Novel Representations, Regularization & Distances for Text Classification

Konstantinos Skianis
Ph.D. thesis defense
LIX, École Polytechnique, France

Supervisor: Michalis Vazirgiannis
Co-supervisor: Guillaume Wisniewski

March 1, 2019

Abstract

> *"The complete meaning of a word is always contextual, and no study of meaning apart from context can be taken seriously."*
>
> — John Rupert Firth, 1935

- Fields: natural language processing, machine learning
  - help machines perceive text as we do
- Thesis: novel components to fully exploit prior knowledge on text
  - better understanding by encoding more information
- Application: text mining
  - text = preferred mean of information storage and knowledge transfer
  - big data era → scale too large and too time-consuming for humans

Contributions

- Derive new graph-based text representations and schemes
- Linguistic groups for structured regularization
- Novel approach for overlapping group regularization
- Enhance existing distance techniques in word embeddings
- Design a neural architecture for distance-based classification

Publications (1/2)

(1) Fragkiskos D. Malliaros and Konstantinos Skianis (2015a). "Graph-based term weighting for text categorization". In: *Someris, ASONAM*. ACM, pp. 1473–1479

(2) Konstantinos Skianis et al. (2016a). "Regularizing Text Categorization with Clusters of Words". In: *EMNLP*, pp. 1827–1837

(3) Konstantinos Skianis et al. (2018a). "Fusing Document, Collection and Label Graph-based Representations with Word Embeddings for Text Classification". In: *TextGraphs, NAACL*, pp. 49–58 (Best Paper Award)

(4) Konstantinos Skianis et al. (2018b). "Orthogonal Matching Pursuit for Text Classification". In: *W-NUT, EMNLP*

(5) Konstantinos Skianis et al. (2019a). "Boosting Tricks for Word Mover's Distance". Manuscript (Submitted in ICWSM 2019)

(6) Konstantinos Skianis et al. (2019c). "Rep the Set: Neural Networks for Learning Set Representations". Manuscript (Submitted in ICML 2019)

(7) Konstantinos Skianis et al. (2019b). "Group Lasso for Linguistic Structured Attention". Manuscript (Will be submitted in TACL 2019)

Publications (2/2)

Not covered in this presentation:

(8) Antoine J-P Tixier et al. (2016). "GoWvis: a web application for Graph-of-Words-based text visualization and summarization". In: *ACL 2016*. ACL, p. 151

(9) Konstantinos Skianis et al. (2016b). "SPREADVIZ: Analytics and Visualization of Spreading Processes in Social Networks". In: *Demo, ICDM*. IEEE, pp. 1324–1327

(10) Giannis Nikolentzos et al. (2018). "Kernel graph convolutional neural networks". In: *International Conference on Artificial Neural Networks*. Springer, Cham, pp. 22–32

(11) Giannis Siglidis et al. (2018). "GraKeL: A Graph Kernel Library in Python". In: *arXiv preprint arXiv:1806.02193*

(12) Stamatis Outsios et al. (2018). "Word Embeddings from Large-Scale Greek Web content". In: *Spoken Language Technology (SLT)*

Outline

Outline

The Machine Learning Era

Solving real-world problems

- data-driven approaches $\rightarrow$ breakthroughs in research & industry
- autonomous driving, protein structure prediction, virtual assistants

Why now?

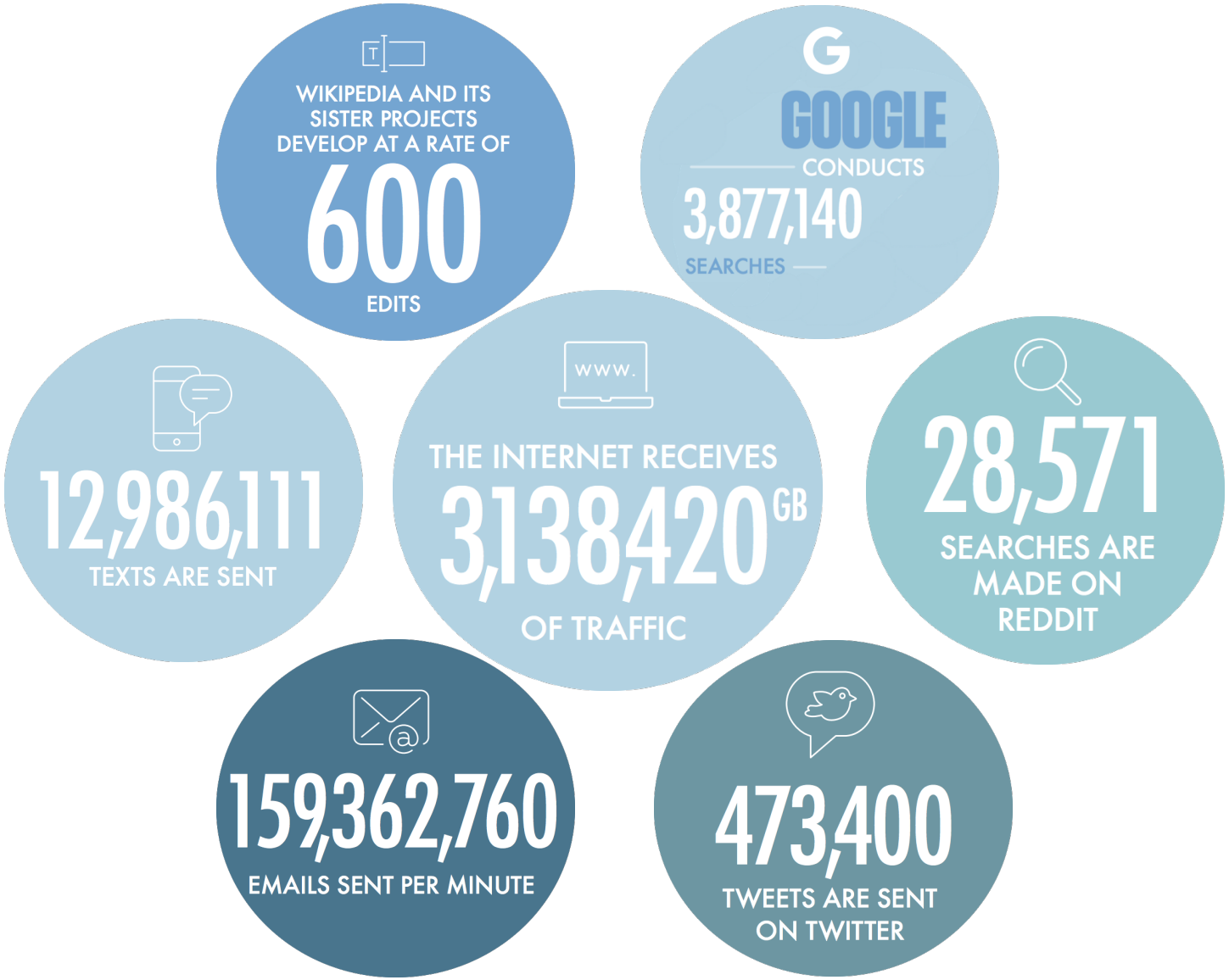
- strong AI algorithms
- large hardware power available
- huge data

What?[1]

- 2.5 quintillion (10^{18}) bytes of data created each day
- Internet of Things (IoT)
- 90% was generated in the last two years
- image, video, sequences, text

[1] `https://www.forbes.com/`

Text is everywhere[2]



[2] https://www.domo.com/learn/data-never-sleeps-6

Motivation

How can we harvest the full potential of all this textual data?

- Distributional hypothesis (Harris, 1954): words that occur in the same contexts tend to have similar meanings
- *"You shall know a word by the company it keeps"* (Firth, 1957)
- Text consists of latent concepts corresponding to distributions of observable words
- We need effective and efficient methods to mine them

Outline

Preliminary NLP concepts

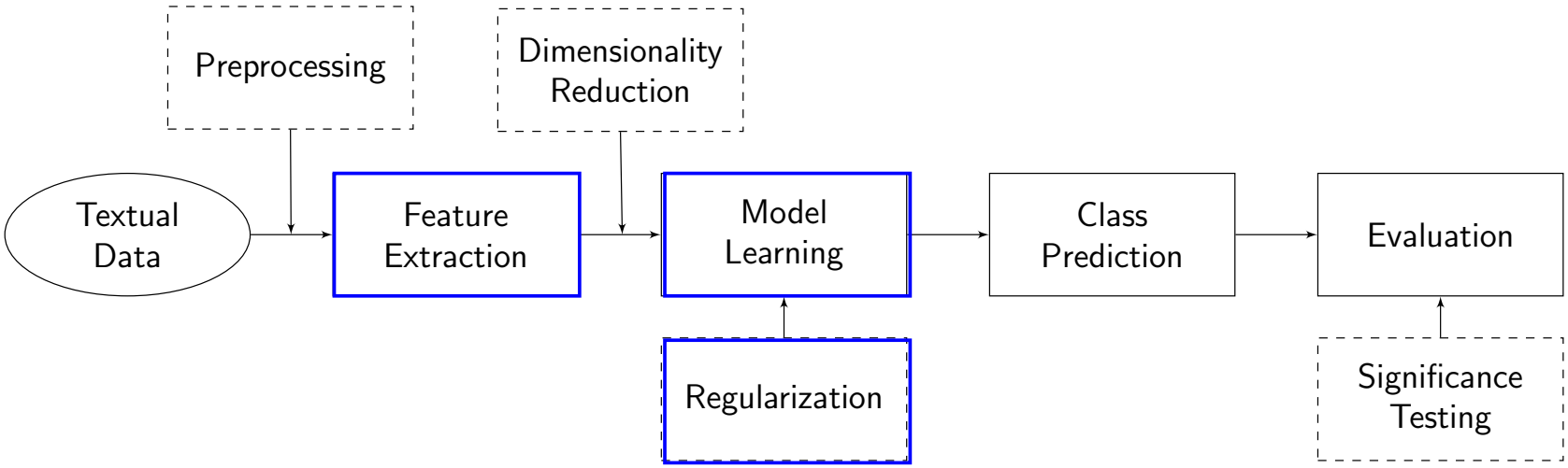
- A dataset of data points corresponds to a collection of documents
- A document is a piece of raw text we are interested in as a whole
  - a Web page, a tweet, a user review, a news article, etc.
- A document is a sequence of words: $d = (t_1, t_2, \ldots, t_{|d|})$
- Each word belongs to a common vocabulary
- A term is a processed word, i. e. we apply dimensionality reduction to the vocabulary (e. g., stemming or stopword removal)

Application

Text classification (TC)

Assigning categories to documents (web page, book, media articles etc.)

- TC still one of the most popular tasks
- Spam filtering, email routing, sentiment analysis, qa, chatbots



Basic pipeline of the text classification task.

Evaluation

- Humans decide what best means, i.e. provide ground-truth data, a gold standard for the expected outcome
  - TC $\rightarrow$ set of documents with golden class labels
- We compare a system's output with the expected outcome:
  - Accuracy: proportion of good predictions
  - macro-average F1-score: harmonic mean between precision and recall, averaged over documents or categories
- Statistical significance:
  - quantify the improvement and decide if we consider it meaningful or simply due to chance

Outline

Traditional steps of TC

Problem: given a document, choose the best classification label

- Feature extraction + supervised learning
- Each data point (document) is represented as a feature vector
  - Bag-of-Words model $\Rightarrow$ binary, frequency
  - TF-IDF (Sparck Jones, 1972)
  - n-gram features (Baeza-Yates and Ribeiro-Neto, 1999)
- A classifier is learnt on a labeled training set of data points:
  - the most frequent category among closest data points, e. g., kNN
  - the category with maximum a posteriori, e. g., Naive Bayes
  - on which side of a separating hyperplane it falls, e. g., SVM

Main approaches

Bag-of-Words & Linear Classifiers

- Document is represented as a multiset of its terms
  ↪ fast and effective with simple classifiers
- The term independence assumption:
  ↪ disregarding co-occurence; keeping only the frequency
- *n*-gram model
  ↪ restrictive in capturing order of terms, huge dimensionality

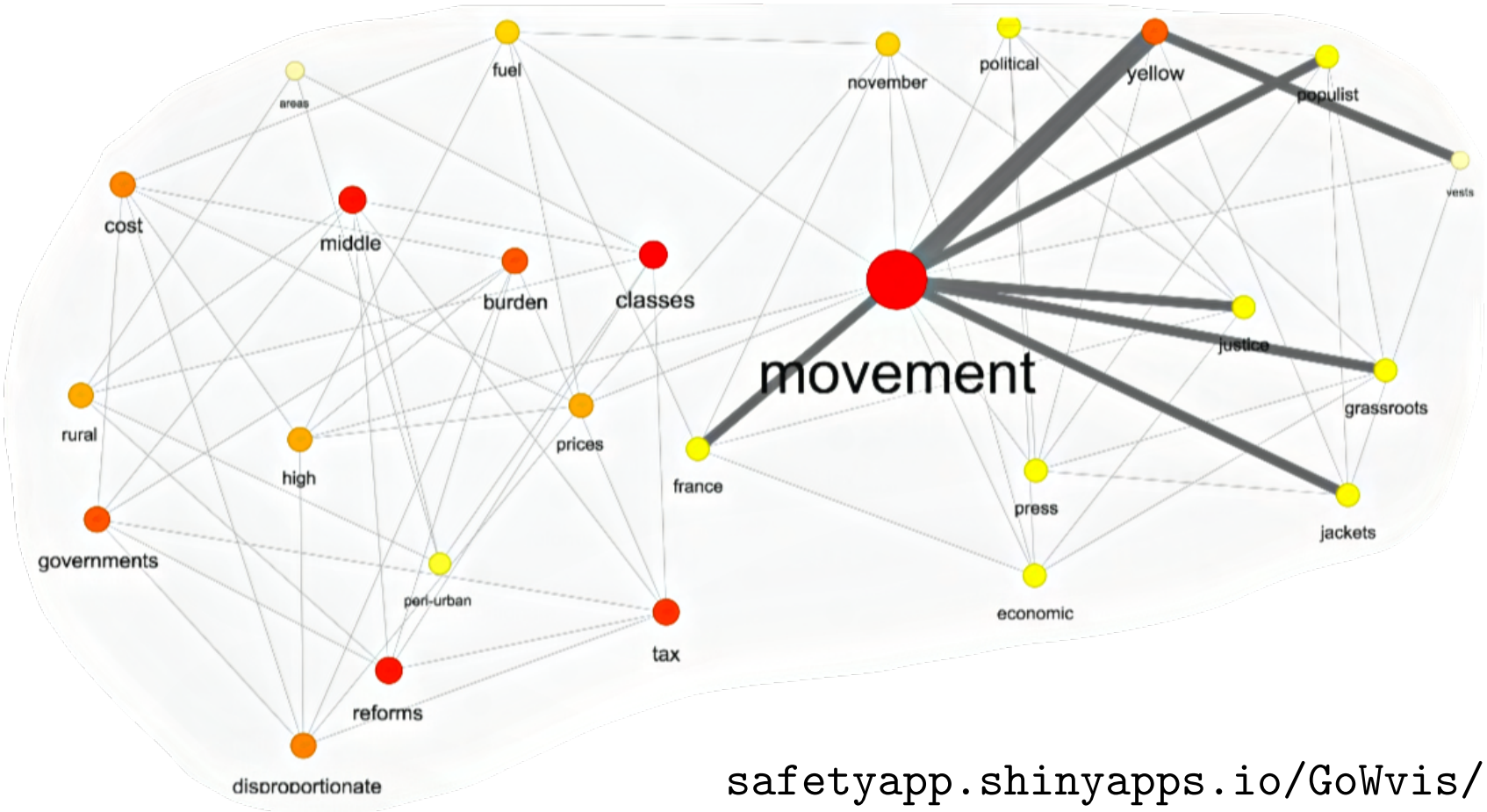
Continuous Vectors & Deep Learning

- Neural (Kim, 2014);(Johnson and Zhang, 2017)
  ↪ Current state-of-the-art results
  ↪ Large pre-trained embeddings needed
- Use the order of words with CNNs (Johnson and Zhang, 2015)
  ↪ Complex architectures with large resources (GPUs)
- Space and time limitations may arise:
  ↪ Computation can be expensive (Joulin et al., 2017)

→ How can we extract more meaningful features?

Graph representations

The yellow vests movement or yellow jackets movement is according to some press a populist, grassroots political movement for economic justice that began in France in November 2018. The movement is motivated by rising fuel prices, high cost of living, and claims that a disproportionate burden of the government's tax reforms were falling on the working and middle classes, especially in rural and peri-urban areas.



safetyapp.shinyapps.io/GoWvis/

Graph-based approaches

Graph-based Text Mining, NLP and IR

- TextRank (Mihalcea and Tarau, 2004)
- Graph-of-Words (Rousseau and Vazirgiannis, 2013)

Graph-mining for TC

- Frequent subgraphs (Deshpande et al., 2005);(Nikolentzos et al., 2017) $\hookrightarrow$ frequent subgraph mining $\rightarrow$ high complexity
- Random walks, other graph centrality criteria (Hassan et al., 2007);(Malliaros and Skianis, 2015)

Centrality criteria

- $\text{Degree}(i) = \frac{|\mathcal{N}(i)|}{|V|-1}$
- $\text{Closeness}(i) = \frac{|V|-1}{\sum_{j \in V} dist(i,j)}$, the sum of the length of the shortest paths between the node and all other nodes in the graph
- $\text{Pagerank}(i) = \frac{1-\alpha}{|V|} + \alpha \sum_{\forall (j,i) \in E} \frac{\text{PR}(j)}{\text{out-deg}(j)}$

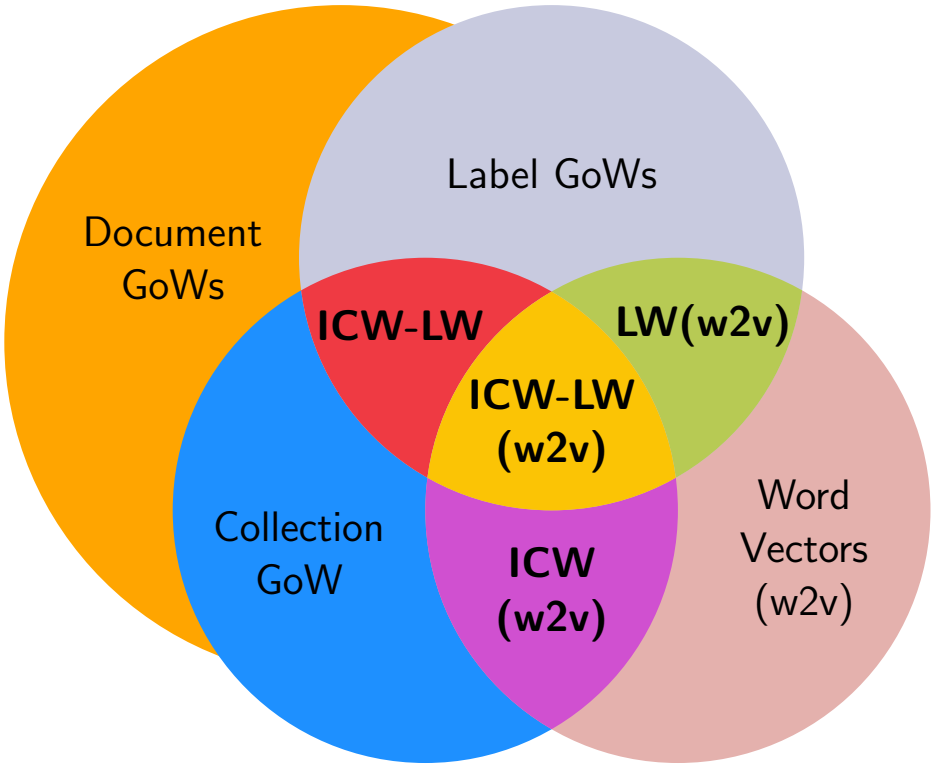
Contributions

Why graphs?

- powerful representation
- huge research literature

Bringing graphs to NLP:

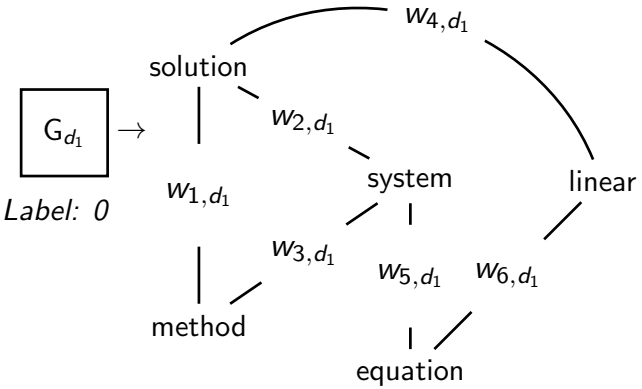
- consider info about *n*-grams
  - expressed by paths in the graph
  - keep the dimensionality low (compared to *n*-grams)
- introduce collection-level GoW
- blend document, collection and label GoWs
- integrate word vector similarities as weights in edges



Intersections reveal the derived schemes.

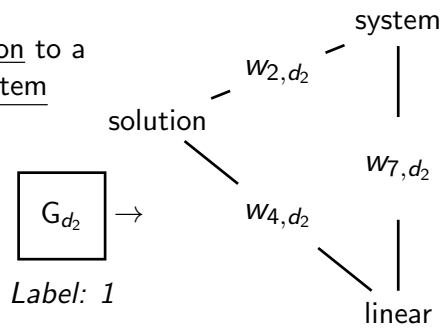
Document, collection and label GoWs

d_1: A method for the solution of systems of linear equations



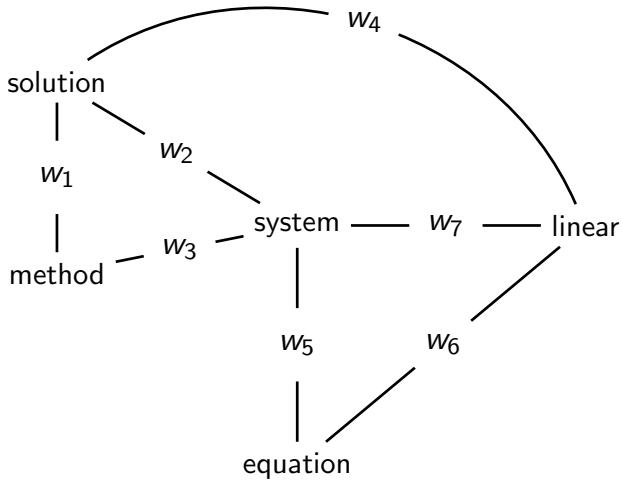
Label: 0

d_2: A solution to a linear system

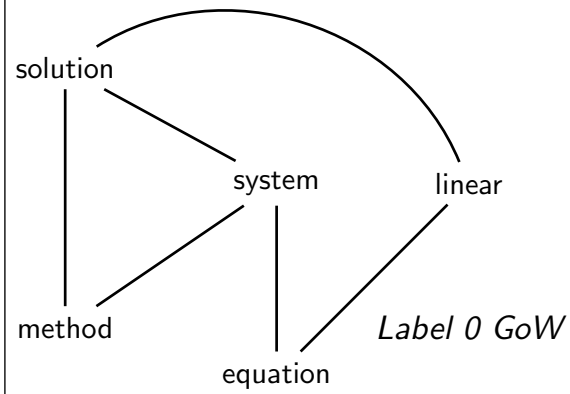


Label: 1

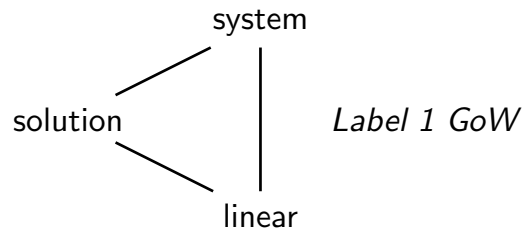
Document-level GoWs for d_1, d_2.



Collection-level GoW $\mathcal{G}$.



Label 0 GoW



Label 1 GoW

Label GoWs for two classes.

Proposed weighting schemes

On the collection GoW, the "Inverse Collection Weight" metric:

$$\text{ICW}(t, \mathcal{D}) = \frac{\max_{v \in \mathcal{D}} \text{TW}(v, \mathcal{D})}{\text{TW}(t, \mathcal{D})}$$

Then, the TW-ICW metric becomes:

$$\text{TW-ICW}(t, d) = \text{TW}(t, d) \times \log(\text{ICW}(t, \mathcal{D}))$$

Given the label GoWs, our weighting scheme is a variant of TW-CRC (Shanavas et al., 2016):

$$\text{LW}(t) = \frac{\max(\deg(t, L))}{\max(\text{avg}(\deg(t, L)), \min(\deg(L))}$$

Last, the TW-ICW-LW metric becomes:

$$\text{TW-ICW-LW}(t, d) = \text{TW}(t, d) \times \log(\text{ICW}(t, \mathcal{D}) \times \text{LW}(t))$$

Edge weighting using word embeddings

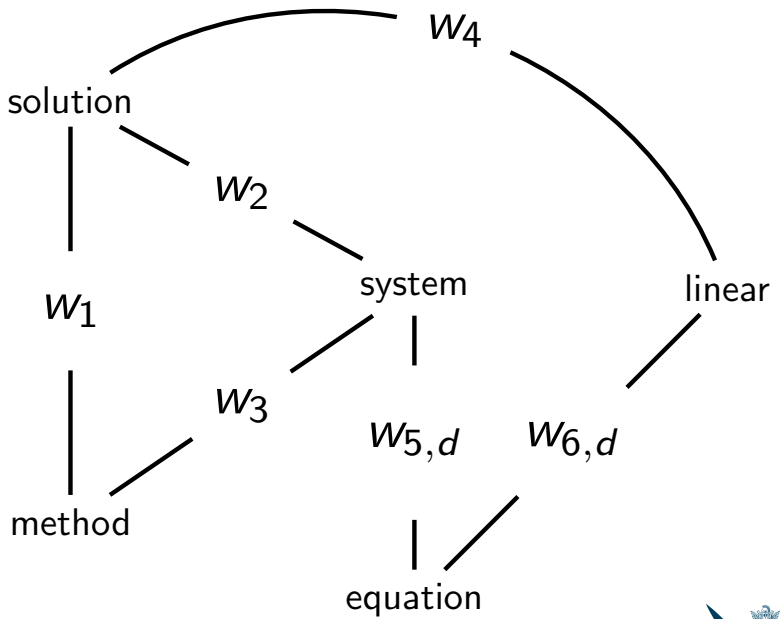
Making the most-out-of graphs via word vectors

Use rich word embeddings in order to extract relationships between terms.

- Inject similarities as weights on edges
  - Reward semantically close words in the document GoW (TW)
  - Penalize them in the collection GoW (ICW)

$$\mathrm{w}(t_1, t_2) = 1 - \frac{\mathrm{sim}^{-1}(t_1, t_2)}{\pi}$$

d_1: A method for the solution of systems of linear equations



Datasets & setup

- Linear SVMs with grid search cross-validation for tuning C
- Removed stopwords
- No stemming or lowercase transformation, to match Google's vectors
- Multi-core document and collection graph construction

	Train	Test	Voc	Avg	#w2v	#ICW
IMDB	1,340	660	32,844	343	27,462	352K
WebKB	2,803	1,396	23,206	179	20,990	273K
20NG	11,293	7,528	62,752	155	54,892	1.7M
Amazon	5,359	2,640	19,980	65	19,646	274K
Reuters	5,485	2,189	11,965	66	9,218	163K
Subj.	6,694	3,293	8,639	11	8,097	58K

#ICW: number of edges in the collection-level graph;
#w2v: number of words in pre-trained vectors.

Results (1/2)

Methods	20NG (MAX)				IMDB (SUM)				SUBJECTIVITY (MAX)			
	$w = 3$		$w = 4$		$w = 2$		$w = 3$		$w = 6$		$w = 7$	
	F1	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**
TF	80.88	81.55	-		84.23	84.24	-		88.42	88.43	-	
w2v	74.43	75.75	-		82.57	82.57	-		87.67	87.67	-	
TF-binary (ngrams)	81.64	82.11*	-		83.02	83.03	-		87.51	87.51	-	
TW (degree)	82.37	83.00*	82.21	82.83*	84.82	84.84	84.67	84.69	88.33	88.33	89.00	89.00*
TW (w2v)	81.88	82.51*	82.21	82.87*	84.66	84.69	84.52	84.54	87.75	87.57	87.66	87.67
TF-IDF	82.44	83.01*	-		83.33	83.33	-		89.06	89.06*	-	
TF-IDF-w2v	82.52	83.09*	-		82.87	82.87	-		89.91	89.91*	-	
TW-IDF (degree)	84.75	85.47*	84.80	85.46*	82.86	82.87	83.02	83.03	89.33	89.34*	89.33	89.34*
TW-IDF (w2v)	84.66	85.32	84.46	85.13	83.47	83.48	83.31	83.33	86.42	86.42	86.51	86.51
TW-ICW (deg, deg)	85.24	85.80*	**85.41**	**86.05***	84.98	85.00	85.13	85.15	89.30	89.31*	89.61	89.61*
TW-ICW (w2v)	**85.33**	**85.93***	85.29	85.90*	85.12	85.15	84.82	84.84	89.61	89.61*	87.30	87.30
TW-ICW-LW (deg)	85.01	85.66*	85.02	85.66*	85.73	85.75	85.28	85.30	**90.12**	**90.13***	**90.27**	**90.28***
TW-ICW-LW (w2v)	82.56	83.11*	82.24	82.81*	85.29	85.30	84.39	84.39	87.70	87.70	87.70	87.70
TW-ICW-LW (pgr)	83.92	84.66	83.80	84.54	84.97	85.00	85.73	85.75	86.60	86.60	86.45	86.45
TW-ICW-LW (cl)	84.61	85.22	84.71	85.27	**87.27**	**87.27***	**86.06**	**86.06**	89.97	89.97*	90.09	90.10*

Macro-F1 and accuracy for window size w. Bold for best performance on each window size and blue for best overall on a dataset. * indicates statistical significance of improvement over TF at $p < 0.05$ using micro sign test.

Results (2/2)

Methods	Amazon (max)				WebKB (sum)				Reuters (max)			
	$w = 2$		$w = 3$		$w = 2$		$w = 3$		$w = 2$		$w = 3$	
	F1	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**
TF	80.68	80.68	-		90.31	91.91	-		91.51	96.34	-	
w2v	79.05	79.05	-		84.54	86.58	-		91.35	96.84	-	
TF-binary (ngrams)	79.84	79.84	-		91.22	92.85	-		86.33	95.34	-	
TW (degree)	80.07	80.07	80.41	80.41	91.69	92.64	91.45	92.49	93.58	97.53*	93.08	97.25*
TW (w2v)	80.07	80.07	79.54	79.54	91.70	92.64	91.00	92.06	93.09	97.35*	93.43	97.25*
TF-IDF	80.26	80.26	-		87.79	89.89	-		91.89	96.71	-	
TF-IDF-w2v	80.49	80.49	-		88.18	90.18	-		91.33	96.80	-	
TW-IDF (degree)	81.47	81.47*	81.55	81.55*	90.38	91.70	90.47	91.84	93.80	97.30*	93.13	97.35*
TW-IDF (w2v)	79.61	79.62	77.60	77.61	90.81	92.20	90.60	91.91	93.38	97.44*	**93.87**	**97.44***
TW-ICW (deg, deg)	82.08	82.08*	82.02	82.02*	91.72	92.78	91.42	92.49	92.91	97.35	93.59	97.39*
TW-ICW (w2v)	80.86	80.87*	78.82	78.82	91.58	92.64	91.84	92.85	93.57	97.30*	92.96	97.25
TW-ICW-LW (deg)	82.72	82.72*	82.91	82.91*	91.86	92.92	91.95	92.92	**93.88**	**97.53***	93.48	97.35*
TW-ICW-LW (w2v)	80.56	80.56	78.32	78.33	90.74	91.99	90.01	91.34	92.51	96.89	92.14	96.98
TW-ICW-LW (pgr)	82.23	82.23*	82.46	82.46*	91.18	92.20	92.23	93.07	93.38	97.35*	93.37	97.35*
TW-ICW-LW (cl)	**82.90**	**82.91***	**83.02**	**83.03***	**92.72**	**93.57***	**92.86**	**93.57***	93.12	97.25	92.87	97.21

Macro-F1 and accuracy for window size w. Bold for best performance on each window size and blue for best overall on a dataset. * indicates statistical significance of improvement over TF at $p < 0.05$ using micro sign test.

Comparison vs state-of-the-art methods

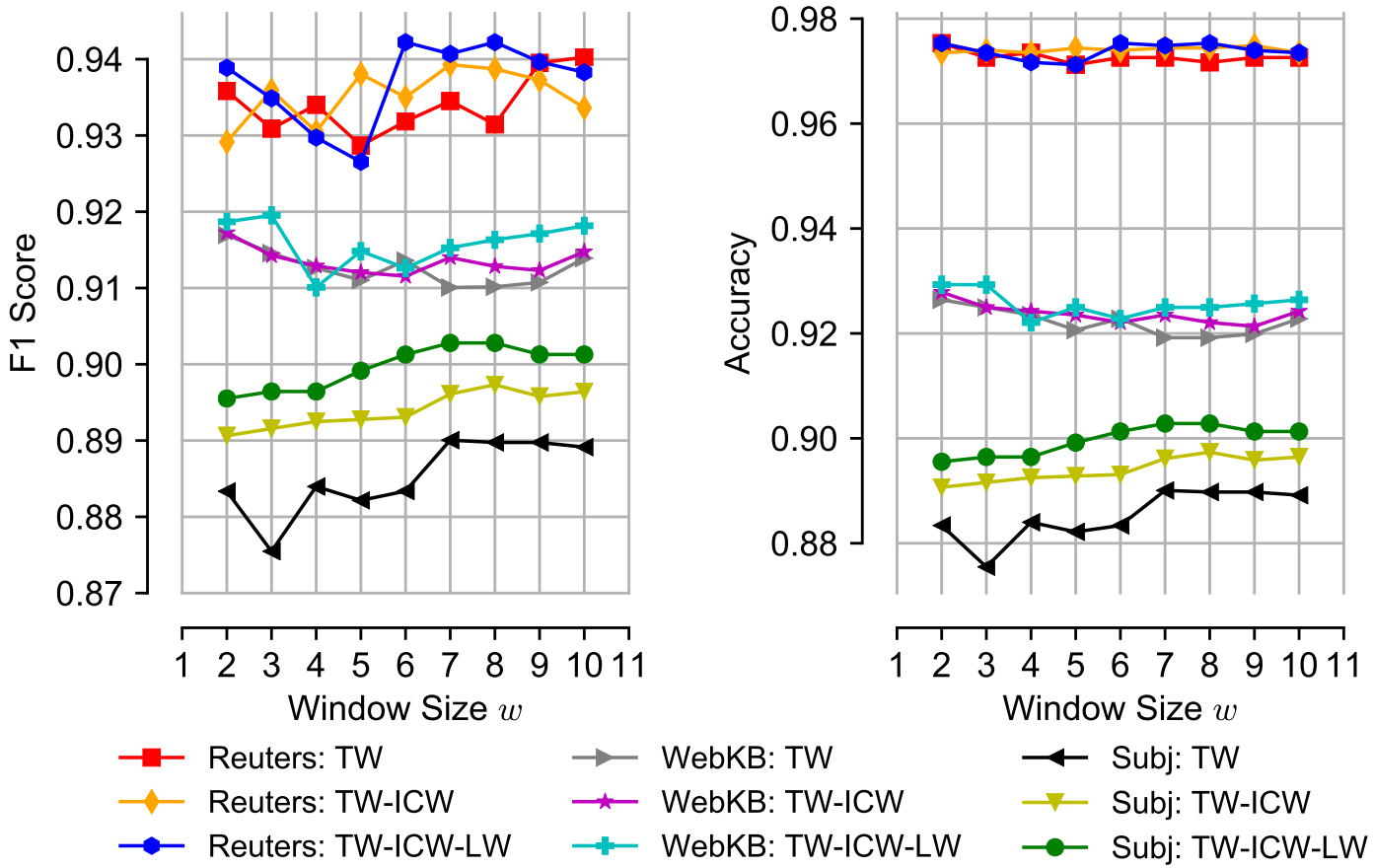
	20NG	IMDB	SUBJ.	AMAZON	WEBKB	REUTERS
CNN (no w2v, 20 ep.) (Kim, 2014)	83.19	74.09	88.16	80.68	88.17	94.75
CNN (w2v, 20 ep.)	81.92	64.09	89.04	82.08	84.05	95.80
FastText (100 ep.) (Joulin et al., 2017)	79.70	84.70	88.60	79.50	92.60	97.00
FastText (w2v, 5 ep.)	80.80	86.10	88.50	80.90	91.40	97.40
TextRank (Mihalcea and Tarau, 2004)	82.56	83.33	84.78	80.49	92.27	97.35
Word Attraction (Wang et al., 2015)	61.24	70.75	86.60	78.29	79.46	91.34
TW-CRC (Shanavas et al., 2016)	85.35	85.15	89.28	81.13	92.71	97.39
TW-ICW-LW (ours)	**86.05**	**87.27**	**90.28**	**83.03**	**93.57**	**97.53**

Comparison in accuracy(%) to deep learning and graph-based approaches.

Discussion

- With label graphs used, word vectors do not improve accuracy
  ↪ terms concerning different labels can be close in vector space
- Closeness in document GoW → best performance in 3/6
  ↪ can only have an affect in larger document lengths and when used along with label graphs

Examining the window size



F1 score (left) and accuracy (right) of TW, TW-ICW and TW-ICW-LW (all degree) on REUTERS, WEBKB and SUBJECTIVITY, for $w = \{2, \ldots, 10\}$.

Summary

Contributions

- a full graph-based framework for TC
- determine the importance of a term using node centrality criteria
  - document, collection and label level schemes
- add word-embedding information as weights on the edges

Future Directions

- could also be applied in IR, keyword extraction, summarization etc.
- *Graph-of-Documents*
  - Graph comparison via graph kernels (Borgwardt et al., 2007)
  - Word Mover's Distance (Kusner et al., 2015)
- Neural Message Passing (Gilmer et al., 2017) & Text GCN (Yao et al., 2019)

What more?

Use these GoW representations for regularization!

Outline

Prediction as loss minimization

Given a training set of N data points $\{(x^i, y^i)\}_{i=1...N}$, find the optimal set of feature weights $\boldsymbol{\theta}^*$ such that:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \underbrace{\underbrace{\sum_{i=1}^{N} \mathcal{L}(x^i, \theta, y^i)}_{\textbf{empirical risk}} + \underbrace{\lambda \Omega(\theta)}_{\textbf{penalty term}}}_{\textbf{expected risk}}$$

A logistic regression loss function:

$$\mathcal{L}(x, \boldsymbol{\theta}, y) = \log(1 + \exp(-y \boldsymbol{\theta}^T x)) \quad (1)$$

Regularization

- huge dimensionality in text
- address overfitting
- more sparse models
- use prior knowledge we may have on the features

Regularization

L_1 and L2 regularization (Tibshirani, 1996);(Hoerl and Kennard, 1970)

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{N} \mathcal{L}(x^i, \theta, y^i) + \lambda \sum_{j=1}^{p} |\theta_j| \text{(lasso)} \tag{2}$$

$$\theta^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{i=1}^{N} \mathcal{L}(x^i, \theta, y^i) + \lambda \sum_{j=1}^{p} \theta_j{}^2 \text{(ridge)} \tag{3}$$

Group lasso (Bakin, 1999);(Yuan and Lin, 2006)

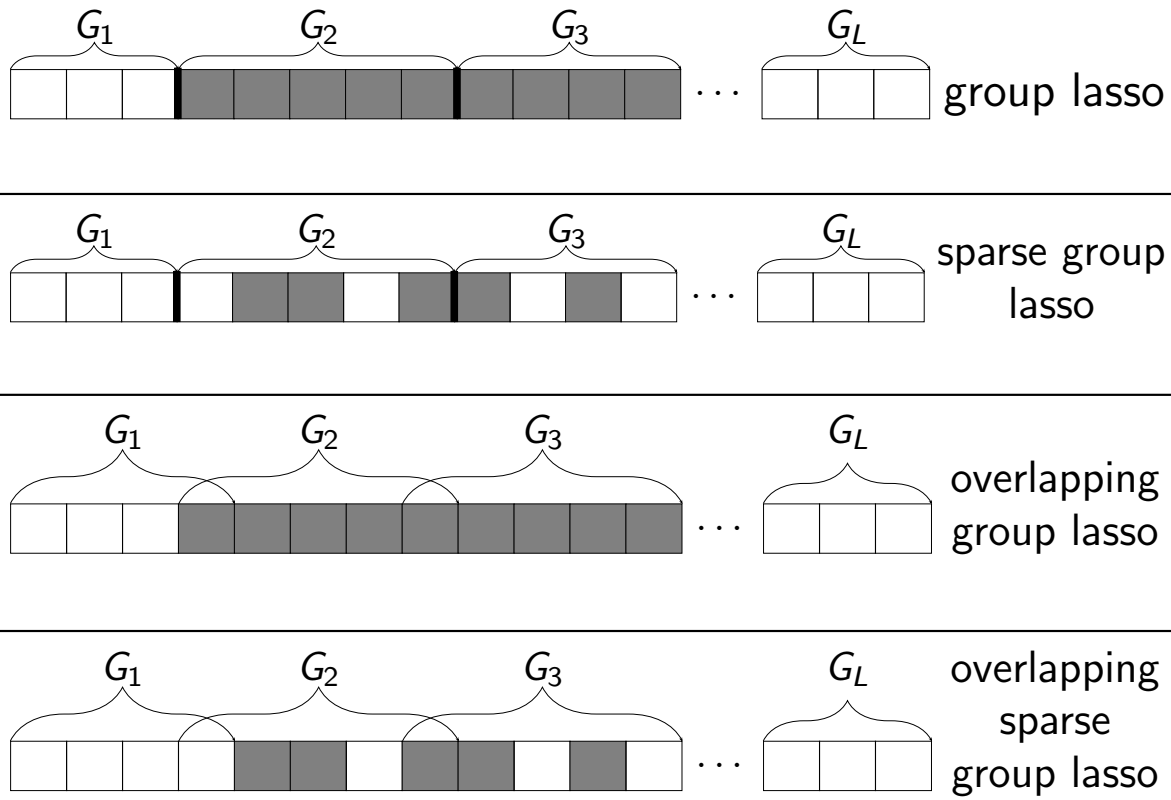
$$\Omega(\theta) = \lambda \sum_{g} \lambda_g \|\theta_g\|_2 \tag{4}$$

where θ_g is the subset of feature weights restricted to group g.

Linguistic structured regularizers

- Sentence, parse, Brown, LDA (Yogatama and Smith, 2014)

Group lasso variants



Grey boxes depict active features. While group lasso selects a whole group, sparse group lasso can select some group's features. In the overlapping case, groups can share features, while in the last, L_1 is applied inside each group.

ADMM for sparse overlapping group lasso

- The objective (Yogatama and Smith, 2014):

$$\min_{\boldsymbol{\theta},\mathrm{v}} \Omega_{las}(\boldsymbol{\theta}) + \Omega_{glas}(\mathrm{v}) + \mathcal{L}(\boldsymbol{\theta}) \tag{5}$$

$$\text{s.t. } \mathbf{v} = M\boldsymbol{\theta}$$

where $\mathbf{v}$ is a copy-vector of $\boldsymbol{\theta}$, M is an indicator matrix of size $L \times V$, linking $\boldsymbol{\theta}$ and their copies $\mathbf{v}$.

- An augmented Lagrangian problem is formed:

$$\Omega_{las}(\boldsymbol{\theta}) + \Omega_{glas}(\mathrm{v}) + \mathcal{L}(\boldsymbol{\theta}) + \mathrm{u}^\top(\mathrm{v} - M\boldsymbol{\theta}) + \frac{\rho}{2}\|\mathrm{v} - M\boldsymbol{\theta}\|_2^2 \tag{6}$$

- Essentially, the problem becomes the iterative update of $\boldsymbol{\theta}$, v and u:

$$\min_{\boldsymbol{\theta}} \Omega_{las}(\boldsymbol{\theta}) + \mathcal{L}(\boldsymbol{\theta}) + \mathrm{u}^\top M\boldsymbol{\theta} + \frac{\rho}{2}\|\mathrm{v} - M\boldsymbol{\theta}\|_2^2 \tag{7}$$

$$\min_{\mathrm{v}} \Omega_{glas}(\mathrm{v}) + \mathrm{u}^\top \mathrm{v} + \frac{\rho}{2}\|\mathrm{v} - M\boldsymbol{\theta}\|_2^2 \tag{8}$$

$$\mathrm{u} = \mathrm{u} + \rho(\mathrm{v} - M\boldsymbol{\theta}) \tag{9}$$

Contribution

Regularizing with Clusters of Words

- LSI topic modeling (K topics)

$$\Omega_{LSI}(\boldsymbol{\theta}) = \sum_{k=1}^{K} \lambda \|\boldsymbol{\theta}_k\|_2 \tag{10}$$

- Community detection on Graph-of-Words (C communities):

$$\Omega_{gow}(\boldsymbol{\theta}) = \sum_{c=1}^{C} \lambda \|\boldsymbol{\theta}_c\|_2 \tag{11}$$

- Clustering in word embeddings (K clusters):

$$\Omega_{word2vec}(\boldsymbol{\theta}) = \sum_{k=1}^{K} \lambda \|\boldsymbol{\theta}_k\|_2 \tag{12}$$

Group lasso pros & cons

Advantages

- powerful regularization method
- in general fast
- sparsity

Drawbacks

- these groupings are either not available or hard to be extracted
- no ground truth groups of words exist to validate their quality
- group lasso may fail to create sparse models

Orthogonal Matching Pursuit for TC

OMP & Group OMP

Greedy feature selection algorithms used in signal processing. (Mallat and Zhang, 1993)

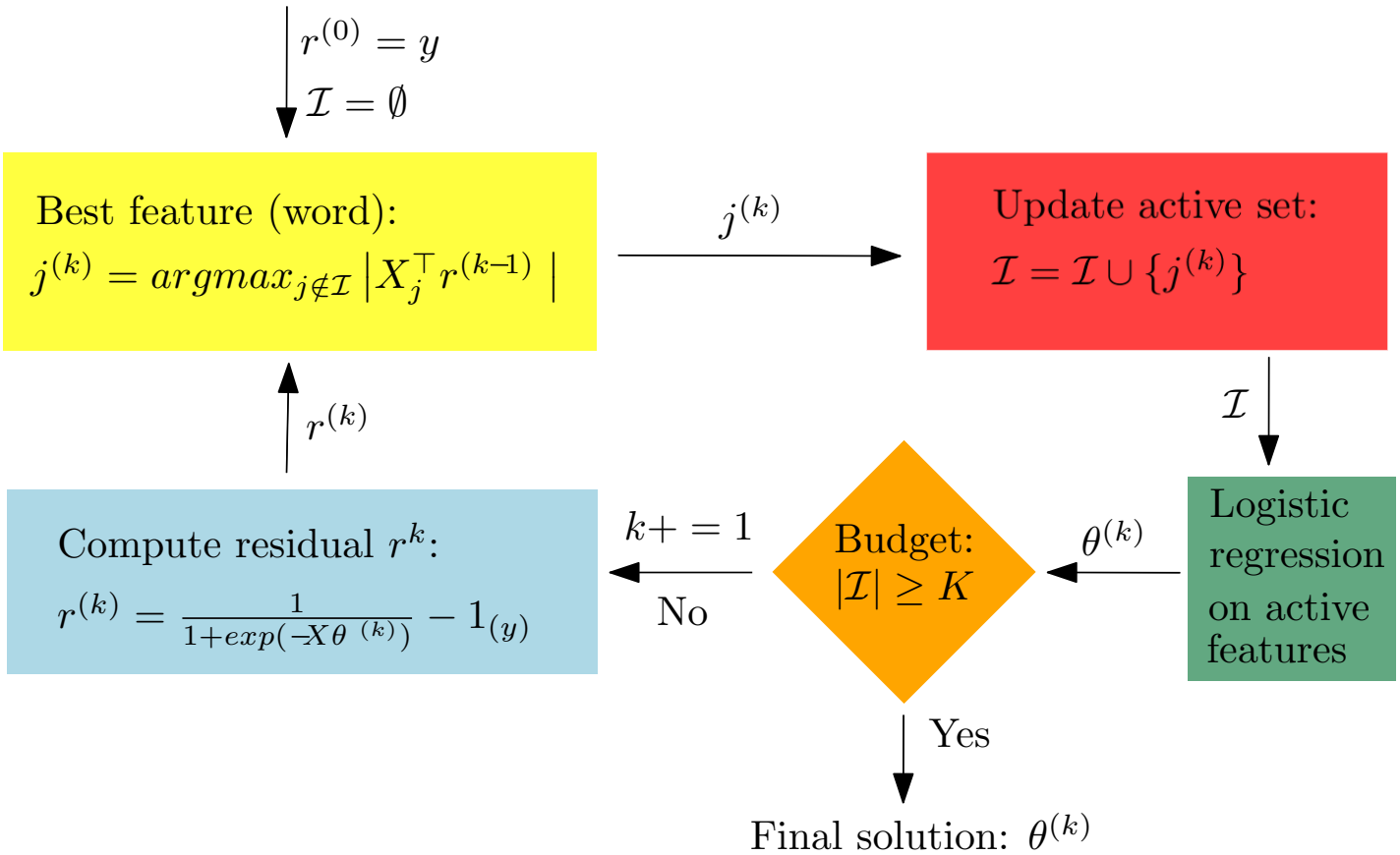
Based on:

- Group OMP for Variable Selection (Swirszcz et al., 2009)
- Logistic Regression (Lozano et al., 2011)

Contributions:

- use OMP as regularizer
- introduce overlapping Group OMP

Pipeline



Text classification with the OMP regularizer.

Datasets

Topic categorization

- 20 NG: four binary classification tasks

Sentiment analysis

- movie reviews (Pang and Lee, 2004)
- floor speeches by U.S. Congressmen deciding "yea"/"nay" votes on the bill under discussion (Thomas et al., 2006)
- Amazon product reviews (Blitzer et al., 2007)

	Dataset	Train	Dev	Test	Voc	#Sents
20NG	science	949	238	790	25787	16411
	sports	957	240	796	21938	14997
	religion	863	216	717	18822	18853
	comp.	934	234	777	16282	10772
Sentiment	vote	1175	257	860	19813	43563
	movie	1600	200	200	43800	49433
	books	1440	360	200	21545	13806
	dvd	1440	360	200	21086	13794
	electr.	1440	360	200	10961	10227
	kitch.	1440	360	200	9248	8998

Descriptive statistics of the datasets.

Results

	dataset	no reg	lasso	ridge	elastic	OMP	group lasso					GOMP
							LDA	LSI	sen	GoW	w2v	
20NG	science	0.946	0.916	0.954	0.954	0.964*	**0.968**	**0.968***	0.942	0.967*	**0.968***	0.953*
	sports	0.908	0.907	0.925	0.920	0.949*	0.959	0.964*	**0.966**	0.959*	0.946*	0.951*
	religion	0.894	0.876	0.895	0.890	0.902*	0.918	0.907*	**0.934**	0.911*	0.916*	0.902*
	computer	0.846	0.843	0.869	0.856	0.876*	0.891	0.885*	0.904	0.885*	**0.911***	0.902*
Sentiment	vote	0.606	0.643	0.616	0.622	0.684*	0.658	0.653	0.656	0.640	0.651	**0.687***
	movie	0.865	0.860	0.870	0.875	0.860*	**0.900**	0.895	0.895	0.895	0.890	0.850
	books	0.750	0.770	0.760	0.780	0.800	0.790	0.795	0.785	0.790	0.800	**0.805***
	dvd	0.765	0.735	0.770	0.760	0.785	0.800	0.805*	0.785	0.795*	0.795*	**0.820***
	electr.	0.790	0.800	0.800	0.825	**0.830**	0.800	0.815	0.805	0.820	0.815	0.800
	kitch.	0.760	0.800	0.775	0.800	0.825	0.845	**0.860***	0.855	0.840	0.855*	0.830

Accuracy on the test sets. Bold font marks the best performance for a dataset, while * indicates statistical significance at $p < 0.05$ using micro sign test against lasso. For GOMP, we use w2v clusters and add all unigram features as individual groups.

Sparsity

	dataset	no reg	lasso	ridge	elastic	OMP	group lasso LDA	group lasso LSI	group lasso sen	group lasso GoW	group lasso w2v	GOMP
20NG	science	100	**1**	100	63	2.7	19	20	86	19	21	5.8
	sports	100	**1**	100	5	1.8	60	11	6.4	55	44	7.7
	religion	100	**1.1**	100	3	1.5	94	31	99	10	85	1.5
	computer	100	1.6	100	7	**0.6**	40	35	77	38	18	4.9
Sentiment	vote	100	**0.1**	100	8	5	15	16	13	97	13	1.5
	movie	100	1.3	100	59	**0.9**	72	81	55	90	62	2.3
	books	100	**3.3**	100	14	4.6	41	74	72	90	99	8.3
	dvd	100	**2**	100	28	2.8	64	8	8	58	64	9
	electr.	100	**4**	100	6	6.3	10	8	43	8	9	12
	kitch.	100	4.5	100	79	**4.3**	73	44	27	75	46	6.5

Fraction (in %) of non-zero feature weights in each model for each dataset.
Bold for best, blue for best group.

Analysis

	dataset	GoW	word2vec
20NG	science	79	691
	sports	137	630
	religion	35	639
	computer	95	594

Number of groups.

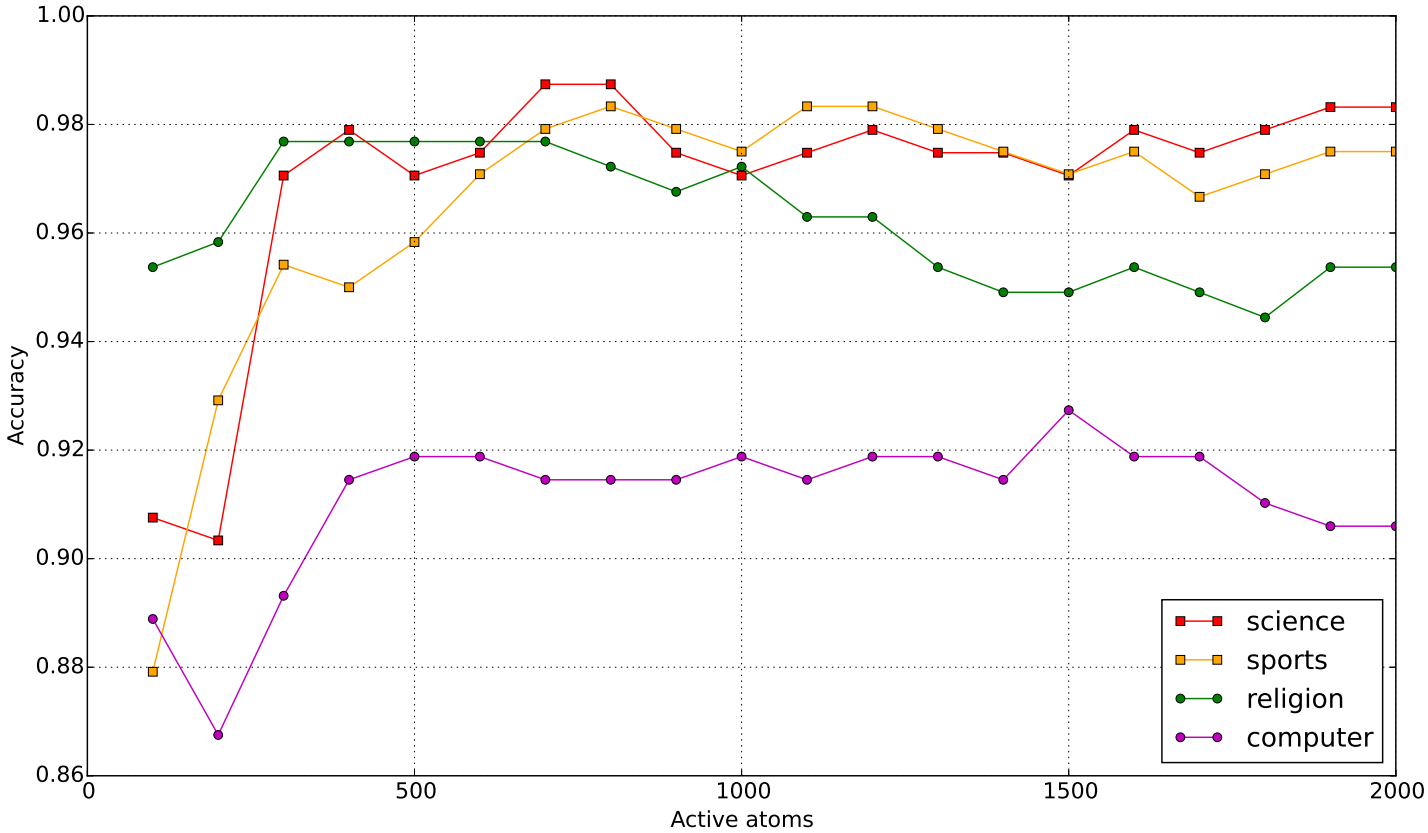
lasso	space: orbit, space, contribute, funding, landing hockey: nhl, hockey, playoffs, cup, wings
OMP	space: space, orbit, moon, planets, scientifically hockey: nhl, hockey, playoff, wings, cup

Features with the largest weights.

	dataset	lasso	ridge	elastic	group lasso					OMP
					LDA	LSI	sentence	GoW	word2vec	
20NG	science	10	**1.6**	**1.6**	15	11	76	12	19	78
	sports	12	**3**	**3**	7	20	67	5	9	21
	religion	12	**3**	7	10	4	248	6	20	38
	computer	7	1.4	**0.8**	8	6	43	5	10	4

Time (in seconds) for learning with best hyperparameters.

Selecting the number of atoms



Accuracy in dev set vs number of active atoms.

Summary

Pros

- OMP requires no prior
- GOMP beats group lasso in some cases (but always in sparsity)
- Fast with relatively small number of dimensions

Cons

- Greedy algorithm $\rightarrow$ GOMP gets slow (adding single terms)
- Groups need to be good

Contributions

- Clusters of words can enhance structured regularization
- Introduce overlapping GOMP and compare it with group lasso
- Creating super-sparse models

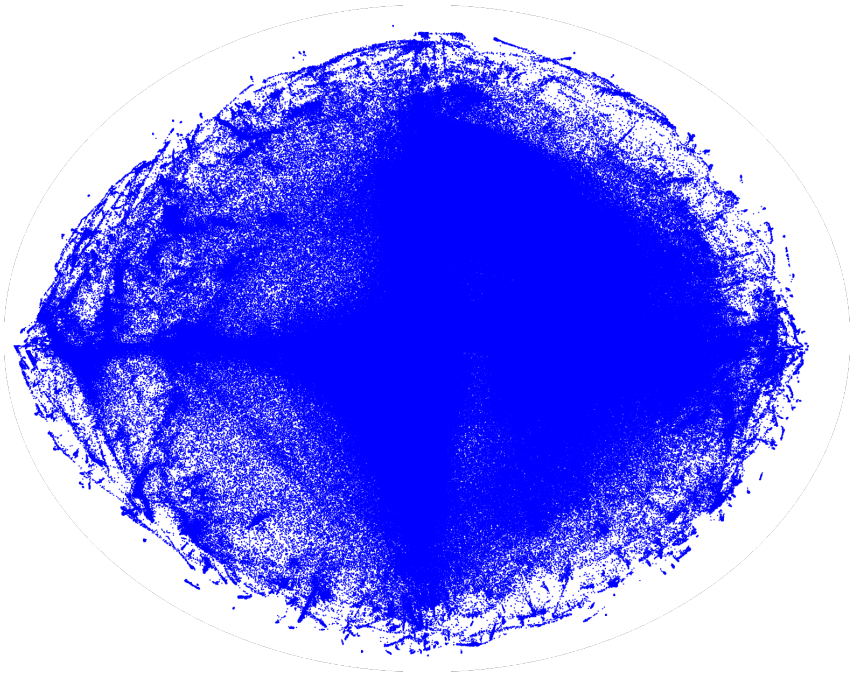
Can we use these linguistic structures (groups) for more?

Yes for distances based on word embeddings!

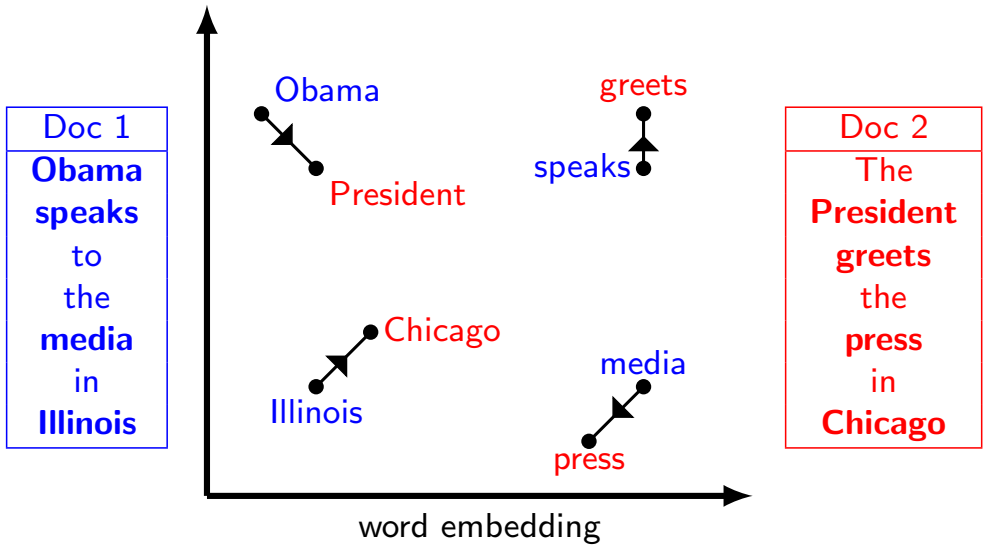
Outline

Distances in word embeddings

- Word embeddings to compute similarity between documents
- Cosine or euclidean distance
- Centroids lose a lot of info



TSNE 2d on Glove.



Word Mover's Distance.

- WMD and Supervised WMD (Kusner et al., 2015);(Huang et al., 2016)
- Optimal transport (Villani, 2008)
- SOTA distance-based classification

Enhancing tricks for WMD

Limitations

- Exact WMD scales at $\mathcal{O}(n^3)$
  $\hookrightarrow$Relaxed WMD
- Not all words are important
- Outlier words
- Labels not exploited

We focus on:

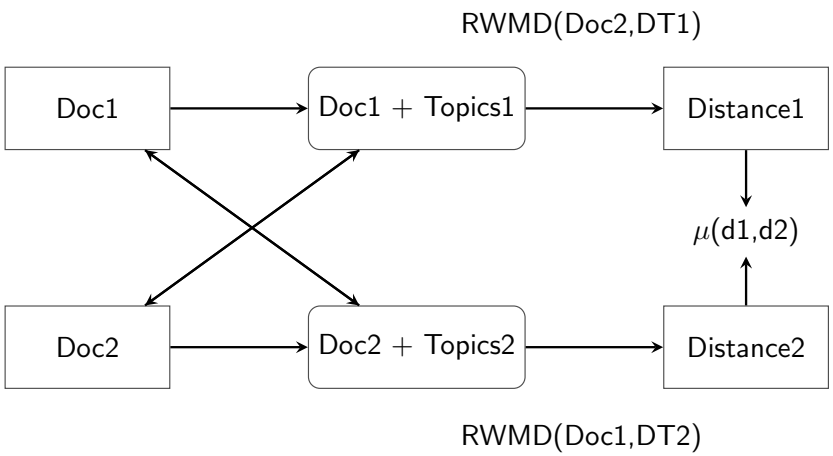
- Stopword removal
- Cross document-topic comparison by clustering word embeddings
- Convex metric learning

Contributions

Stopword removal

- More than 10 stopword lists
- SMART (Salton and Buckley, 1971)
- Gensim & spacy (Stone et al., 2010)

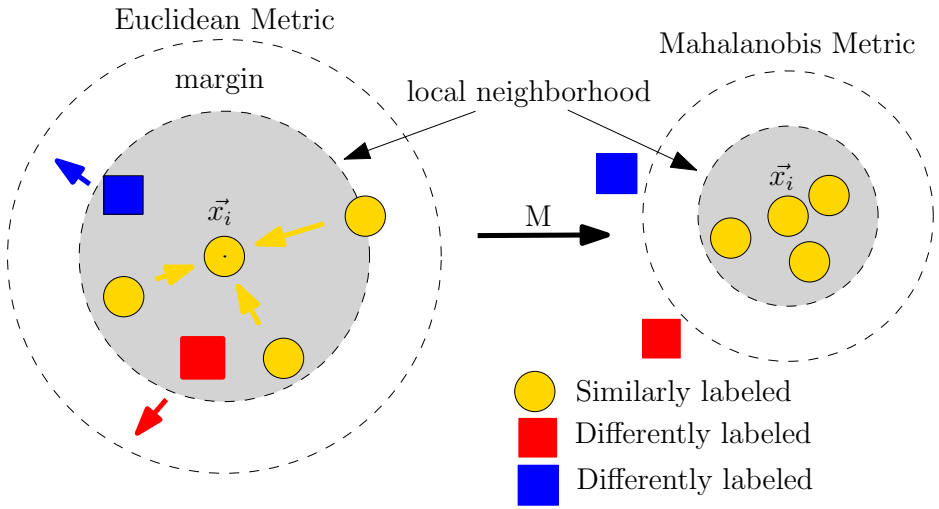
Cross document-topic comparison



DT1 = Doc1 words + Topics1.

Convex metric learning

- Maximally Collapsing Metric Learning (Globerson and Roweis, 2006)
- Large Margin Nearest Neighbor (Weinberger and Saul, 2009)



The LMNN architecture.

Datasets & setup

Datasets

(1) BBCSPORT: sports articles between 2004-2005

(2) TWITTER: sentiment tweets

(3) RECIPE: recipes by region

(4) OHSUMED: cardiovascular medical abstracts

(5) CLASSIC: academic papers by publisher

(6) REUTERS: news topics

(7) AMAZON: product reviews by sentiment

(8) 20NEWS: news articles in 20 categories

Dataset	n	Voc	Unique Words(avg)	y
BBCSPORT	517	13243	117	5
TWITTER	2176	6344	9.9	3
RECIPE	3059	5708	48.5	15
OHSUMED	3999	31789	59.2	10
CLASSIC	4965	24277	38.6	4
REUTERS	5485	22425	37.1	8
AMAZON	5600	42063	45.0	4
20NG	11293	29671	72	20

Datasets in our TC experiments.

Setup

- Pretrained w2v (Mikolov et al., 2013)
- Stopwords by (Stone et al., 2010) (used in Gensim & spacy)
- In k-means we set $k = 500$

Results

		BBCSPORT	TWITTER	RECIPE	OHSUMED	CLASSIC	REUTERS
Unsupervised	LSI	4.30 ± 0.60	31.70 ± 0.70	45.40 ± 0.50	44.20	6.70 ± 0.40	6.30
	WMD	4.60 ± 0.70	28.70 ± 0.60	42.60 ± 0.30	44.50	2.88 ± 0.10	3.50
	Stopword RWMD	4.27 ± 1.19	27.51 ± 1.00	43.98 ± 1.40	44.27	3.25 ± 0.50	5.25
	All, 5nn	6.00 ± 1.34	29.23 ± 1.09	42.52 ± 1.18	46.73	3.18 ± 0.44	6.26
	All, 5nn, Mean	4.00 ± 1.55	28.58 ± 2.29	42.53 ± 0.67	43.90	3.08 ± 0.62	5.76
	k-means, 5nn	5.91 ± 2.65	28.56 ± 1.20	42.23 ± 1.15	46.50	2.98 ± 0.66	4.71
	k-means, 5nn, Mean	3.82 ± 1.72	28.50 ± 1.51	41.95 ± 1.04	44.05	3.08 ± 0.51	4.57
Supervised	S-WMD (NCA)	2.10 ± 0.50	27.50 ± 0.50	39.20 ± 0.30	**34.30**	3.20 ± 0.20	3.20
	LMNN	**1.73 ± 0.67**	28.86 ± 2.22	40.88 ± 1.88	39.59	**2.76 ± 0.30**	4.02
	MCML	2.45 ± 1.27	**27.15 ± 1.36**	**38.93 ± 1.24**	42.38	3.56 ± 0.49	**2.92**

Comparison in *k*nn test error(%) to LSI, WMD and S-WMD. Blue shows best results in unsupervised methods and bold indicates best result for a dataset.

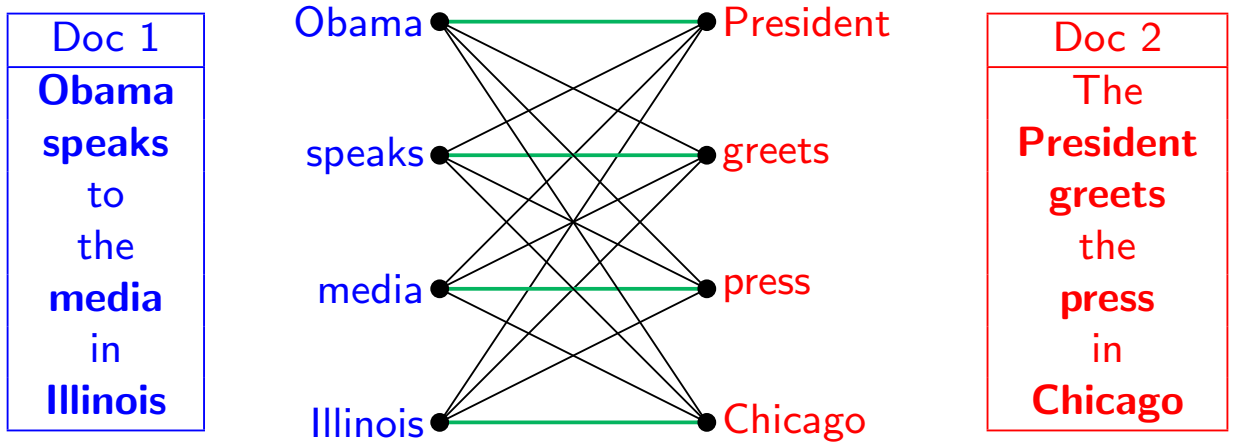
Summary

Contributions

- many stopword lists discovered
- adding neighbour words via clustering helps
- convex metric learning can boost accuracy

Can we do more?

Learning document representations via distances in word embeddings.



Bipartite Graph Matching for two documents.

Sets in word embeddings

What is a set?

Complex data sets composed of simpler objects.
$\hookrightarrow$ NLP: documents as sets of word embeddings

Standard machine learning algorithms:

- fixed dimensional data instances
- data representations and learning are independent

Text classification as set classification

(1) distance/similarity measure or kernel that finds a correspondence between each pair of sets

(2) use instance-based machine learning algorithm ($k$nn or SVM)

$\hookrightarrow$ high computational and memory complexity (all to all comparison)

Main approaches

Neural networks:

- PointNet (Qi et al., 2017) and DeepSets (Zaheer et al., 2017) transform the vectors of the sets into new representations, then apply permutation-invariant functions
- unordered sets → ordered sequences → RNN (Vinyals et al., 2015)

Kernels:

- estimate a probability distribution on each set of vectors, then derive their similarity using distribution-based comparison measures such as Fisher kernels (Jaakkola and Haussler, 1999)
- map the vectors to multi-resolution histograms, then compare them with a weighted histogram intersection measure to find an approximate correspondence (Grauman and Darrell, 2007)

Distance metric learning:

- learning a distance function over objects
- text → Supervised Word Mover's Distance (Huang et al., 2016)

Pros & cons

Advantages

- easily extended (NNs)
- fast and scalable with GPUs (NNs)
- very effective in several tasks (NNs & kernels)

Limitations

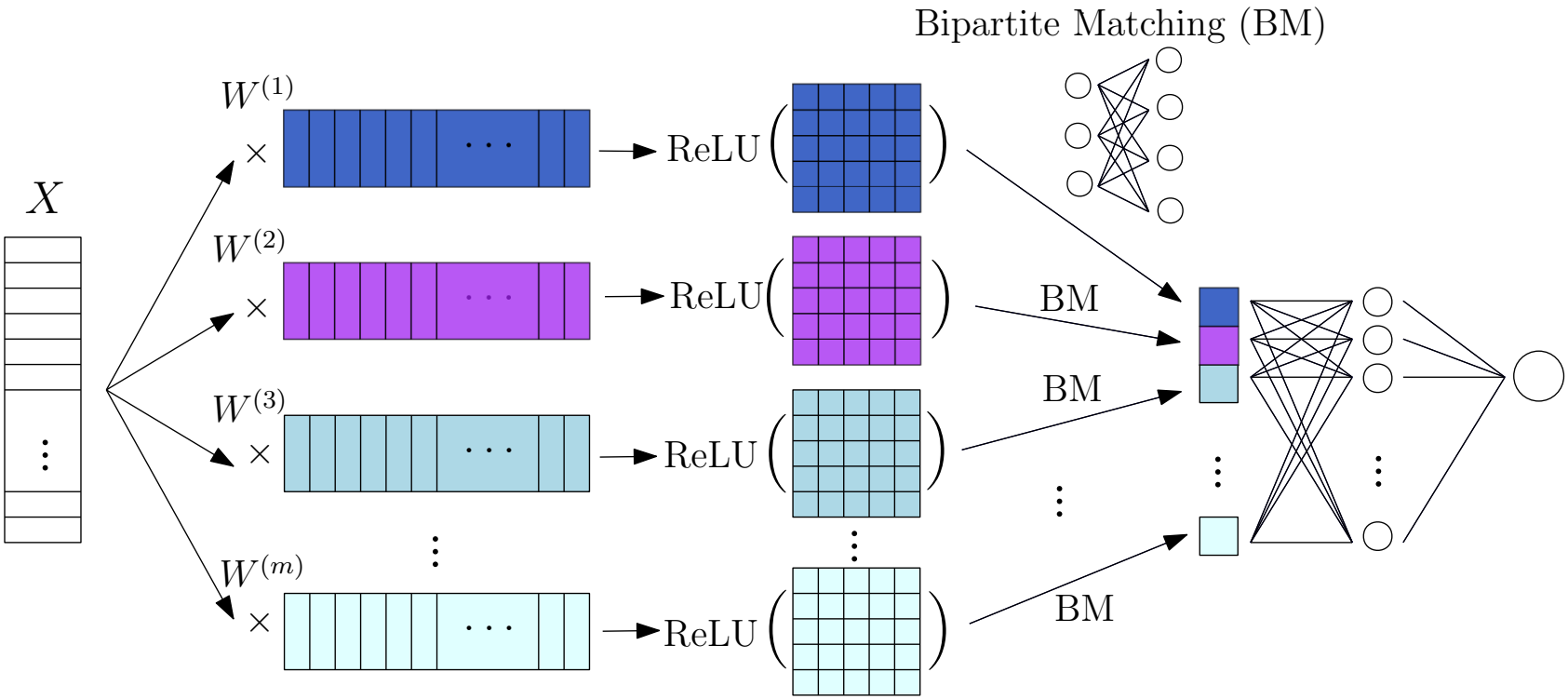
- simple permutation invariant functions
- limits the expressive power of these architectures
- kernels $\Rightarrow$ high computational complexity
- data representation and learning are independent from each other

Contributions

- **RepSet** a novel architecture for performing machine learning on sets which, in contrast to traditional approaches, is capable of adapting data representation to the task at hand
- **ApproxRepSet**, a simplified architecture which can be interpreted as an approximation of the proposed model, able to handle very large datasets
- evaluation of the proposed architecture on several benchmark datasets in text classification

Learning sets via distances in word embeddings

RepSet: Neural Networks for Learning Set Representations



Each element of the input set is compared with the elements of all "hidden sets", and the emerging matrices serve as the input to bipartite matching. The values of the BM problems correspond to the representation of the input set.

Relaxed variant (ApproxRepSet)

Given an input set of vectors, $X = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{k_1}\}$ and a hidden set $Y_i = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{k_2}\}$, first we identify which of the two sets has the highest cardinality. If $|X| \geq |Y_i|$, we solve the following linear program:

$$\begin{aligned} &\max \sum_{i=1}^{k_1} \sum_{j=1}^{k_2} x_{ij} f(\mathbf{v}_i, \mathbf{u}_j) \text{ subject to:} \\ &\sum_{i=1}^{k_1} x_{ij} \leq 1 \quad \forall j \in \{1, \ldots, k_2\} \\ &x_{ij} \geq 0 \quad \forall i \in \{1, \ldots, k_1\}, \forall j \in \{1, \ldots, k_2\} \end{aligned} \tag{13}$$

- multiple elements of Y_i (the bigger set) can be matched with the same element of X
- the optimal solution matches an element of Y with one of X if their inner product is positive and is the highest among the inner products between all the pairs

Results

	BBCSPORT	TWITTER	RECIPE	OHSUMED	CLASSIC	REUTERS	AMAZON	20NG
LSI	4.30 ± 0.60	31.70 ± 0.70	45.40 ± 0.50	44.20	6.70 ± 0.40	6.30	9.30 ± 0.40	28.90
WMD	4.60 ± 0.70	28.70 ± 0.60	42.60 ± 0.30	44.50	**2.88** ± 0.10	3.50	7.40 ± 0.30	26.80
S-WMD	2.10 ± 0.50	27.50 ± 0.50	39.20 ± 0.30	34.30	3.20 ± 0.20	3.20	5.80 ± 0.10	26.80
DeepSets	25.45 ± 20.1	29.66 ± 1.62	70.25	71.53	5.95 ± 1.50	10.00	8.58 ± 0.67	38.88
NN-mean	10.09 ± 2.62	31.56 ± 1.53	64.30 ± 7.30	45.37	5.35 ± 0.75	11.37	13.66 ± 3.16	38.40
NN-max	2.18 ± 1.75	30.27 ± 1.26	43.47 ± 1.05	35.88	4.21 ± 0.11	4.33	7.55 ± 0.63	32.15
NN-attention	4.72 ± 0.97	29.09 ± 0.62	43.18 ± 1.22	**31.36**	4.42 ± 0.73	3.97	6.92 ± 0.51	28.73
RepSet	**2.00** ± 0.89	**25.42** ± 1.10	**38.57** ± 0.83	**33.88**	3.38 ± 0.50	3.15	**5.29** ± 0.28	**22.98**
Approx	4.27 ± 1.73	27.40 ± 1.95	40.94 ± 0.40	35.94	3.76 ± 0.45	**2.83**	5.69 ± 0.40	23.82

Classification test error of the proposed architecture and the baselines.

BBCSPORT	Points of sets	Centroids
1	cup, club, united, striker, arsenal	Modric
2	scrum, nations, scotland, ireland, france	rugby
3	winner, olympic, tennis, court, Olympic_gold_medalist	Olympic_Medalist
4	captain, player, striker, ball, game	skipper
5	wickets, series, cricket, bat, side	batsmen

Terms of the employed pre-trained model that are most similar to the points and centroids of the elements of 5 hidden sets.

Summary

Contributions:

- RepSet, neural networks for learning set representations
  - exhibits powerful permutation invariance properties
  - highly interpretable
  - easily extended to deeper architectures
- introduced a relaxed version (ApproxRepSet)
  - involves fast matrix operations and scales to large datasets
- effective on text categorization
- PyTorch implementation (fast in GPU)

Future work:

- replacing matching with optimal transport?
- other tasks?
  - graph mining: graphs as sets of node embeddings
  - computer vision: images as sets of local features

Outline

Summary

> *"Caminante, no hay camino, se hace camino al andar.*
> *Wanderer, there is no path, the path is made by walking."*
>
> — Antonio Machado

In this thesis, I have:

- Explored in details the fields of NLP and ML
- Worked in methods to exploit context and discover latent concepts
- Tested in real world tasks

Ph.D. thesis contributions

- Created new graph-based representations to model text in more meaningful structures and derived innovative metrics out of them
- Extracted novel linguistic groups and developed new methods for structured regularization
- Boosted existing document comparison techniques and designed a new set representation learning model via distances
- Applied them to text classification

Future work

- **TW-IDW**: Graph-of-Documents instead of a Bag-of-Documents in order for instance to compute an alternative to IDF
- **Graph neural networks**: involves neural message passing
- **Linguistic structured attention**: a group lasso attention mechanism for deep learning architectures
- **Adversarial structured regularization**
- **Gradient-based learning for WMD**

Thank you!

References I


Baeza-Yates, Ricardo A. and Berthier Ribeiro-Neto (1999). *Modern Information Retrieval*. MIR. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.


Bakin, Sergey (May 1999). "Adaptive regression and model selection in data mining problems". Ph.D. Canberra, Australia: The Australian National University.


Blitzer, John, Mark Dredze, and Fernando Pereira (2007). "Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification". In: *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. ACL '07. ACL, pp. 440–447.


Borgwardt, Karsten M., Nicol N. Schraudolph, and S.v.n. Vishwanathan (2007). "Fast Computation of Graph Kernels". In: *Advances in Neural Information Processing Systems 19*. Ed. by B. Schölkopf, J. C. Platt, and T. Hoffman. NIPS. MIT Press, pp. 1449–1456. URL: http://papers.nips.cc/paper/2973-fast-computation-of-graph-kernels.pdf.


Deshpande, Mukund et al. (2005). "Frequent Substructure-Based Approaches for Classifying Chemical Compounds". In: *IEEE Trans. on Knowl. and Data Eng.* TKDE 17.8, pp. 1036–1050.


Firth, J. R. (1957). "A synopsis of linguistic theory 1930-55.". In: 1952-59, pp. 1–32.


Gilmer, Justin et al. (2017). "Neural message passing for quantum chemistry". In: *arXiv preprint arXiv:1704.01212*. ICML.


Globerson, Amir and Sam T Roweis (2006). "Metric learning by collapsing classes". In: *Advances in neural information processing systems*, pp. 451–458.


Grauman, Kristen and Trevor Darrell (2007). "The pyramid match kernel: Efficient learning with sets of features". In: *Journal of Machine Learning Research*. JMLR 8.Apr, pp. 725–760.


Harris, Zellig S (1954). "Distributional structure". In: *Word* 10.2-3, pp. 146–162.

References II

Hassan, Samer, Rada Mihalcea, and Carmen Banea (2007). "Random-Walk Term Weighting for Improved Text Classification.". In: *ICSC*. ICSC, pp. 242–249.

Hoerl, Arthur E. and Robert W. Kennard (1970). "Ridge regression: Biased estimation for nonorthogonal problems". In: *Technometrics* 12.1, pp. 55–67.

Huang, Gao et al. (2016). "Supervised Word Mover's Distance". In: *Advances in Neural Information Processing Systems*, pp. 4862–4870.

Jaakkola, Tommi and David Haussler (1999). "Exploiting generative models in discriminative classifiers". In: *Advances in Neural Information Processing Systems*. NIPS, pp. 487–493.

Johnson, Rie and Tong Zhang (2015). "Effective Use of Word Order for Text Categorization with Convolutional Neural Networks". In: *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*. NAACL, pp. 103–112.

— (2017). "Deep pyramid convolutional neural networks for text categorization". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vol. 1, pp. 562–570.

Joulin, Armand et al. (2017). "Bag of tricks for efficient text classification". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. EACL, pp. 427–431.

Kim, Yoon (2014). "Convolutional neural networks for sentence classification". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. EMNLP.

Kusner, Matt et al. (2015). "From word embeddings to document distances". In: *International Conference on Machine Learning*. ICML, pp. 957–966.

References III

Lozano, Aurelie C., Grzegorz Swirszcz, and Naoki Abe (2011). "Group Orthogonal Matching Pursuit for Logistic Regression". In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011*. AISTATS '11, pp. 452–460. URL: http://www.jmlr.org/proceedings/papers/v15/lozano11a/lozano11a.pdf.

Mallat, Stéphane G and Zhifeng Zhang (1993). "Matching pursuits with time-frequency dictionaries". In: *IEEE Transactions on signal processing*. IEEE Transactions on signal processing '93 41.12, pp. 3397–3415.

Malliaros, Fragkiskos D. and Konstantinos Skianis (2015a). "Graph-based term weighting for text categorization". In: *Someris, ASONAM*. ACM, pp. 1473–1479.

— (2015b). "Graph-Based Term Weighting for Text Categorization". In: *Proceedings of ASONAM*. ASONAM, pp. 1473–1479.

Mihalcea, Rada and Paul Tarau (2004). "TextRank: Bringing Order into Text". In: *EMNLP*. EMNLP, pp. 404–411.

Mikolov, T. et al. (2013). "Efficient estimation of word representations in vector space.". In: *ICLR Workshop*.

Nikolentzos, Giannis et al. (2017). "Shortest-Path Graph Kernels for Document Similarity". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. EMNLP, pp. 1890–1900.

Nikolentzos, Giannis et al. (2018). "Kernel graph convolutional neural networks". In: *International Conference on Artificial Neural Networks*. Springer, Cham, pp. 22–32.

Outsios, Stamatis et al. (2018). "Word Embeddings from Large-Scale Greek Web content". In: *Spoken Language Technology (SLT)*.

Pang, Bo and Lilian Lee (2004). "A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts". In: *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. ACL '04. ACL, pp. 271–278.

Qi, Charles R et al. (2017). "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. CVPR, pp. 652–660.

References IV


Rousseau, François and Michalis Vazirgiannis (2013). "Graph-of-word and TW-IDF: new approach to ad hoc IR". In: *CIKM*. CIKM, pp. 59–68.


Salton, Gerard and C Buckley (1971). *The SMART information retrieval system*.


Shanavas, Niloofer et al. (2016). "Centrality-Based Approach for Supervised Term Weighting". In: *Data Mining Workshops (ICDMW), 2016 IEEE 16th International Conference on*. ICDM. IEEE, pp. 1261–1268.


Siglidis, Giannis et al. (2018). "GraKeL: A Graph Kernel Library in Python". In: *arXiv preprint arXiv:1806.02193*.


Skianis, Konstantinos, François Rousseau, and Michalis Vazirgiannis (2016a). "Regularizing Text Categorization with Clusters of Words". In: *EMNLP*, pp. 1827–1837.


Skianis, Konstantinos et al. (2016b). "SPREADVIZ: Analytics and Visualization of Spreading Processes in Social Networks". In: *Demo, ICDM*. IEEE, pp. 1324–1327.


Skianis, Konstantinos, Fragkiskos D. Malliaros, and Michalis Vazirgiannis (2018a). "Fusing Document, Collection and Label Graph-based Representations with Word Embeddings for Text Classification". In: *TextGraphs, NAACL*, pp. 49–58.


Skianis, Konstantinos, Nikolaos Tziortziotis, and Michalis Vazirgiannis (2018b). "Orthogonal Matching Pursuit for Text Classification". In: *W-NUT, EMNLP*.


Skianis, Konstantinos et al. (2019a). "Boosting Tricks for Word Mover's Distance". Manuscript.


Skianis, Konstantinos et al. (2019b). "Group Lasso for Linguistic Structured Attention". Manuscript.


Skianis, Konstantinos et al. (2019c). "Rep the Set: Neural Networks for Learning Set Representations". Manuscript.

References V


Sparck Jones, Karen (1972). "A statistical interpretation of term specificity and its application in retrieval". In: *Journal of documentation* 28.1, pp. 11–21.


Stone, Benjamin, Simon Dennis, and Peter J Kwantes (2010). "Comparing methods for document similarity analysis". In: *TopiCS, DOI* 10.


Swirszcz, Grzegorz, Naoki Abe, and Aurelie C Lozano (2009). "Grouped Orthogonal Matching Pursuit for Variable Selection and Prediction". In: *Advances in Neural Information Processing Systems 22*. Ed. by Y. Bengio et al. NIPS '09. Curran Associates, Inc., pp. 1150–1158. URL: http://papers.nips.cc/paper/3878-grouped-orthogonal-matching-pursuit-for-variable-selection-and-prediction.pdf.


Thomas, Matt, Bo Pang, and Lillian Lee (2006). "Get Out The Vote: Determining Support Or Opposition From Congressional Floor-Debate Transcripts". In: *Proceedings of EMNLP*. EMNLP '06, pp. 327–335.


Tibshirani, Robert (1996). "Regression shrinkage and selection via the lasso". In: *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288.


Tixier, Antoine J-P, Konstantinos Skianis, and Michalis Vazirgiannis (2016). "GoWvis: a web application for Graph-of-Words-based text visualization and summarization". In: *ACL 2016*. ACL, p. 151.


Villani, Cédric (2008). *Optimal transport: old and new*. Vol. 338. Springer Science & Business Media.


Vinyals, Oriol, Samy Bengio, and Manjunath Kudlur (2015). "Order matters: Sequence to sequence for sets". In: *International Conference on Learning Representations*. ICLR.


Wang, Rui, Wei Liu, and Chris McDonald (2015). "Corpus-independent Generic Keyphrase Extraction Using Word Embedding Vectors". In:


Weinberger, Kilian Q and Lawrence K Saul (2009). "Distance metric learning for large margin nearest neighbor classification". In: *Journal of Machine Learning Research* 10.Feb, pp. 207–244.

References VI

Yao, Liang, Chengsheng Mao, and Yuan Luo (2019). "Graph Convolutional Networks for Text Classification". In: *Association for the Advancement of Artificial Intelligence*.

Yogatama, Dani and Noah A. Smith (2014). "Making the Most of Bag of Words: Sentence Regularization with Alternating Direction Method of Multipliers". In: *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*. ICML'14. Beijing, China: JMLR.org, pp. I–656–I–664. URL: http://dl.acm.org/citation.cfm?id=3044805.3044880.

Yuan, Ming and Yi Lin (2006). "Model selection and estimation in regression with grouped variables". In: *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 68.1, pp. 49–67.

Zaheer, Manzil et al. (2017). "Deep sets". In: *Advances in Neural Information Processing Systems*. NIPS, pp. 3391–3401.

Regularizer examples

$= 0$	piscataway combination jil@donuts0.uucp jamie reading/seeing chambliss left-handedness abilities lubin acad sci obesity page erythromycin bottom
$\neq 0$	and space the launch health for use that medical you space cancer and nasa hiv health shuttle for tobacco that cancer that research center space hiv aids are use theory keyboard data telescope available are from system information space ftp

Examples with LSI regularizer.

$= 0$	village town edc fashionable trendy trendy fashionable points guard guarding crown title champion champions
$\neq 0$	numbness tingling dizziness fevers laryngitis bronchitis undergo undergoing undergoes undergone healed mankind humanity civilization planet nasa kunin lang tao kay kong

Examples with word2vec regularizer.

$= 0$	islands inta spain galapagos canary originated anodise advertises jewelry mercedes benzes diamond trendy octave chanute lillienthal
$\neq 0$	vibrational broiled relieving succumb spacewalks dna nf-psychiatry itself commented usenet golded insects alternate self-consistent retrospect

Examples with graph-of-words regularizer.

GOMP algorithm

Algorithm 1 Logistic Overlapping GOMP

Input: $X = [x_1, \ldots, x_N]^\top \in \mathbb{R}^{N \times d}$, $y \in \{-1, 1\}^N$, $\{G_1, \ldots, G_J\}$ (group structure), K (budget), ϵ (precision), λ (regularization factor).

Output: $\mathcal{I} = \emptyset$, $r^{(0)} = y$, $k = 1$

while $|\mathcal{I}| \leq K$ **do**
  $j^{(k)} = \text{argmax}_j \frac{1}{|G_j|} |X_{G_j}^\top r^{(k-1)}|_2^2$
  if $|X_{G_{j^{(k)}}}^T r^{(k-1)}|_2^2 \leq \epsilon$ **then**
    break
  end if
  $\mathcal{I} = \mathcal{I} \cup \{G_{j^{(k)}}\}$
  for $i = 1$ **to** J **do**
    $G_i = G_i \setminus G_{j^{(k)}}$
  end for
  $\theta^{(k)} = \text{argmin}_\theta \sum_{i=1}^N \mathcal{L}(x_i, \theta, y_i) + \lambda \|\theta\|_2^2$ s.t. $supp(\theta) \subseteq \mathcal{I}$
  $r^{(k)} = \frac{1}{1+\exp\{-X\theta^{(k)}\}} - \mathbb{1}_{\{y\}}$
  $k \mathrel{+}= 1$
end while
return $\theta^{(k)}, \mathcal{I}$

Overlapping GOMP
Difference with GOMP

- Overlapping GOMP extends the standard GOMP in the case where the groups of indices are overlapping, i. e. $G_i \cap G_j \neq \emptyset$ for $i \neq j$
- The main difference with GOMP is that each time a group becomes active, we remove its indices from each inactive group: $G_i = G_i \setminus G_{j(k)}, \forall i \in \{1, \ldots, J\}$
- In this way, the theoretical properties of GOMP hold also in the case of the overlapping GOMP

Computing the gradients (1/2)

The gradient of the loss function with respect to the j^{th} row of the weight matrix of the penultimate layer is:

$$\frac{\partial L}{\partial \mathbf{W}_j^{(c)}} = \mathbf{p}_j - \mathbf{y}_j \tag{14}$$

We next define the following differentiable function:

$$g(\mathbf{D}^{(k)*}, \mathbf{W}^{(k)}) = \sum_{i=1}^{k_1} \sum_{j=1}^{k_2} \mathbf{D}_{ij}^{(k)*} f(\mathbf{v}_i, \mathbf{u}_j^{(k)}) \tag{15}$$

where $\mathbf{D}^{(k)*}$ is the matrix that contains the optimal values of the variables of the BM problem for the corresponding hidden set Y_k.

$$\frac{\partial}{\partial \mathbf{W}^{(k)}} g(\mathbf{D}^{(k)*}, \mathbf{W}^{(k)}) = \frac{\partial}{\partial \mathbf{W}^{(k)}} tr(\mathbf{D}^{(k)*\top} \mathbf{G}^{(k)}) \tag{16}$$

Computing the gradients (2/2)

We have:

- $\mathbf{X}$, a matrix whose rows correspond to the elements of set X.
- $\mathbf{G}^{(k)} = \text{ReLU}(\mathbf{XW})$, weights that have been set to zero during the forward pass are stored as zero values in the optimal solution $\mathbf{D}^{(k)*}$.
- Indeed, no edge was created whenever the value of the dot product was negative. Then, none of these pairs have been used during the forward pass.

This yields :

$$\begin{aligned}\frac{\partial}{\partial \mathbf{W}^{(k)}} g(\mathbf{D}^{(k)*}, \mathbf{W}^{(k)}) =& \frac{\partial}{\partial \mathbf{W}^{(k)}} tr(\mathbf{D}^{(k)*\top} \mathbf{X} \mathbf{W}^{(k)}) \\ =& \mathbf{X}^\top \mathbf{D}^{(k)*}\end{aligned} \tag{17}$$

Which finally gives:

$$\frac{\partial L}{\partial \mathbf{W}^{(k)}} = (\mathbf{p}_k - \mathbf{y}_k) \cdot \mathbf{X}^\top \mathbf{D}^{(k)*} \tag{18}$$