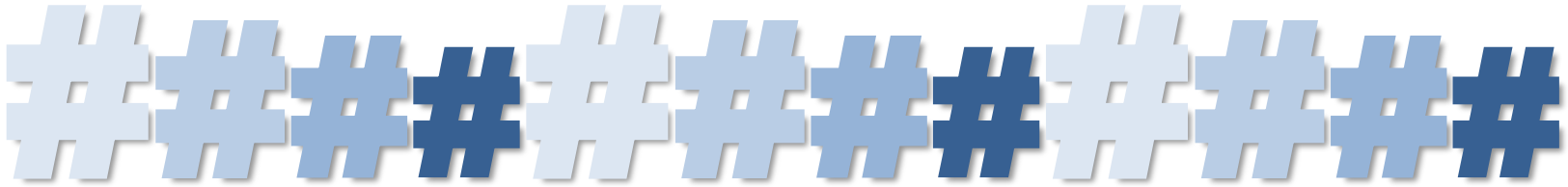
Recursive Hashing

Approximating Similarities of Structured Data



Bin Li

Outline

- Recursive Hashing for Structure Data
- Algorithm 1: Nested Subtree Hashing (NSH)
- Algorithm 2: Recursive Minwise Hashing (RMH)

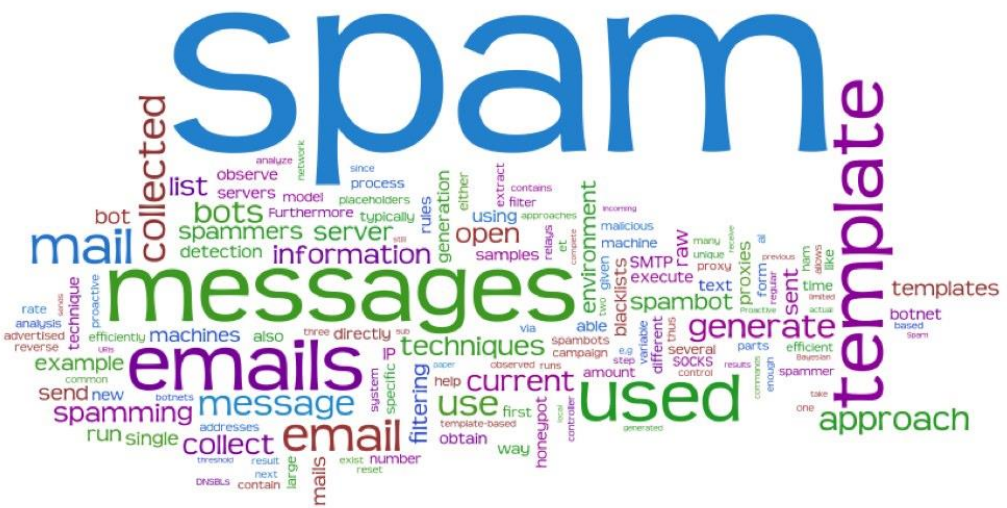
Big Data

- Big data subverts the traditional learning paradigm
  - Huge volumes
  - High-speed streams
  - Infinite features

- We can hardly do ML as usual in big data scenarios
  - Cannot do with acceptable time/space
  - Cannot do in batch mode
  - Cannot do in a predefined feature space

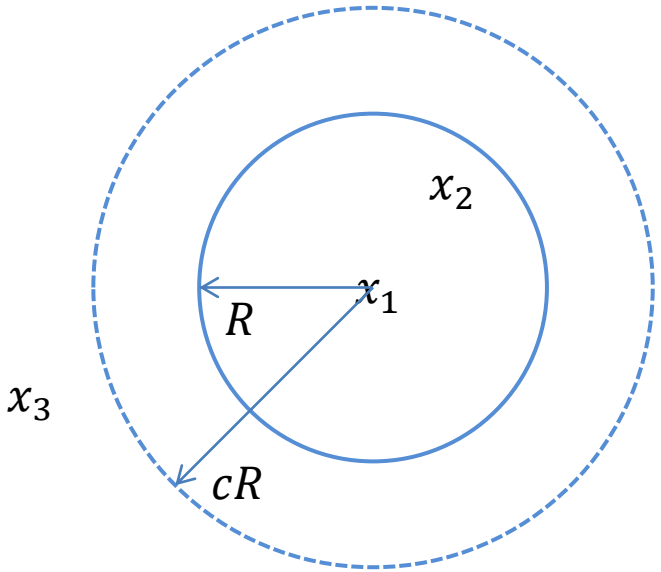
Example

- Spam Detection
  - Millions of emails per second
  - Emerging spam words (e.g., $$$, e.x.t.r.a.) – Infinite features



Possible Solution

- Locality-Sensitive Hashing (LSH)
  - Most address huge volume problems ☺☺
  - Some address high speed stream problems ☺
  - Some address infinite dimensionality problems ☺



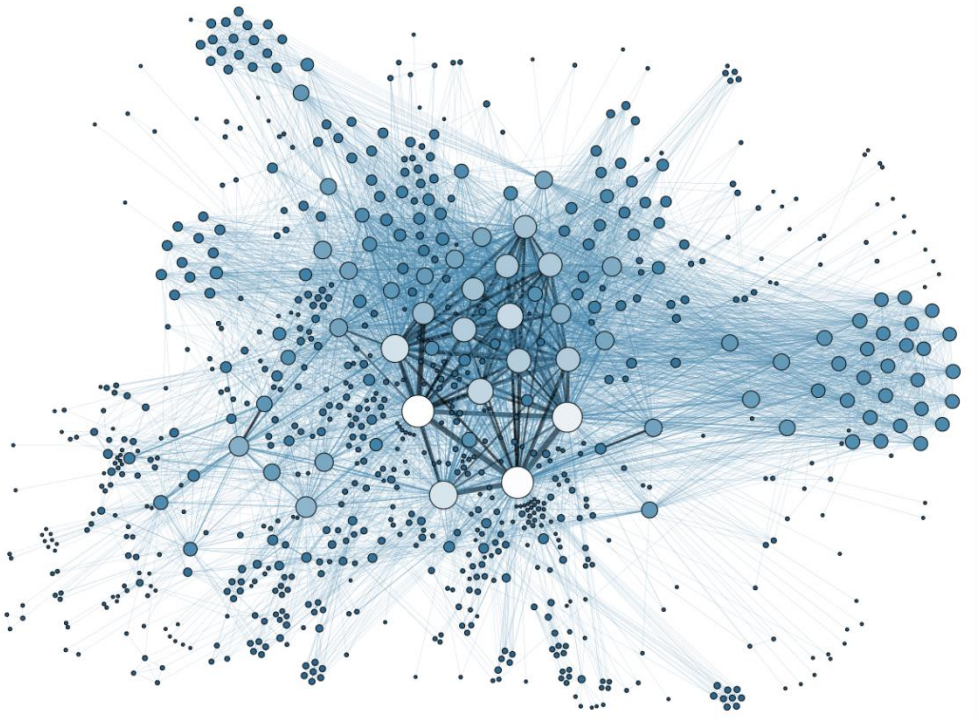
$d(x_1, x_2) \leq R$, then $h(x_1) = h(x_2)$ with high probability
$d(x_2, x_3) \geq cR$, then $h(x_2) = h(x_3)$ with low probability

Related Work

- Spectral Hashing (learning to hash)
  - [Weiss *et al*, 2009] Spectral Hashing
  - [Gong & Lazebnik, 2011] Iterative Quantization: A Procrustean Approach to Learning Binary Codes
- Feature Hashing (random hash)
  - [Shi *et al*, 2009] Hash Kernels for Structured Data
  - [Weinberger *et al*, 2009] Feature Hashing for Large Scale Multitask Learning
- Min-wise Hashing (random hash)
  - [Li & König, 2011] Theory and Applications of b-Bit Minwise Hashing
  - [Li *et al*, 2011] Hashing Algorithms for Large-scale Learning

But…

- No reported work for hashing structured data!
  - Trees
  - Graphs
  - Arrays



Motivation

- Traditional graph similarity
  - (Step 1) Enumerate substructures as a feature space
  - (Step 2) Represent each graph as a feature vector
  - Drawbacks: two scans, fixed feature space, high complexities

- Traditional text similarity
  - (Step 1) Define a dictionary (feature space)
  - (Step 2) Represent each text as a bag-of-words
  - Drawbacks: fixed feature space, loss of hierarchical semantics

This Work

- Contributions
  - Directly hash hierarchical structures – Information lossless
  - Tolerate emerging substructures – Streaming features
  - Approximate the original similarities – Unbiased estimator

- Publications on recursive hashing
  - [ICDM'12] Nested Subtree Hash Kernels for Large-Scale Graph Classification over Streams
  - [SDM'14] Context-Preserving Hashing for Fast Text Classification
  - I'd like to thank a UTS Early Career Researcher Grant (2012-2013)

Outline

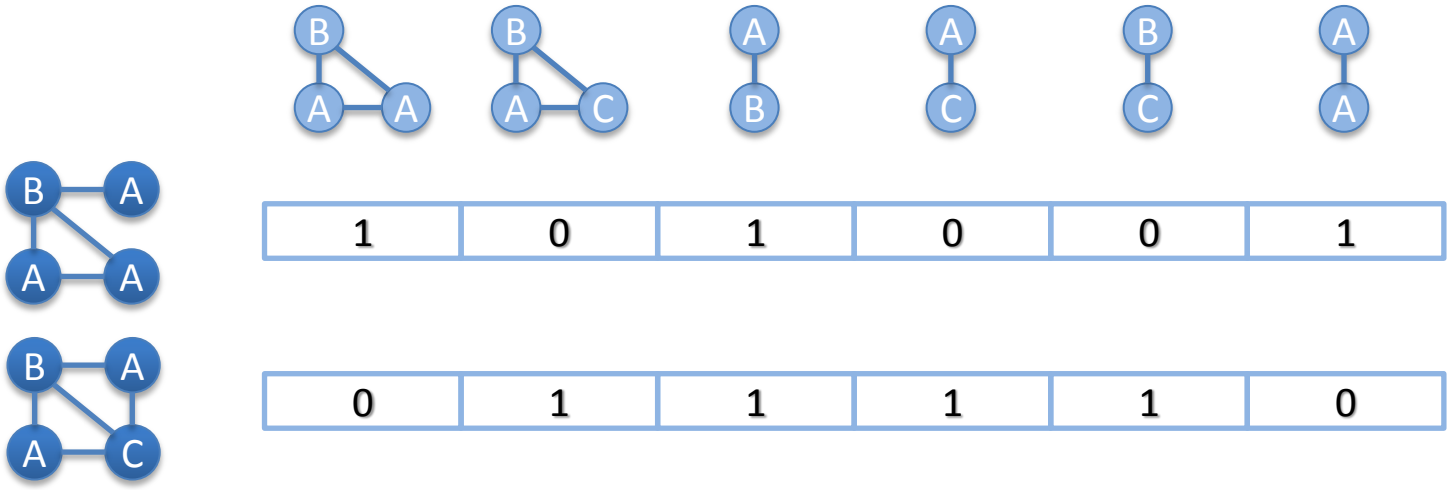
- Recursive Hashing for Structure Data
- Algorithm 1: Nested Subtree Hashing (NSH)
- Algorithm 2: Recursive Minwise Hashing (RMH)

Graphs

- Graph is a natural representation of many structured data
  - Chemical compounds (nodes→atoms, edges→bonds)
  - Social networks (nodes→users, edges→interactions)
  - Program flowcharts (nodes→activities, edges→flows)

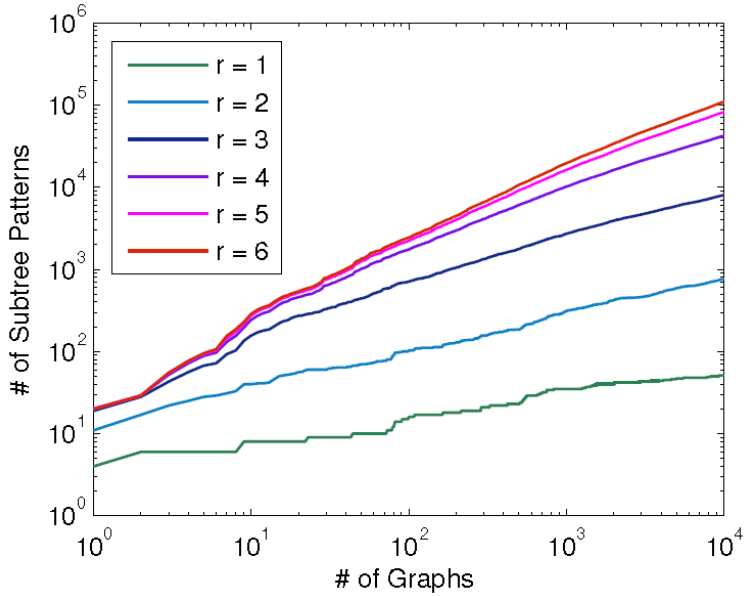
Graph Features

- No intrinsic feature space for graphs
  - Because of different sizes, different node/edge labels
- Enumerate a set of substructures as a common feature space
  - Subgraphs, subtrees, paths, etc.



Graph Kernels

- A graph kernel in the common feature space spanned by
  - Subgraphs (mostly based on frequent subgraph mining)
  - Walks (e.g., Marginalized kernels [Kashima et al, 2003])
  - Paths (e.g., Shortest-path kernels [Borgwardt et al, 2005])
  - Subtrees (e.g., Subtree kernels [Shervashidze et al, 2009]) ← Fatest! But new subtrees fast emerge!



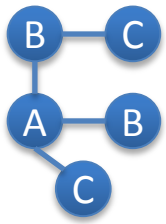
Our Goal

- Limitations of graph kernels
  - A pre-scan over all graphs for constructing a common feature space
  - The feature space keeps expanding if new graphs are observed

- Aims
  - No prescan
  - Fixed dimensionality of feature space
  - Linear time complexity
  - Tolerate concept-drift (substructure distribution changes in a stream)

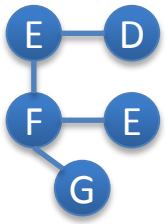
(Preliminary) Subtree Kernel

[Shervashidze & Borgwardt, 2009] Fast Subtree Kernels on Graphs



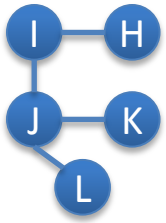
Subtree height = 1 $\mathbf{x}^{(1)} = [1,2,2]$

A (Ø)	B (Ø)	C (Ø)
1	2	2



Subtree height = 2 $\mathbf{x}^{(2)} = [1,2,1,1]$

D (C-B)	E (B-AC)	F (A-BBC)	G (C-A)
1	2	1	1

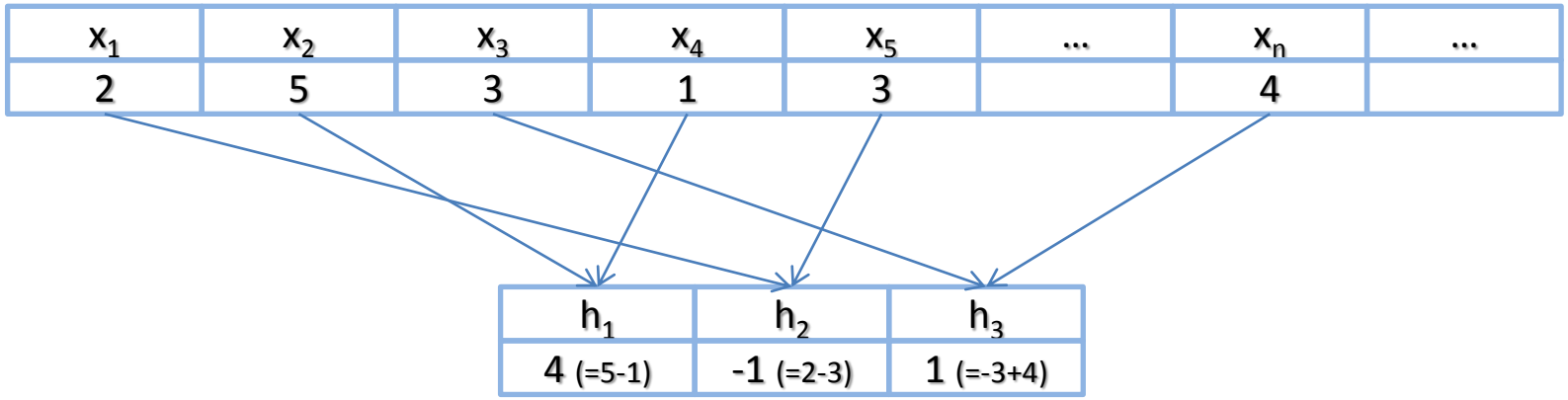


Subtree height = 3 $\mathbf{x}^{(3)} = [1,1,1,1,1]$

H (D-E)	I (E-DF)	J (F-EEG)	K (E-F)	L (G-F)
1	1	1	1	1

(Preliminary) Feature Hashing

[Weinberger *et al*, 2009] Feature Hashing for Large Scale Multitask Learning



First random hash function $h: \mathbb{N} \to \{1, \dots, M\}$ allocates bins

Second random hash function $\delta: \mathbb{N} \to \{\pm 1\}$ determines signs

Unbiased estimator: $\langle \mathbf{x}, \mathbf{x}' \rangle = \mathbb{E}[\langle \mathbf{h}, \mathbf{h}' \rangle]$

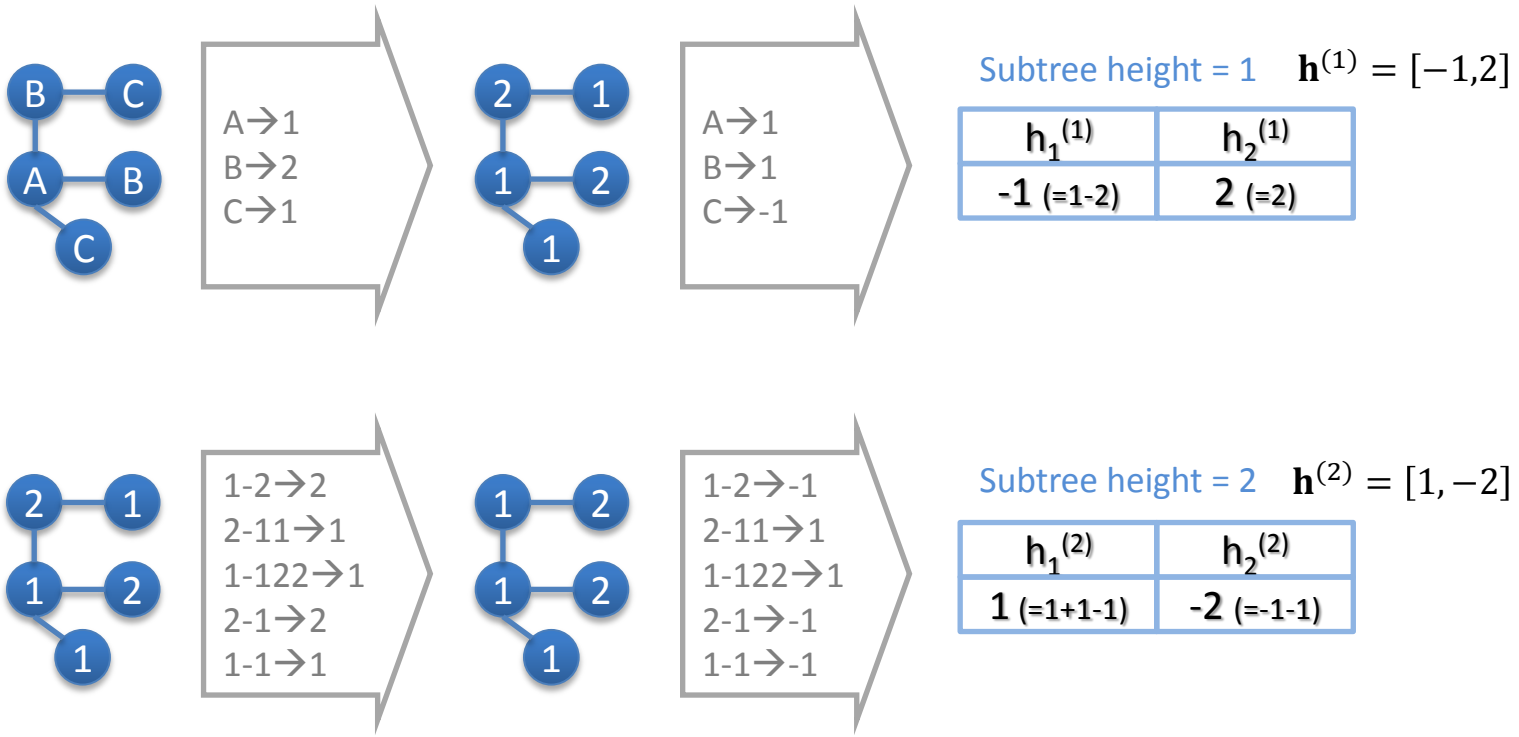
Idea

- Observations
  - Subtree Kernels are efficient but cannot deal with emerging subtree patterns
  - Feature hashing can address emerging features but only applicable to vectors

- Why not apply feature hashing to each level of subtrees?
  - Keep the effectiveness of subtree kernels
  - Address the infinite feature (subtree pattern) problem

Nested Subtree Hashing

- Alternate “feature hashing” and “subtree labeling” level by level



Analysis

- NSH is a sparse linear projection $\mathbf{h}^{(r)} = \mathbf{R}^{(r)^{\mathrm{T}}}\mathbf{x}^{(r)}$

$$\mathbf{h}^{(1)} = \mathbf{R}^{(1)^{\mathrm{T}}}\mathbf{x}^{(1)} = \mathbf{U}^{(1)^{\mathrm{T}}}\mathbf{V}^{(1)}\mathbf{x}^{(1)}$$
$$\mathbf{h}^{(2)} = \mathbf{R}^{(2)^{\mathrm{T}}}\mathbf{x}^{(2)} = \mathbf{U}^{(2)^{\mathrm{T}}}\mathbf{V}^{(2)}\left[\otimes \mathbf{U}^{(1)}\right]^{\mathrm{T}}\mathbf{x}^{(2)}$$
$$\mathbf{h}^{(3)} = \mathbf{R}^{(3)^{\mathrm{T}}}\mathbf{x}^{(3)} = \mathbf{U}^{(3)^{\mathrm{T}}}\mathbf{V}^{(3)}\left[\otimes\left(\left(\otimes \mathbf{U}^{(1)}\right)\mathbf{U}^{(2)}\right)\right]^{\mathrm{T}}\mathbf{x}^{(3)}$$
$$\mathbf{h}^{(4)} = \cdots$$

- The projection matrix is generated using the two random hash functions

$$\mathbf{U}_{ij}^{(r)} = \begin{cases} 1 & \text{if } \hbar\left(str_i^{(r)}\right) = j \\ 0 & \text{otherwise} \end{cases}, \qquad \mathbf{V}_{ij}^{(r)} = \begin{cases} \delta\left(str_i^{(r)}\right) & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

Properties

- An unbiased and highly concentrated estimator of the subtree kernel

$$\mathbb{E}\left[\left\langle \mathbf{h}_i^{(r)}, \mathbf{h}_j^{(r)} \right\rangle\right] = \mathbb{E}\left[\left\langle \mathbf{R}^{(r)^{\mathrm{T}}} \mathbf{x}_i^{(r)}, \mathbf{R}^{(r)^{\mathrm{T}}} \mathbf{x}_j^{(r)} \right\rangle\right] = \left\langle \mathbf{x}_i^{(r)}, \mathbf{x}_j^{(r)} \right\rangle$$

- The bound of the convergence rate becomes tighter as (*r*) increases

- More efficient in both time and space, compared to the subtree kernel

- More robust to tolerate concept drift over a stream of graphs

Datasets

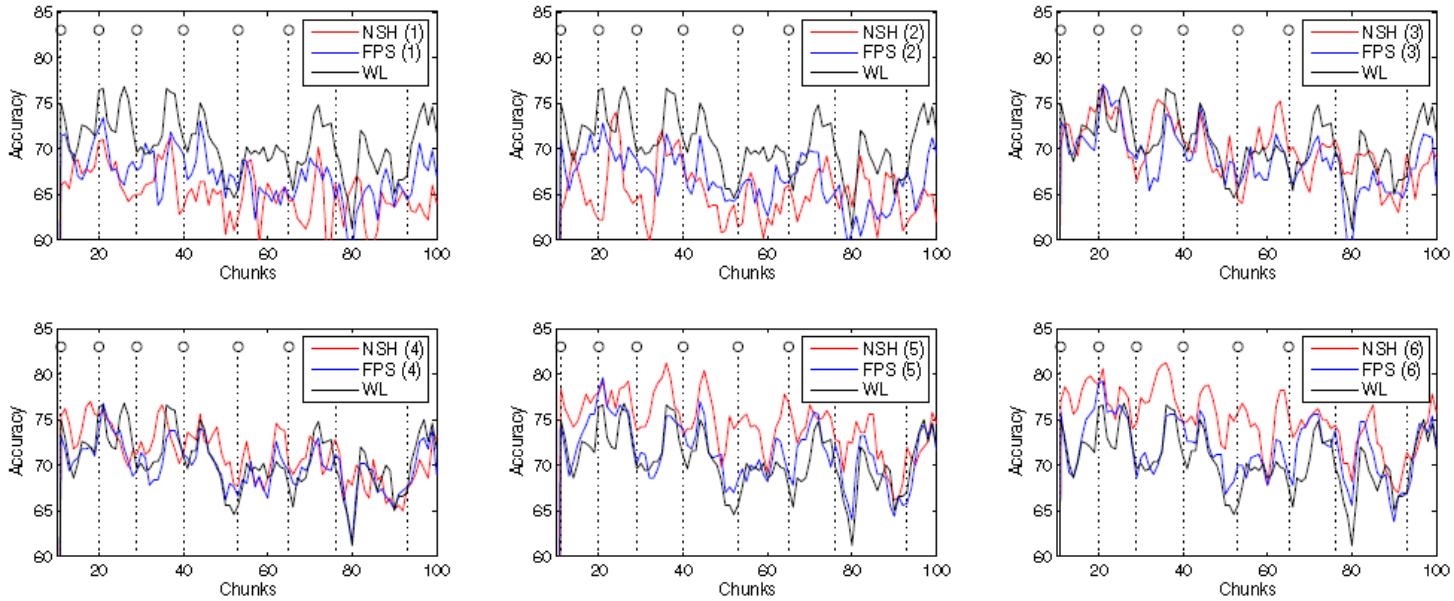
- NCI cancer screen datasets (chemical compounds)
  - Randomly select a negative subset to make each dataset balanced
  - Simulate a stream of graphs by sequentially reading the nine graph sets

Bioassay-ID (Data sets)	Compounds (Graphs)	Active (Pos)	Description
NCI1	42161	2232	Lung cancer
NCI33	41860	1806	Melanoma
NCI41	28547	1690	Prostate cancer
NCI47	42133	2181	Nervous sys. tumor
NCI81	42401	2620	Colon cancer
NCI83	28958	2437	Breast cancer
NCI109	42382	2252	Ovarian tumor
NCI123	41806	3388	Leukemia
NCI145	41850	2120	Renal cancer

Results

- "NSH (n)" denote different dimensionality settings
- "WL" uses the real number of subtree patterns in the dataset

Kernels	r = 1	r = 2	r = 3	r = 4	r = 5	r = 6
NSH (1)	30	50	50	50	50	50
NSH (2)	30	100	100	100	100	100
NSH (3)	30	500	500	500	500	500
NSH (4)	30	500	1000	1000	1000	1000
NSH (5)	30	500	5000	5000	5000	5000
NSH (6)	30	500	5000	10000	10000	10000

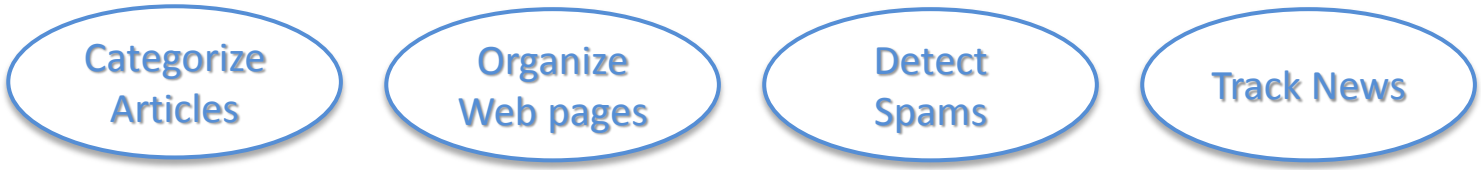


Outline

- Recursive Hashing for Structure Data
- Algorithm 1: Nested Subtree Hashing (NSH)
- Algorithm 2: Recursive Minwise Hashing (RMH)

Text Classification

- Bag-of-words (BoW) model
  - The occurrence (frequency) of each word is used as a feature
  - Disregarding grammar and even word order



- Limitations of a “flat-set” representation
  - Cannot preserve context
  - Lose hierarchical semantics

- We propose a “nested-set ” representation

Flat-sets (BoW)

Texts from Google
1. Technical blog about Google Maps Embed **video** and image in **Google** maps for multimedia **news** ... www.mulinblog.com/.../embed-**video**-and-images-in-**google**-maps-guide... ▾ Mar 6, 2013 - Embed **video** and images in **Google** maps: A beginner's guide for multimedia ... Using **Flickr** and **Youtube** to host your photos and **video**, you can add ... Here's how to embed the photo: when **viewing** your photo on **Flickr**, click
2. Abstract of a SDM-13 paper on Transfer Learning **Multi**-Transfer: **Transfer Learning** with **Multiple** Views and **Multiple** ... knowledgecenter.siam.org/190SDM/ ▾ by B Tan - Related articles FM and **Google News**. **different** vocabularies of **google news**. ... on **Youtube**, actually proposed to place **transfer learning** under the **multi**- the piece of ... as **Transfer Learning** with **Multi**- **transfer learning** problem where **source** and ... FM and im- lem, **transfer learning** aims to borrow knowledge from ages from **Flickr** may have ...
3. Abstract of a SDM-13 paper on Transfer Learning On Handling Negative **Transfer** and Imbalanced **Distributions** in ... **knowledge**center.siam.org/50SDM/ ▾ by J Gao - Related articles As there are usually **multi**- and sentiment classification [12] demonstrate the power ple ... **knowledge** can be **transferred**, of **transfer** learning. **multiple source transfer** ... In this paper, we **propose** methods to many applications due to the existence of a **novel two-phase framework** to effectively **transfer** knowl- irrelevant **sources** ...

sim(1,2) > sim(2,3) ???

1. {embed **video** image **Google** map beginner guide multimedia use **Flickr** **Youtube** host photo video add view}
2. {FM **Google** news different vocabulary available **Youtube** actually propose place transfer learning **multi** problem source **video** borrow knowledge age **Flickr**}
3. {usually multi sentiment classification demonstrate power knowledge transfer learning multiple source paper propose method application existence novel two-phase framework irrelevant source}

Nested-sets

Texts from Google
1. Technical blog about Google Maps Embed **video** and image in **Google** maps for multimedia **news** ... www.mulinblog.com/.../embed-**video**-and-images-in-**google**-maps-guide... Mar 6, 2013 - Embed **video** and images in **Google** maps: A beginner's guide for multimedia ... Using **Flickr** and **Youtube** to host your photos and **video**, you can add ... Here's how to embed the photo: when **viewing** your photo on **Flickr**, click
2. Abstract of a SDM-13 paper on Transfer Learning **Multi**-Transfer: **Transfer Learning** with **Multiple** Views and **Multiple** ... knowledgecenter.siam.org/190SDM/ by B Tan - Related articles FM and **Google News**. **different** vocabularies of **google news**. ... on **Youtube**, actually proposed to place **transfer learning** under the **multi**- the piece of ... as **Transfer Learning** with **Multi**- **transfer learning** problem where **source** and ... FM and im- lem, **transfer learning** aims to borrow knowledge from ages from **Flickr** may have ...
3. Abstract of a SDM-13 paper on Transfer Learning On Handling Negative **Transfer** and Imbalanced **Distributions** in ... **knowledge**center.siam.org/50SDM/ by J Gao - Related articles As there are usually **multi**- and sentiment classification [12] demonstrate the power ple ... **knowledge** can be **transferred**, of **transfer** learning. **multiple source transfer** ... In this paper, we **propose** methods to many applications due to the existence of a **novel two-phase framework** to effectively **transfer** knowl- irrelevant **sources** ...

1. **{{**embed video image Google map beginner guide multi-media**}** **{**use Flickr Youtube host Photo video add**}** **{**embed photo view Flickr click**}}**

2. **{{**FM Google news different vocabulary**}** **{**available Youtube actually propose place transfer learning multi**}** **{**problem transfer learning multi source**}** **{**.....**}}**

3. **{{**usually multi sentiment classification demonstrate power**}** **{**knowledge transfer learning multiple source**}** **{**paper propose method application existence novel two-phase framework transfer irrelevant source**}}**

sim(1,2) < sim(2,3) !!!

Our Goal

- Limitations of a “flat-set” representation
  - Cannot preserve context
  - Lose hierarchical semantics

- Aims
  - Fast similarity computation between texts
  - Preserve context information
  - Preserve hierarchical semantics

(Preliminary) Min-wise Hashing

- Jaccard similarity of two flat-sets

$$J(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}$$

- Min-hash is the first position of element after a random permutation

$$\hbar(S) = \min\big(\pi(S)\big)$$

- Property of min-wise hashing

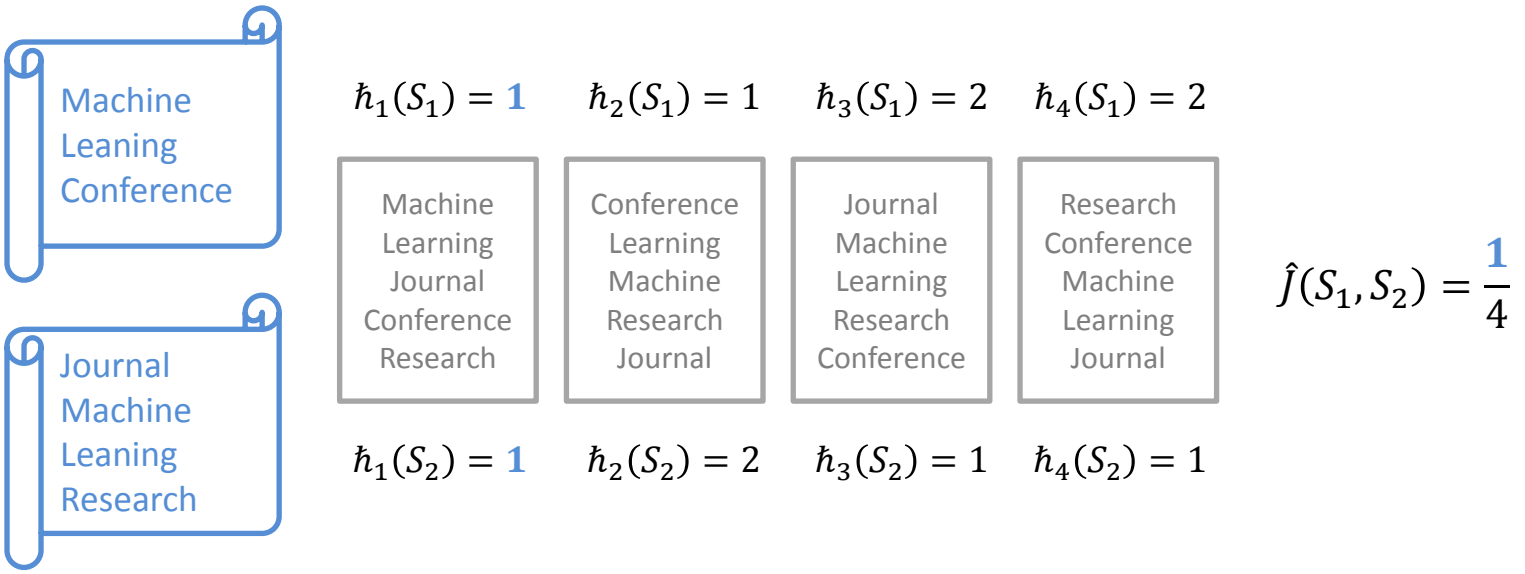
$$J(S_1, S_2) = \mathbb{E}\big[\mathbf{1}\big(\hbar(S_1) = \hbar(S_2)\big)\big]$$

- Empirically using *K* independent random permutations for approximation

$$\hat{J}(S_1, S_2) = \frac{\sum_{k=1}^{K} \mathbf{1}(\hbar_k(S_1) = \hbar_k(S_2))}{K}$$

Text Similarity

- Min-hash based text classification
  - Represent each text as a bag-of-words
  - Compute pair-wise Jaccard similarities based on min-hashes
  - Embed the similarity matrix into a kernel machine



Multi-level Exchangeability

- Nested-set representation

[Paragraph-level]
{A1, A2}

[Sentence-level]
A1 = {B11, B12}
A2 = {B21, B22}

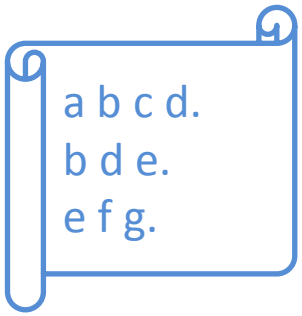
[Word-level]
B11 = {MLRG, seminar, good}
B12 = {attend, every, week}
B21 = {presenter, week}
B22 = {topic, recursive, hashing}

Semantic hierarchy preserved

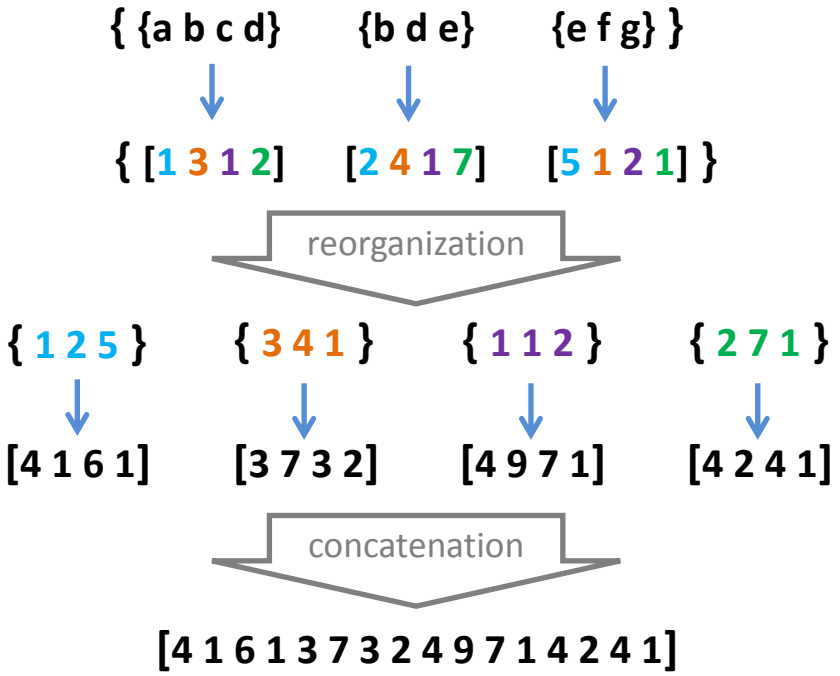
Context preserved

Recursive Min-wise Hashing

- The similarity between nested-sets is expected to satisfy
  - Set-elements can be compared in probability
  - Similarities at low levels (e.g., words) can be propagated to high levels (e.g., sentences)



"↓" denotes min-hash



Analysis

- Jaccard similarity of two nested-sets

$$J\left(S_1^{(1)}, S_2^{(1)}\right) = \mathbb{E}\left[\mathbf{1}\left(\hbar_1\left(S_1^{(1)}\right) = \hbar_1\left(S_2^{(1)}\right)\right)\right]$$
$$J\left(S_1^{(2)}, S_2^{(2)}\right) = \mathbb{E}\left[\mathbf{1}\left(h_2\left(\left\{\hbar_1\left(S_{1,*}^{(1)}\right)\right\}\right) = h_2\left(\left\{\hbar_1\left(S_{2,*}^{(1)}\right)\right\}\right)\right)\right]$$
$$J\left(S_1^{(3)}, S_2^{(3)}\right) = \cdots$$
$$J\left(S_1^{(r)}, S_2^{(r)}\right) = \mathbb{E}\left[\mathbf{1}\left(h_r\left(\left\{\hbar_{(r-1)}\left(\ldots\left\{h_1\left(S_{1,*}^{(1)}\right)\right\}\ldots\right)\right\}\right) = h_r\left(\left\{\hbar_{(r-1)}\left(\ldots\left\{h_1\left(S_{2,*}^{(1)}\right)\right\}\ldots\right)\right\}\right)\right)\right]$$

- Bounds – Locality Sensitive Hashing

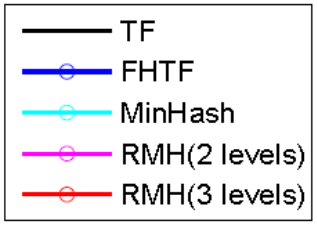
$$\text{If } J\left(S_1^{(r)}, S_2^{(r)}\right) \geq s, \text{then } \hat{J}\left(S_1^{(r)}, S_2^{(r)}\right) \geq (1-\delta)s \text{ with probability at least } 1-\epsilon$$
$$\text{If } J\left(S_1^{(r)}, S_2^{(r)}\right) \leq s, \text{then } \hat{J}\left(S_1^{(r)}, S_2^{(r)}\right) \leq (1+\delta)s \text{ with probability at least } 1-\epsilon$$

$$\text{where } 0 < \delta < 1, \epsilon > 0, \text{and } K > (2\delta^{-2}s^{-1}\log\epsilon^{-1})^{1/r}$$

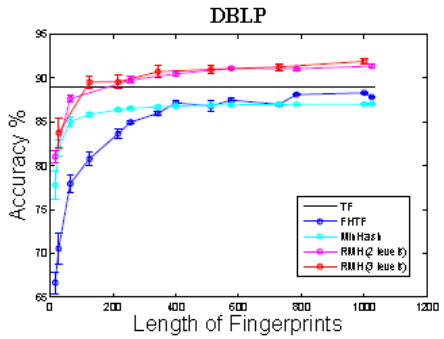
Datasets

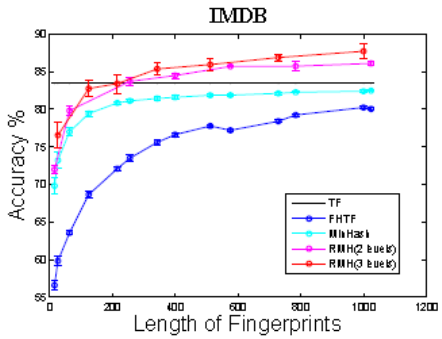
- Paper Abstract Classification (DBLP)
  - Extract abstracts from 15,195 papers as a binary classification task (AI vs. CV)
- Review Polarity Classification (IMDB)
  - 20,000 movie reviews with balanced positive/negative samples as a binary classification task
- Review Category Classification (Amazon)
  - 10,000 Book reviews and 10,000 Music reviews to form a binary review-category classification task

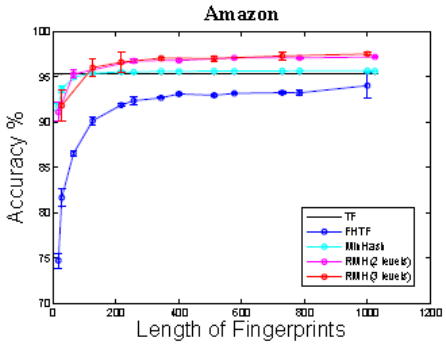
Results

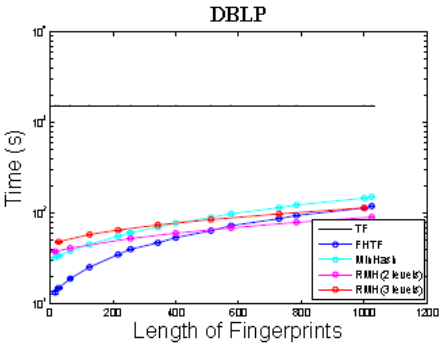


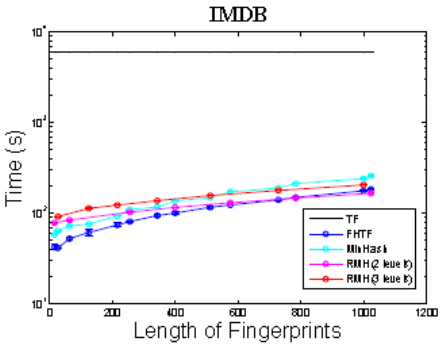
(TF)	Term Frequency + SVM
(FHTF)	Feature Hashing on TF + SVM
(MinHash)	Min-wise hashing + SVM
(RMH)	Recursive Min-wise Hashing + SVM

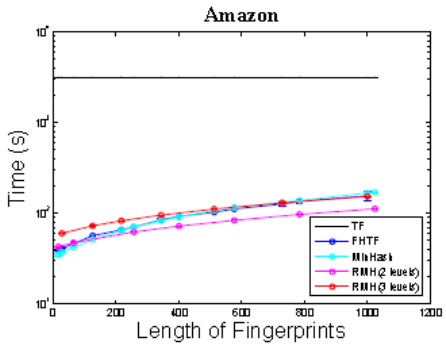












Discussion

- Conclusion
  - Locality-sensitive hashing for structured data
  - Propose a new hashing paradigm – Recursive Hashing
  - Broaden the application areas of hashing techniques

- Future Work
  - Applicable to many hierarchically represented objects (e.g., images)
  - Extendable with other basic LSH schemes (e.g., random projections)

Thanks