence modelling from these two networks
  3. Evidence combination and keyword extraction for each specific time span.

# Dynamic User Profiling

## Outline of the proposed framework



# Dynamic User Profiling

## Word generation

**Algorithm** : DST-based Word Generation

**Input:** current VAE/GAN, previous VAE/GAN, lowerBound $\epsilon_1$, upperBound $\epsilon_2$, User Vocabulary, English, and number of draws $N$

**Output:** A bunch of generated word vectors that represent user preferences

Compute generator at the current timestamp $G^{(t)}$ via (12)
Let set $S = \emptyset$
Let the count for $\Omega$ set $count_\Omega = 0$
**while** $|S| \leq N$ **do**
    Generate one word vector $v$ by $G^{(t)}$
    Extract top-$n$ keywords in user vocabulary that are most similar to $v$, denoted as $key_{largest}, ..., key_{smallest}$
    **if** $sim(key_{largest}, v) \leq \epsilon_1$ **then**
        Extract top-$k$ keywords in English that are most similar to $v$
        Add top-$k$ keywords into $S$
        Extend User Vocabulary with these new tokens (a set of new vocabularies is added into $S$)
    **else if** $sim(key_{smallest}, v) \geq \epsilon_2$ **then**
        $count_\Omega$ += 1
        (Nothing is added into $S$ because of drawing a common word)
    **else**
        Add top-$n$ keywords to $S$
Return $S$, $count_\Omega$

Two concepts in DS theory are incorporated into the process of word generation:

- Open-world assumption allows generating new words first appearing in user vocabulary ⇒ this is essentially important for capturing the dynamic change of user preferences over time.
- Total Ignorance allows to skip common words (not stop words) when generating new tokens.

# Dynamic User Profiling

## Evidence modelling and combination

- The generators in the trained VAE and GAN work independently as two experts to generate bunches of tokens for modeling user preferences.
- In a specific time span, two bunches of words generated from VAE and GAN generators are used as two independent sources of evidence supporting determination of user preferences.
- Mass functions representing these sources of evidence are determined via maximum a posterior estimation (MAP) and then combined via Dempster's rule for inferring users' keyword distributions in the time span.

JAIST JAPAN ADVANCED INSTITUTE OF SCIENCE AND TECHNOLOGY

# Dynamic User Profiling

## Overall Process of the Proposed Framework

**Algorithm 1:** The entire process for finding user preferences within a specific time interval

**Input:** a positive integer $k$, $batch_t$ of user texts, previous generators $G^{(t-1)}$

**Output:** top-$k$ keywords

**Phase 1:**

Learn the latent space of current $batch_t$ via VAE

Learn the latent space of current $batch_t$ via GAN

Calculate the generator $G_{vae}^{(t)}$ via (12)

Calculate the generator $G_{gan}^{(t)}$ via (12)

**Phase 2:**

Generate the first bunch of words via $G_{vae}^{(t)}$ and Alg. 1

Generate the second bunch of words via $G_{gan}^{(t)}$ and Alg. 1

Infer the mass function corresponds to $G_{vae}^{(t)}$ via (19), (20)

Infer the mass function corresponds to $G_{gan}^{(t)}$ via (19), (20)

**Phase 3:**

Combine two mass functions via (21), (22)

Compute pignistic probability of singletons via (8)

Sort pignistic probabilities in descending order

Pick up top-$k$ keywords called set $S$

Return $S$ as user preferences at time $t$

**End Algorithm.**

# Experimental Results

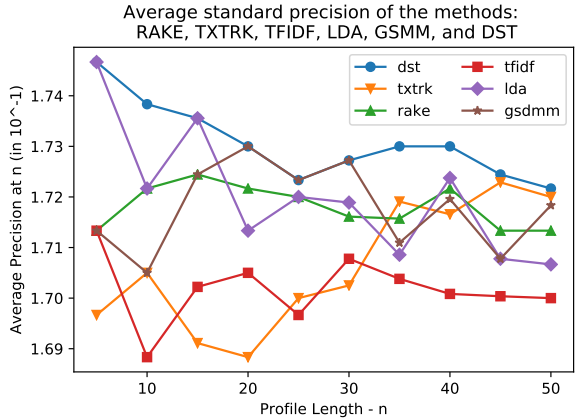
**Datasets: Twitter [Liang, 2018] and Facebook**

Each user corpus is organized as below:

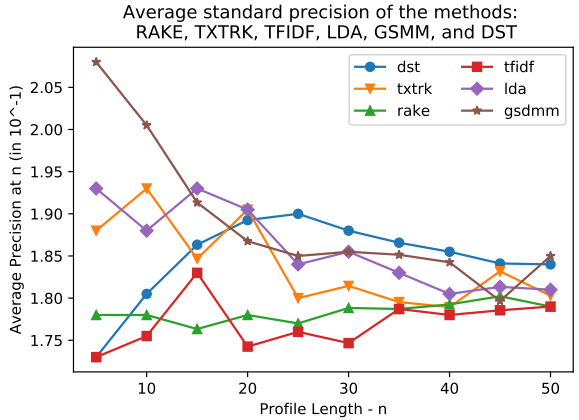
- Grouped into 12 batches corresponds to 12 intervals according to timestamps.
- For each batch, 95% are used for training and 5% are used for testing
- The training sets are used to train VAE and GAN separately
- The test sets are used to evaluate the efficiency of the proposed model
- Metrics: Precision, Semantic precision, Mean Reciprocal Rank (MRR), and Mean Average Precision (MAP), Runtime [Croft *et al.*, 2010; Liang, 2018].

# Experimental Results

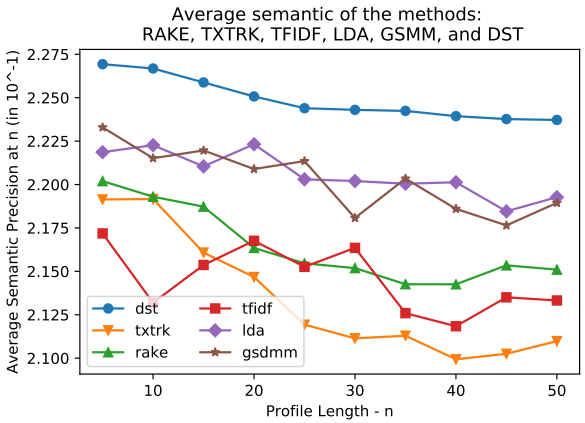
## Precision and Semantic precision



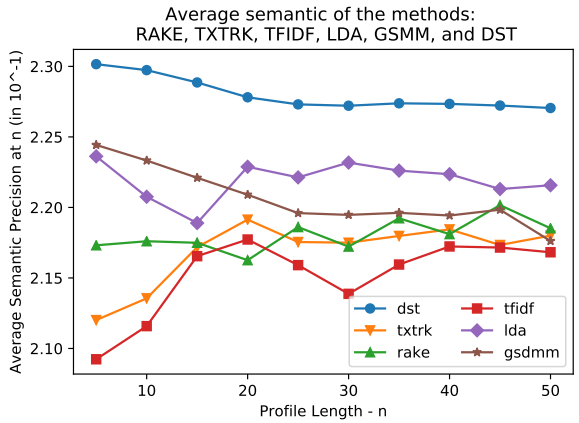

Twitter dataset: Precision




Facebook dataset: Precision



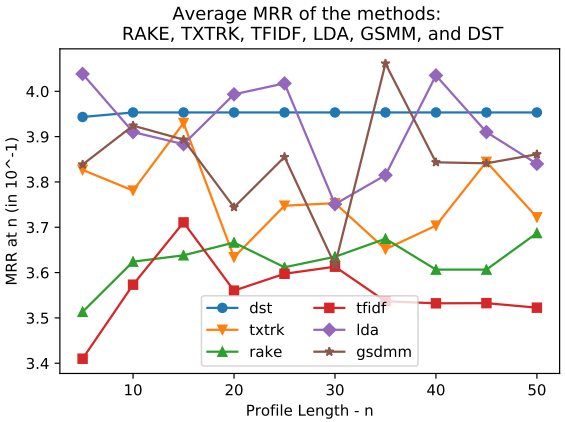

Twitter dataset: Semantic precision



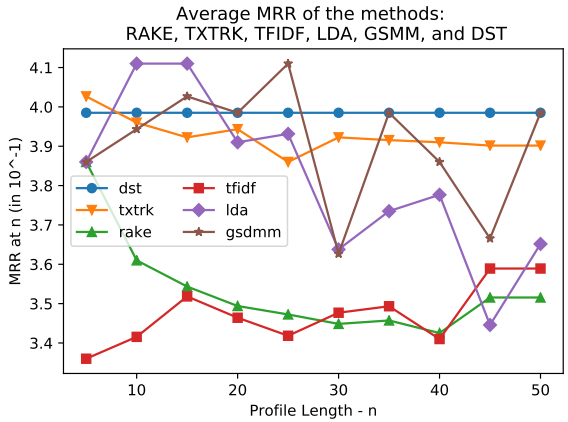

Facebook dataset: Semantic precision

# Experimental Results

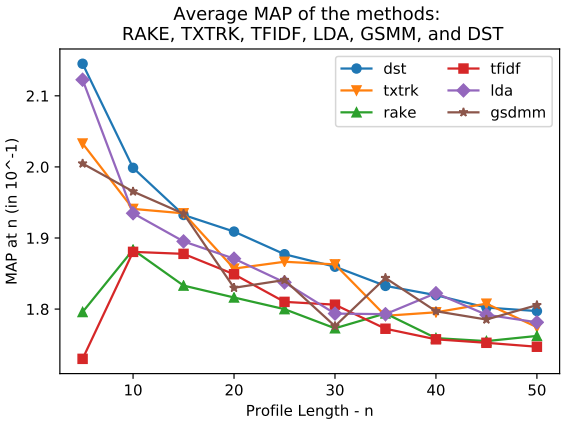
## MRR and MAP



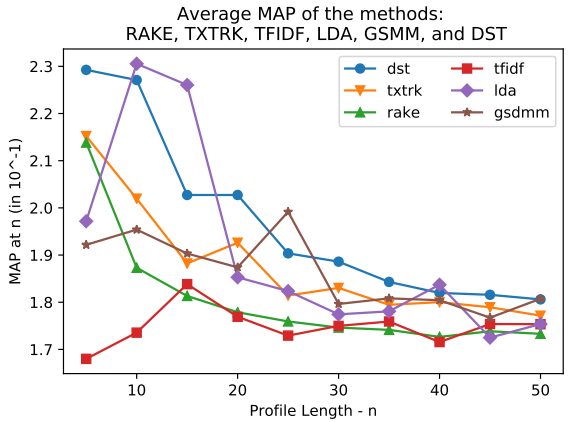

Twitter dataset: MRR




Facebook dataset: MRR



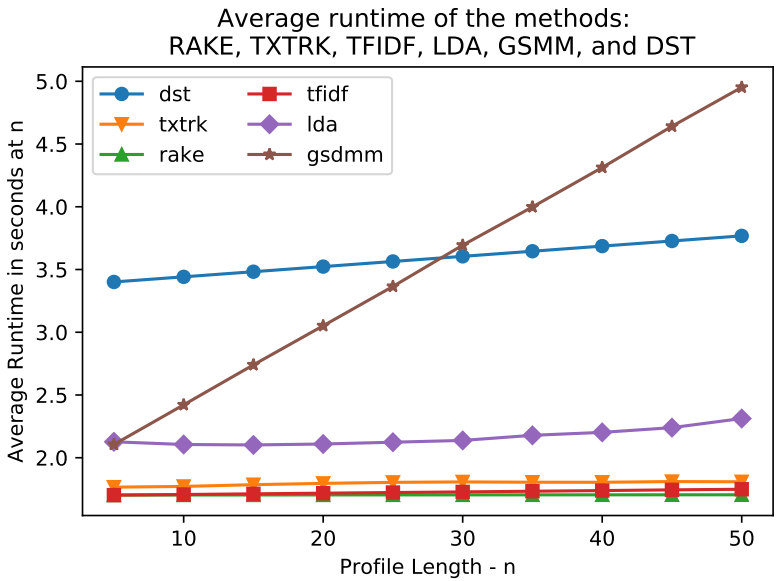

Twitter dataset: MAP



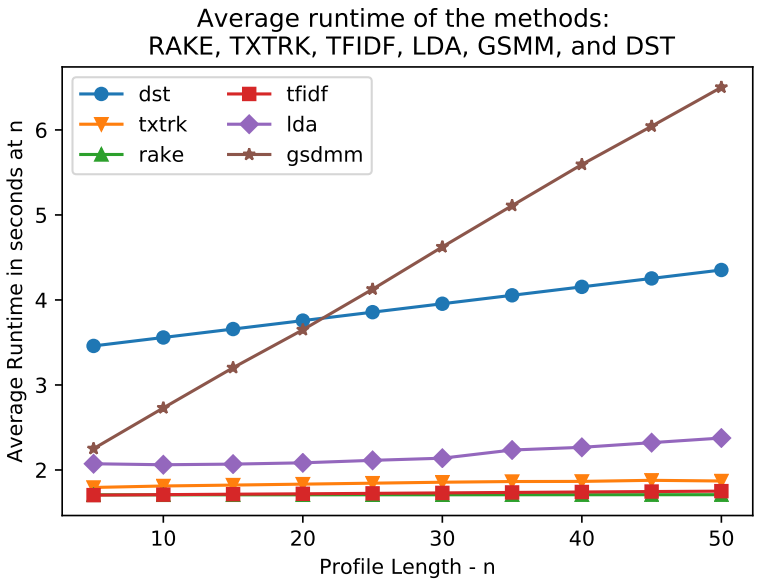

Facebook dataset: MAP

# Experimental Results

## Runtime




Twitter dataset: Runtime




Facebook dataset: Runtime

# References


1. D.-V. Vo, J. Karnjana, V.-N. Huynh. An Integrated Framework of Learning and Evidential Reasoning for User Profiling using Short Texts. *Information Fusion* **70** (2021) 27–42.
2. D.-V. Vo, T.-T. Tran, K. Shirai, V.-N. Huynh. Deep Generative Networks Coupled with Evidential Reasoning for Dynamic User Preferences Inferring Using Short Texts. *IEEE Transactions on Knowledge and Data Engineering* **35**(7)(2022) 6811–6826.

# Outline

# Summary

- Applications of DST in ensemble classification and recommendation systems were briefly summarized.
- An integrated approach that incorporates advanced ML techniques and DS theory to address the problem of identifying user profiles has been also discussed.
- With a general mechanism for reasoning with uncertainty and information combination offered by DS theory, the proposed approach is flexible enough to be adapted to address the problem of user profiling in more general contexts (e.g., data from multiple sources and in different modes, user profiles with 2-gram or 3-gram keywords).