Incremental Spectral Sparsification for Large-Scale Graph-Based Semi-Supervised Learning

Daniele Calandriello, Alessandro Lazaric, Michal Valko, Ioannis Koutis

Motivation

- In many applications ***unsupervised samples*** are abundant but labels are expensive (e.g., image classification)
- Graph-based SSL (e.g., HFS) exploits the graph structure to learn ***accurate predictions***
- ***Large graphs***, obtained by construction or by input, are very common (e.g., Facebook $n \approx 10^9$ users)
- HFS has complexity

$$\mathcal{O}(n^3) \text{ time} \quad \mathcal{O}(n^2) \text{ space}$$

⇒ **Can we reduce the computational complexity without compromising the accuracy?**

Transductive SSL

The graph

- Input set $\mathcal{X} = \{x_1, \ldots, x_n\}$ in $\mathbb{R}^d$
- Build the graph $\mathcal{G} = (\mathcal{X}, \mathcal{E})$ with $|\mathcal{E}| = m$
- The weights $a_{e_{i,j}}$ encodes the distance between nodes

The transductive SSL setting

- There exists a label y_i for each node x_i in $\mathcal{G}$
- $\mathcal{S}$ is the set of l labeled nodes
- $\mathcal{T}$ is the set of $u = n - l$ unlabeled nodes
- A SSL algorithm receives as input $\mathcal{X}$, $\mathcal{S}$ and their labels and returns a labeling function $\mathbf{f} : \mathcal{X} \to \mathbb{R}$
- Empirical error

$$\widehat{R}(\mathbf{f}) = \frac{1}{l} \sum_{i=1}^{l} (\mathbf{f}(x_i) - \mathbf{y}(x_i))^2$$

- Generalization error

$$R(\mathbf{f}) = \frac{1}{u} \sum_{i=1}^{u} (\mathbf{f}(x_i) - \mathbf{y}(x_i))^2$$

Assumptions

- The graph $\mathcal{G}$ is fixed (e.g., in spam classification the email corpus is fixed)
- The labeled set $\mathcal{S}$ is drawn at random over $\mathcal{X}$ (e.g., which mails the users classify as span or ham)

Harmonic Function Solution

Graph Structure

- Weighted adjacency matrix $A_\mathcal{G}$
- Degree matrix $D_\mathcal{G}$
- Laplacian matrix $L_\mathcal{G}$

HFS Solution

$$\boxed{\widehat{\mathbf{f}} = \arg\min_{\mathbf{f} \in \mathbb{R}^n} \tfrac{1}{l} (\mathbf{f} - \mathbf{y})^\mathsf{T} I_\mathcal{S} (\mathbf{f} - \mathbf{y}) + \gamma \mathbf{f}^\mathsf{T} L_\mathcal{G} \mathbf{f}}$$

$$\widehat{\mathbf{f}} = (\gamma l L_\mathcal{G} + I_\mathcal{S})^+ \mathbf{y}_\mathcal{S}$$

Algorithmic Stability

How much the solution is affected by the training set:

$$\begin{matrix} \mathcal{S} \\ \mathcal{S}' \end{matrix} = \begin{pmatrix} y_1, y_2, y_3, \ldots, y_{l-1} & y_l & 0 & 0, \ldots, 0 \\ y_1, y_2, y_3, \ldots, y_{l-1} & 0 & y_{l+1} & 0, \ldots, 0 \end{pmatrix} \to \begin{matrix} \mathbf{f} \\ \mathbf{f}' \end{matrix}$$

β-stable algorithm

$$\forall x \in \mathcal{X}: \quad |(\mathbf{f}(x) - \mathbf{y}(x))^2 - (\mathbf{f}'(x) - \mathbf{y}(x))^2| \leq \beta$$

Stable-HFS [4]

$$\boxed{\widehat{\mathbf{f}} = \arg\min_{\mathbf{f} \in \mathbb{R}^n} \tfrac{1}{l} (\mathbf{f} - \mathbf{y})^\mathsf{T} I_\mathcal{S} (\mathbf{f} - \mathbf{y}) + \gamma \mathbf{f}^\mathsf{T} L_\mathcal{G} \mathbf{f} + \tfrac{\mu}{l} \mathbf{f}^\mathsf{T} \mathbf{1}}$$

$$\mu = \frac{(\gamma l L_\mathcal{G} + I_\mathcal{S})^+ \mathbf{y}_\mathcal{S})^\mathsf{T} \mathbf{1}}{(\gamma l L_\mathcal{G} + I_\mathcal{S})^+ \mathbf{1})^\mathsf{T} \mathbf{1}}, \quad \widehat{\mathbf{f}} = (\gamma l L_\mathcal{G} + I_\mathcal{S})^+ (\mathbf{y}_\mathcal{S} - \mu \mathbf{1})$$

Proposition [1] If the labels $\widetilde{\mathbf{y}}_\mathcal{S}$ are centered, then Stable-HFS returns a function $\widehat{\mathbf{f}}$ such that w.p. $1 - \delta$ satisfies,

$$R(\widehat{\mathbf{f}}) \leq \widehat{R}(\widehat{\mathbf{f}}) + \beta + \left(2\beta + \frac{c^2(l+u)}{lu}\right) \sqrt{\frac{\pi(l,u) \log(1/\delta)}{2}}$$

with

$$\pi(l,u) = \frac{lu}{l + u - 0.5} \frac{2\max\{l,u\}}{2\max\{l,u\} - 1},$$

$$\beta \leq \frac{1.5 M \sqrt{l}}{(l \gamma \lambda_2(\mathcal{G}) - 1)^2} + \frac{4M}{l \gamma \lambda_2(\mathcal{G}) - 1}$$

References

[1] Cortes, C., Mohri, M., Pechyony, D., and Rastogi, A. Stability of transductive regression algorithms. In *ICML*, 2008.

[2] Koutis, Ioannis, Miller, Gary L., and Peng, Richard. A nearly-m log n time solver for SDD linear systems. In *FOCS*, 2011.

[3] Fergus, Rob, Weiss, Yair, and Torralba, Antonio. Semi-Supervised Learning in Gigantic Image Collections. In *NIPS*, 2009.

[4] Belkin, M., Matveeva, I., and Niyogi, P.. Regularization and Semi-Supervised Learning on Large Graphs. In *COLT*, 2004.

[5] Ji, Ming, Yang, Tianbao, Lin, Binbin, Jin, Rong, and Han, Jiawei. A Simple Algorithm for Semi-supervised Learning with Improved Generalization Error Bound. In *ICML*, 2012.

[6] Kelner, J. A. and Levin, A.. Spectral Sparsification in the Semi-streaming Setting. *Theory of Comput. Syst.*, 53(2):243–262, 2013.

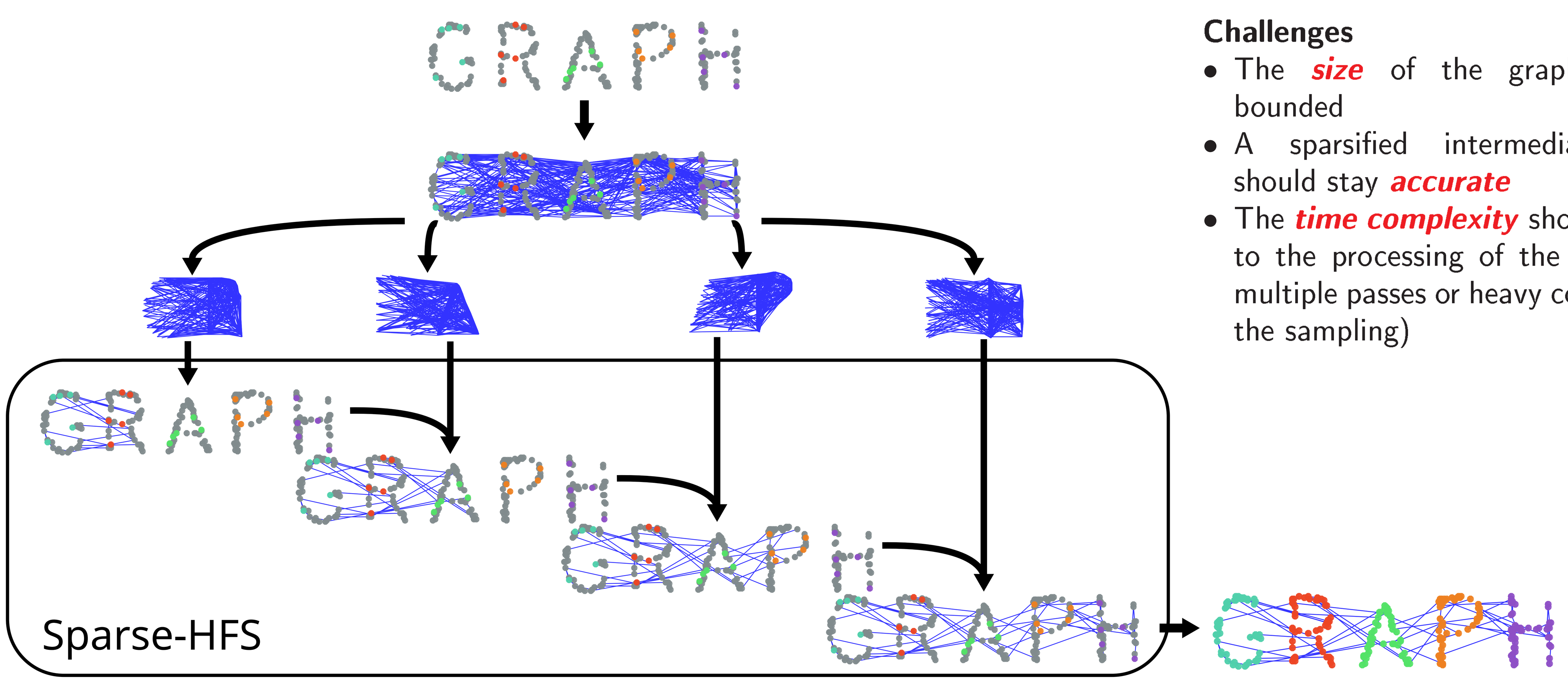
Sparse HFS

Input: Graph $\mathcal{G} = (\mathcal{X}, \mathcal{E})$, labels $\mathbf{y}_\mathcal{S}$, accuracy ε
Output: Solution $\widehat{\mathbf{f}}$, sparsified graph $\mathcal{H}$
Let $\alpha = 1/(1-\varepsilon)$ and $N = \alpha^2 n \log^2(n)/\varepsilon^2$
Partition $\mathcal{E}$ ***in*** $\tau = \lceil m/N \rceil$ ***blocks*** $\Delta_1, \ldots, \Delta_\tau$
if $(\tau > 1)$, $\mathcal{H}_1 =$ **sparsify**$(\emptyset, \Delta_1, N, \alpha)$,
else $\mathcal{H}_1 = \Delta_1$
for $t = 2, \ldots, \tau$ **do**
  Load Δ_t in memory
  Compute $\mathcal{H}_t =$ **sparsify**$(\mathcal{H}_{t-1}, \Delta_t, N, \alpha)$
end for
Center the labels $\widetilde{\mathbf{y}}_\mathcal{S}$
Compute $\widetilde{\mathbf{f}}$ with Stable-HFS with $\widetilde{\mathbf{y}}_\mathcal{S}$ using a ***SDD solver***

Sparsify routine (*adapted from [6]*)
Input: Sparsifier $\mathcal{H}$, block Δ, # edges N, effective resistance acc. α
Output: A sparsifier $\mathcal{H}'$, probabilities $\{\widetilde{p}'_e : e \in \mathcal{H}'\}$.
Compute estimates of $\widetilde{R}'_e$ for any edge in $\mathcal{H} + \Delta$ such that $1/\alpha \leq \widetilde{R}'_e / R'_e \leq \alpha$ with an SDD solver [2]
Compute probabilities and weights

$$\widetilde{p}'_e = (a_e \widetilde{R}'_e)/(\alpha(n-1)), \quad w_e = a_e/(N \widetilde{p}'_e)$$

For all edges $e \in \mathcal{H}$ compute $\widetilde{p}'_e \leftarrow \min\{\widetilde{p}_e, \widetilde{p}'_e\}$ and initialize $\mathcal{H}' = \emptyset$
for all edges $e \in \mathcal{H}$ **do**
  Add edge e to $\mathcal{H}'$ with weight w_e with prob. $\widetilde{p}'_e / \widetilde{p}_e$
end for
for all edges $e \in \Delta$ **do**
  for $i = 1$ to N **do**
    Add edge e to $\mathcal{H}'$ with weight w_e with prob. $\widetilde{p}'_e$
  end for
end for



Challenges

- The ***size*** of the graph should stay bounded
- A sparsified intermediate sub-graph should stay ***accurate***
- The ***time complexity*** should stay limited to the processing of the edges (i.e., no multiple passes or heavy computations for the sampling)

Theoretical Guarantees

Setting. ε is the accuracy and δ is the probability of failure. The graph $\mathcal{G}$ has n nodes, m edges, eigenvalues $0 = \lambda_1(\mathcal{G}) < \lambda_2(\mathcal{G}) \leq \ldots \leq \lambda_n(\mathcal{G})$. The labels are bounded $|\mathbf{y}(x)| \leq M$ and $\mathcal{F}$ is the set of centered functions such that $|\mathbf{f}(x) - \mathbf{y}(x)| \leq c$.

Lemma 1. For any partitioning of $\mathcal{E}$ into τ blocks, Sparse-HFS returns a graph $\mathcal{H}$ s.t. for any $i = 1 \ldots n$ with prob. $1 - \delta$

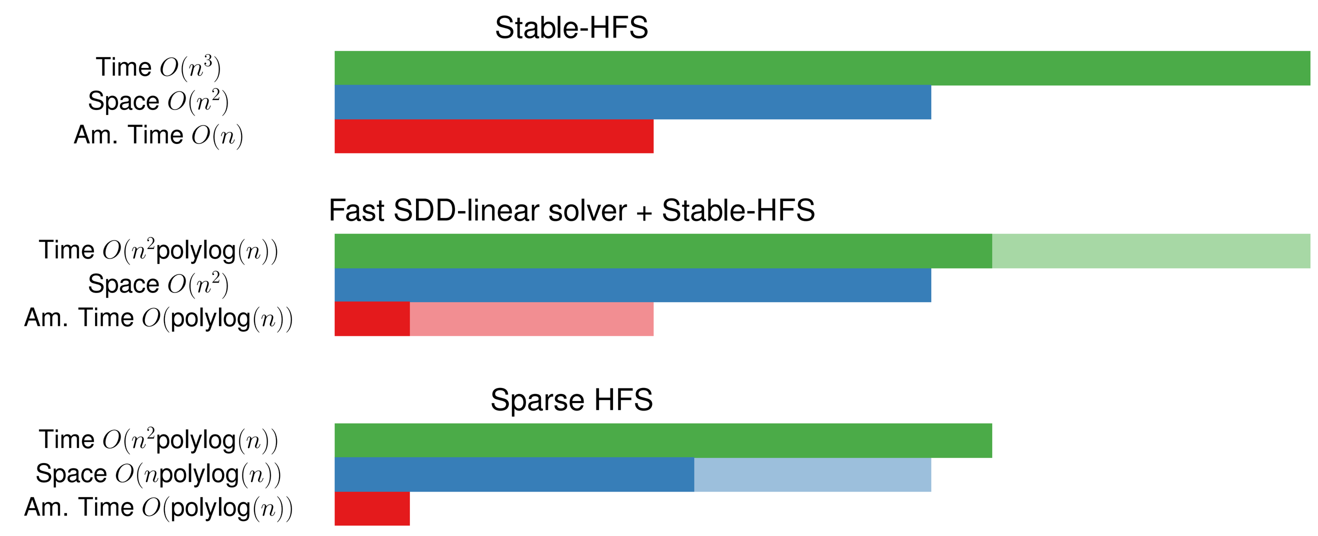
$$\boxed{(1-\varepsilon)\lambda_i(\mathcal{G}) \leq \lambda_i(\mathcal{H}) \leq (1+\varepsilon)\lambda_i(\mathcal{G}),}$$

Theorem 1. If the labels $\widetilde{\mathbf{y}}_\mathcal{S}$ are centered, then Sparse-HFS (ε) returns a function $\widetilde{\mathbf{f}}$ that w.p. $1 - \delta$ satisfies

$$R(\widetilde{\mathbf{f}}) \leq \widehat{R}(\widehat{\mathbf{f}}) + \beta + \left(2\beta + \frac{c^2(l+u)}{lu}\right) \sqrt{\frac{\pi(l,u) \ln \frac{1}{\delta}}{2}} + \boxed{\left(\frac{\varepsilon l \gamma \lambda_2(\mathcal{G}) M}{((1-\varepsilon) l \gamma \lambda_2(\mathcal{G}) - 1)^2}\right)^2},$$

$$\beta \leq \boxed{\frac{1.5 M \sqrt{l}}{((1-\varepsilon) l \gamma \lambda_2(\mathcal{G}) - 1)^2} + \frac{4M}{(1-\varepsilon) l \gamma \lambda_2(\mathcal{G}) - 1}}.$$

Complexity Comparison



Lemma 1-bis. Let

$$\boxed{N = \alpha^2 \frac{n \log^2(n)}{\varepsilon^2},}$$

with $\alpha = 1/(1-\varepsilon)$ and $\tau = m/N$, then with prob. $1-\delta$ Sparse-HFS has an amortized time per edge of $\mathcal{O}(\log^3(n))$ and it requires $\mathcal{O}(N)$ memory.

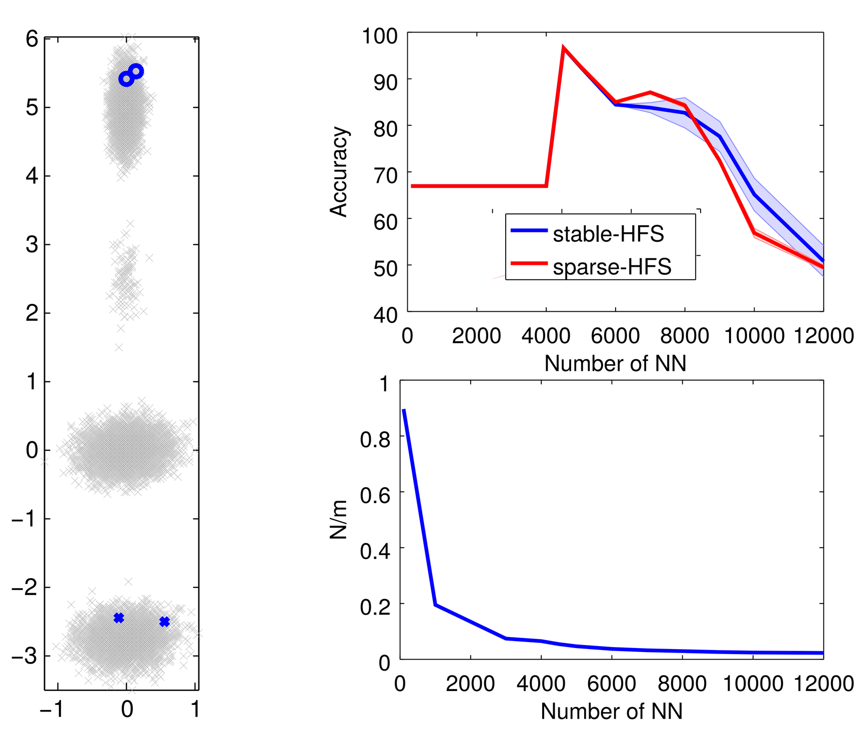
Experiments

	Guarant.	Space	Preproc. Time	Solving Time
Sparse-HFS	✓	$N = \mathcal{O}(n \log^2(n))$	$\mathcal{O}(m \log^3(n))$	$\mathcal{O}(N \log(n)) = \mathcal{O}(n \log^3(n))$
Stable-HFS	✓	$\mathcal{O}(m)$	$\mathcal{O}(m)$	$\mathcal{O}(mn)$
Simple-HFS	O	$\mathcal{O}(m)$	$\mathcal{O}(mq)$	$\mathcal{O}(q^4)$
EigFun	✗	$\mathcal{O}(nd + nq + b^2)$	$\mathcal{O}(qb^3 + db^3)$	$\mathcal{O}(q^3 + nq)$
SubSampling	✗	$\mathcal{O}(sk)$	$\mathcal{O}(m)$	$\mathcal{O}(s^2k + n)$

- HFS-based algorithms: k-NN used to construct the graph ($m = nk$ edges)
- Simple-HFS [5] q eigenfunctions
- EigFun [3] q eigenfunctions, b bins
- SubSampling s subsampled points, k neighbors

Toy Example

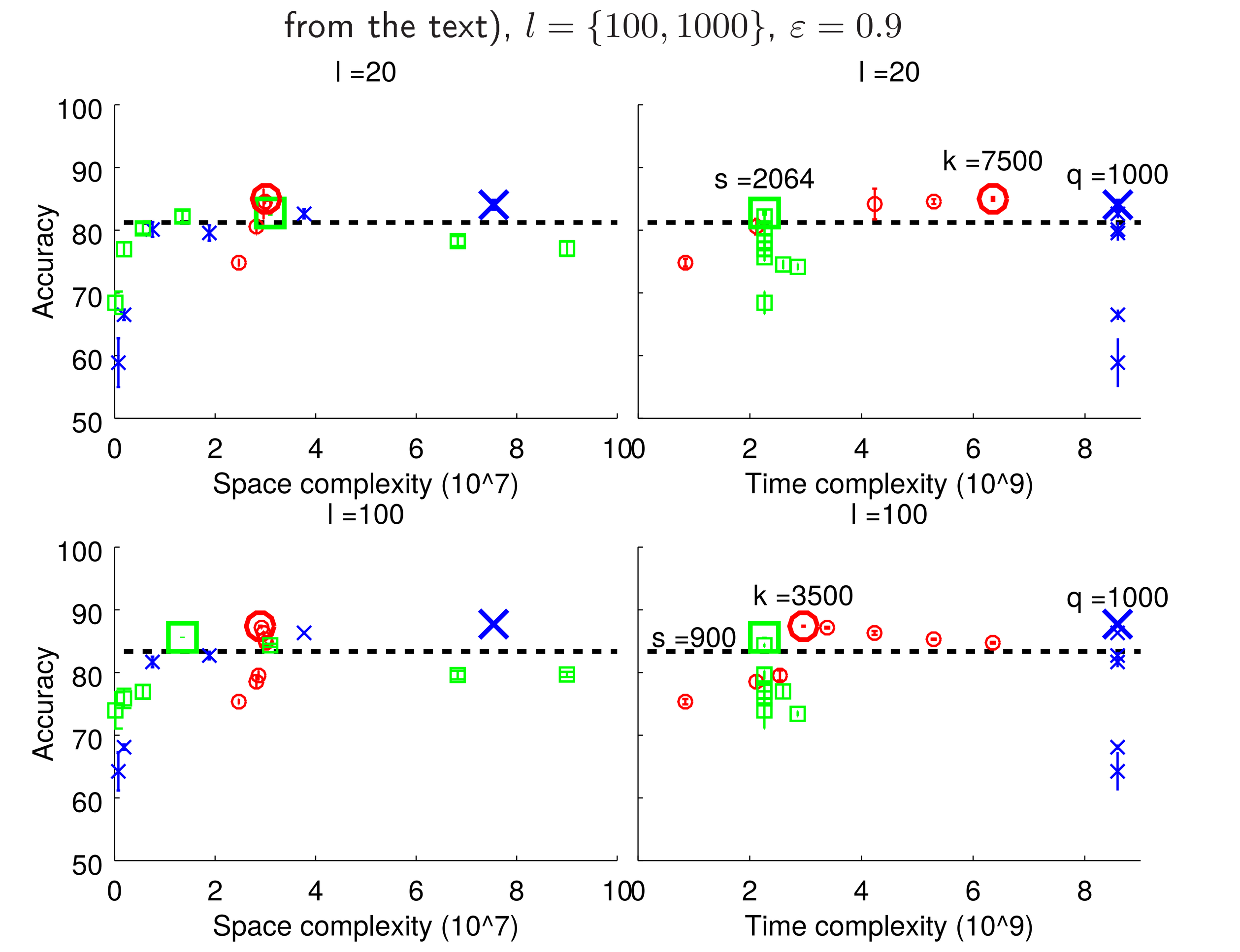
$n = 12100$ points, $d = 2$, $y = -1$ for two top clusters and $y = 1$ for two bottom clusters, $l = 4$ two on top, two on bottom, $\varepsilon = 0.8$



Until $k = 4500$, $\mathcal{G}$ is not connected, the space complexity ratio is $|\mathcal{H}|/|\mathcal{G}| = O(1/k)$.

Spam Classification (TREC07)

$n = 75419$ mails, $y =$ HAM or SPAM, $d = 68697$ features (TF-IDF extracted from the text), $l = \{100, 1000\}$, $\varepsilon = 0.9$



○ Sparse-HFS ($k = 1000..7500$), × EigFun ($b = 50$, $q = 10..1000$), □ SubSampling ($s = 15000$, $k = 100..10000$), - - - 1-NN.