Approximate Manifold Regularization: Scalable Algorithm and Generalization Analysis

Jian Li[1,2], **Yong Liu**[1], **Rong Yin**[1,2] and **Weiping Wang**[1*]

[1]Institute of Information Engineering, Chinese Academy of Sciences
[2]School of Cyber Security, University of Chinese Academy of Sciences
{lijian9026, liuyong, yinrong, wangweiping}@iie.ac.cn

Abstract

Graph-based semi-supervised learning is one of the most popular and successful semi-supervised learning approaches. Unfortunately, it suffers from high time and space complexity, at least quadratic with the number of training samples. In this paper, we propose an efficient graph-based semi-supervised algorithm with a sound theoretical guarantee. The proposed method combines Nystrom subsampling and preconditioned conjugate gradient descent, substantially improving computational efficiency and reducing memory requirements. Extensive empirical results reveal that our method achieves the state-of-the-art performance in a short time even with limited computing resources.

1 Introduction

Recently, the explosive growth of computing power and applications of the network makes data generation and acquisition more easily. However, most of the collected data are unlabeled, while data annotation is laborious. Further, semi-supervised learning (SSL) methods are developed to estimate specific learner from a few labeled samples together with a significant amount of unlabeled data, such as transductive support vector machines [Joachims, 1999] and graph-based methods [Belkin *et al.*, 2006; Camps-Valls *et al.*, 2007]. Graph-based manifold regularization methods draw wide attention of SSL area due to their good performance and relative simplicity of implementation [Belkin *et al.*, 2006]. Despite those advantages of manifold regularization, it remains challenges to process gigantic datasets, for suffering high computational complexity, typically kernel matrix related operations at least $\mathcal{O}(n^2)$ and construction of graph Laplacian at least $\mathcal{O}(n \log n)$, where n is total sample size.

To tackle those scalability issues, many approaches were proposed [Liu *et al.*, 2012; Jiang *et al.*, 2017; Liu *et al.*, 2019]: (1) *Accelerate construction of Laplacian graph*. Methods based on the fast spectral decomposition of Laplacian matrix have been well-studied in [Talwalkar *et al.*, 2013], which use a few eigenvalues of graph Laplacian to represent manifold structure. Graph sparsification approaches were devised to approximate Laplacian graph by a line or spanning tree [Cesa-Bianchi *et al.*, 2013] and improved by minimizing tree cut (MTC) in [Zhang *et al.*, 2016]. (2) *Accelerate operations associated with kernel matrix*. Several distributed approaches have been applied to semi-supervised learning [Chang *et al.*, 2017], decomposing a large scale problem into smaller ones. Anchor Graph regularization (Anchor) constructs an anchor graph with the training samples and a few anchor points to approximate Laplacian graph [Liu *et al.*, 2010]. The work of [McWilliams *et al.*, 2013; Rastogi and Sampath, 2017] applied random projections including Nyström methods and random features into manifold regularization. Gradient methods are introduced to solve manifold regularization on the primal problem, such as preconditioned conjugate gradient [Melacci and Belkin, 2011], stochastic gradient descent [Wang *et al.*, 2012].

In this paper, we focus on the latter scalability issue. With sound theoretical guarantees, we devise a novel graph-based SSL framework, substantially reducing computational time and memory requirements. More precisely, our approach approximates Laplacian regularized least squares (LapRLS) by Nyström methods and then accelerates the solution with preconditioned conjugate gradient methods. It's a non-trivial extension of FALKON [Rudi *et al.*, 2017] to graph SSL with technical challenges in algorithm design and theoretical analysis. Theoretical analysis demonstrates that $\mathcal{O}(\sqrt{m})$ labeled samples and $\mathcal{O}(\log m)$ iterations (m is the number of labeled samples) can guarantee good statistical properties. Complexity analysis shows our method solve LapRLS with $\mathcal{O}(n\sqrt{n})$ time and $\mathcal{O}(n)$ space (n is the number of all samples).

2 Related Work

To overcome the computational and memory bottleneck of LapRLS, practical algorithms were developed, including Nyström methods [Williams and Seeger, 2001] of which statistical properties are well studied in [Rastogi and Sampath, 2017], and preconditioned conjugate gradient (PCG) which reduces the number of iterations [Cutajar *et al.*, 2016]. FALKON approach combines Nyström methods and PCG in supervised learning [Rudi *et al.*, 2017]. Further, our work extends the combination to SSL with high computation gains and sound statistical guarantees. The approach improves computational efficiency from $\mathcal{O}(n^3)$ to $\mathcal{O}(n\sqrt{n})$ and reduce memory cost from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$.

*Corresponding author

3 Preliminaries

3.1 Problem Definition

Assume there is a fixed but unknown distribution ρ on $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Y} = \mathbb{R}$. Further, m labeled samples $\{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_m, y_m)\} \in \mathcal{X} \times \mathcal{Y}$ are drawn i.i.d from ρ and $n - m$ unlabeled samples $\{\mathbf{x}_{m+1}, \cdots, \mathbf{x}_n\} \in \mathcal{X}$ are drawn i.i.d according to the marginal distribution ρ_X of ρ.

3.2 Manifold Regularization

Manifold learning methods based on the spectral graph, known as graph-based SSL, is a typical solution to semi-supervised learning [Zhu *et al.*, 2003; Belkin *et al.*, 2006], which is to find a smooth low-dimensional manifold embedded in the high-dimensional vector space, based on sample points. Correctly, Laplacian regularization [Belkin *et al.*, 2006] is extensively used in graph-based SSL.

For a Mercer kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, there is an associated reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ of functions $f : \mathcal{X} \to \mathbb{R}$ with corresponding norm $\| \cdot \|_{\mathcal{H}}$. The following optimization is considered in manifold regularization:

$$\widehat{f}_\lambda = \arg\min_{f \in \mathcal{H}} \sum_{i=1}^{m} \ell(y_i, f(\mathbf{x}_i)) + \lambda_A \|f\|_{\mathcal{H}}^2 + \lambda_I \mathbf{f}^T \mathbf{L} \mathbf{f}, \quad (1)$$

where ℓ is loss function, $\mathbf{L}$ is Laplacian matrix by $\mathbf{L} = \mathbf{D} - \mathbf{W}$, $\mathbf{f} = [f(\mathbf{x}_1), \cdots, f(\mathbf{x}_n)]^T$, λ_A controls the complexity of the function in the *ambient* space, and λ_I controls the complexity of the function in the *intrinsic* space. Here, $\mathbf{W} \in \mathbb{R}^{n \times n}$ records undirected weight between points and the diagonal matrix $\mathbf{D}$ is given by $D_{ii} = \sum_{j=1}^{n} W_{ij}$.

The minimizer of the optimization problem (1) admits an expansion in terms of both labeled and unlabeled data

$$\widehat{f}_\lambda(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}_i, \mathbf{x}).$$

3.3 Laplacian Regularized Least Squares (LapRLS)

With squared loss function, the problem (1) becomes LapRLS

$$\arg\min_{f \in \mathcal{H}} \sum_{i=1}^{m} (y_i - f(\mathbf{x}_i))^2 + \lambda_A \|f\|_{\mathcal{H}}^2 + \lambda_I \mathbf{f}^T \mathbf{L} \mathbf{f}. \quad (2)$$

Setting the derivative of the objective function be zero leads a closed form solution

$$\widehat{\boldsymbol{\alpha}} = (\mathbf{J}\mathbf{K} + \lambda_A \mathbf{I} + \lambda_I \mathbf{L}\mathbf{K})^{-1} \mathbf{y}_n, \quad (3)$$

where $\mathbf{K}_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$ is $n \times n$ kernel matrix on train data, $\mathbf{J} = \text{diag}(1, \cdots, 1, 0, \cdots, 0)$ with the first m diagonal entries as 1 and the rest 0, and $\mathbf{y}_n = [y_1, y_2, \cdots, y_m, 0, \cdots, 0]^T$ with corresponding m labels and the rest filled by 0. Note that when $\lambda_I = 0$, Equation (3) gives zero coefficients over unlabeled data, thus the form reduces to the standard RLS.

4 Algorithm

We devise a fast and scalable graph-based semi-supervised learning framework `Nyström-PCG` shown as Algorithm 1, which consists of two steps: (1) Nyström with uniform sampling on train data for the LapRLS problem, resulting in a linear system $\mathbf{H}\boldsymbol{\alpha} = \mathbf{z}$. (2) Define a preconditioner $\mathbf{P}$ to approximate $\mathbf{H}$, and then solve $\mathbf{P}^{-1}\mathbf{H}\boldsymbol{\alpha} = \mathbf{P}^{-1}\mathbf{z}$ by PCG.

4.1 Nyström subsampling on LapRLS

We consider Nyström subsampling to reduce memory requirement, which uses a smaller matrix obtained from random column sampling to approximate the empirical kernel matrix. Thus, a smaller hypothesis space $\mathcal{H}_s$ is introduced

$$\mathcal{H}_s = \{f \in \mathcal{H} | f = \sum_{i=1}^{s} \alpha_i K(\mathbf{x}_i, \cdot), \boldsymbol{\alpha} \in \mathbb{R}^s\},$$

where $s \leq n$ and $\mathbf{x}_s = (\widetilde{\mathbf{x}}_1, \cdots, \widetilde{\mathbf{x}}_s)$ are Nyström centers selected by uniform subsampling from the training set. The minimizer of LapRLS (2) over the space $\mathcal{H}_s$ is in the form:

$$\begin{aligned} &\widehat{f}_\lambda^s(\mathbf{x}) = \sum_{i=1}^{s} \alpha_i K(\mathbf{x}_i, \mathbf{x}), \quad \text{with} \\ &\boldsymbol{\alpha} = (\underbrace{\mathbf{K}_{ms}^T \mathbf{K}_{ms} + \lambda_A \mathbf{K}_{ss} + \lambda_I \mathbf{K}_{ns}^T \mathbf{L} \mathbf{K}_{ns}}_{\mathbf{H}})^\dagger \underbrace{\mathbf{K}_{ms}^T \mathbf{y}}_{\mathbf{z}}, \end{aligned} \quad (4)$$

where $\mathbf{H}^\dagger$ denotes the Moore-Penorse pseudoinverse of a matrix $\mathbf{H}$, $(\mathbf{K}_{ms})_{ij} = K(\mathbf{x}_i, \widetilde{\mathbf{x}}_j)$ with $i \in \{1, \cdots, m\}$ and $j \in \{1, \cdots, s\}$, $(\mathbf{K}_{ss})_{kj} = K(\widetilde{\mathbf{x}}_k, \widetilde{\mathbf{x}}_j)$ with $k, j \in \{1, \cdots, s\}$ and $\mathbf{y} = [y_1, \cdots, y_m]^T \in \mathbb{R}^m$.

4.2 Solving the linear system by preconditioning

Nyström subsampling for LapRLS problems resulting solution (4) is also a linear system, so we consider how to accelerate the solution by preconditioning that is

$$\mathbf{P}^{-1}\mathbf{H}\boldsymbol{\alpha} = \mathbf{P}^{-1}\mathbf{z}.$$

As we all know, the number of iterations for preconditioning methods depends on the condition number $cond(\mathbf{P}^{-1}\mathbf{H})$, such that the preconditioner needs to be approximate to $\mathbf{H}$. To obtain a smaller condition number but also avoid inefficient computation, we define the following preconditioners:

- $m \leq \sqrt{n}$

$$\mathbf{P} = \mathbf{K}_{ms}^T \mathbf{K}_{ms} + \lambda_A \mathbf{K}_{ss} + \frac{\lambda_I n^2}{s^2} \mathbf{K}_{ss} \mathbf{L}_{ss} \mathbf{K}_{ss}. \quad (5)$$

- $m > \sqrt{n}$

$$\mathbf{P} = \frac{m}{s} \mathbf{K}_{ss}^T \mathbf{K}_{ss} + \lambda_A \mathbf{K}_{ss} + \frac{\lambda_I n^2}{s^2} \mathbf{K}_{ss} \mathbf{L}_{ss} \mathbf{K}_{ss}. \quad (6)$$

In each iteration of any PCG solver, calculation of $\mathbf{H}\boldsymbol{\alpha}$ is needed. To accelerate computation, $\mathbf{H}\boldsymbol{\alpha}$ is decomposed into a series of matrix-vector multiplications

$$\mathbf{H}\boldsymbol{\alpha} = \mathbf{K}_{ms}^T (\mathbf{K}_{ms} \alpha) + \lambda_A \mathbf{K}_{ss} \alpha + \lambda_I \mathbf{K}_{ns}^T (\mathbf{L}(\mathbf{K}_{ns} \alpha)). \quad (7)$$

Remark 1. *We use LU or QR decomposition to calculate matrix inversion $\mathbf{P}^{-1}$ because they show significant improvement in speed than Cholesky decomposition.*

Remark 2. *The storage of kernel matrix $\mathbf{K}_{ns}$ needs at least $\mathcal{O}(ns)$ memory, but it turns to be $\mathcal{O}(s^2)$ when we perform matrix multiplications in $s \times s$ blocks.*

Algorithm 1 Nyström LapRLS with PCG (`Nyström-PCG`)

Input: m labeled samples $\{(\mathbf{x}_i, y_i)\}_{i=1}^m$, $n-m$ unlabeled samples $\{\mathbf{x}_j\}_{j=m+1}^n$. Parameters: λ_A, λ_I, kernel method K and subsampling size s.

Output: coefficients $\boldsymbol{\alpha}$

1: Construct Laplacian graph matrix $\mathbf{L}$.
2: Select s Nyström centers with uniform sampling from training set $\{\widetilde{\mathbf{x}}_1, \cdots, \widetilde{\mathbf{x}}_s\} \in \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$.
3: Calculate inverse of the preconditioner $\mathbf{P}^{-1}$ (5) or (6).
4: Use any PCG solver to solve $\mathbf{P}^{-1}\mathbf{H}\boldsymbol{\alpha} = \mathbf{P}^{-1}\mathbf{z}$ with calculating $\mathbf{H}\boldsymbol{\alpha}$ by Eq. (7) and performing matrix multiplication in blocks.

4.3 Complexity Analysis

Time complexity.
Before the start of iterations, we need to compute the preconditioner $\mathbf{P}$, which needs $\mathcal{O}(ms^2 + s^2 + s^3)$ time for (5) and $\mathcal{O}(s^3)$ time for (6). The inverse of preconditioner $\mathbf{P}^{-1}$ needs $\mathcal{O}(s^3)$ as well. In each iteration of PCG, calculation of $\mathbf{H}\boldsymbol{\alpha}$ and $\mathbf{P}^{-1}\mathbf{v}$ are needed, where $\mathbf{v}$ represents a vector $\mathbf{H}\boldsymbol{\alpha}$ or $\mathbf{z}$. For $\mathbf{P}^{-1}\mathbf{v}$, it just needs $\mathcal{O}(s^2)$ for matrix-vector multiplication. And computation for $\mathbf{H}\boldsymbol{\alpha}$ as in Eq. (7), it needs $\mathcal{O}(ms + s^2 + ns + nk)$ where $\mathbf{L}$ is usually a sparse matrix which is almost at $n \times k$ size. Thus, each iteration needs $\mathcal{O}(ns)$ time. Combing calculation of $\mathbf{P}^{-1}$ and t iterations of PCG, total time complexity is $\mathcal{O}(s^3 + nst)$.

Space complexity.
Main memory requirement comes from storing $\mathbf{K}_{ns}$ which is at $\mathcal{O}(ns)$ and $\mathbf{L}$ which is at $\mathcal{O}(nk)$, where $k \ll s$ is a small constant. Indeed, matrix multiplications are performed in a series of $s \times s$ blocks so that space complexity can be $\mathcal{O}(s^2)$.

5 Theoretical Result

Define $\widehat{f}^s_{\lambda,t}$ as the estimator of `Nyström-PCG`. The goal of nonparametric regression is to minimize the *excess risk*:

$$\mathcal{E}(f) = \int_{\mathcal{X}\times\mathcal{Y}} (f(\mathbf{x}) - y)^2 d\rho(\mathbf{x}, y).$$

Under some basic conditions in nonparametric learning, we derive the *excess risk* bound for `Nyström-PCG`.

Assumption 1 (Basic assumptions)**.** *(1) There exists $\kappa \geq 1$ such that $K(x,x) \leq \kappa^2$ for any $x \in X$. (2) There exists $f_{\mathcal{H}} \in \mathcal{H}$, such that $\mathcal{E}(f_{\mathcal{H}}) = \inf_{f\in\mathcal{H}} \mathcal{E}(f)$.*

Theorem 1. *Let $\eta \in (0, 1]$ and $m_0 \in \mathbb{N}$. Under Assumption 1 and assume $|y| \leq b$, $\forall b > 0$, if $m \geq m_0$ and*

$$\lambda = \frac{8\kappa^2}{\sqrt{m}} \log\left(\frac{4}{\eta}\right), \quad s \geq 5(67 + 20\sqrt{m}) \log\frac{48\kappa^2 n}{\eta}$$
$$t \geq \frac{1}{2}\log m + 2\log(2b + 3\kappa) + 5,$$

then the following holds with probability at least $1 - \eta$,

$$\mathcal{E}(\widehat{f}^s_{\lambda,t}) - \mathcal{E}(f_{\mathcal{H}}) \leq \frac{c_0 \log^2 \frac{24}{\eta}}{\sqrt{m}},$$

where the constant m_0 is independent on λ, s, m, t and the constant c_0 is independent on λ, s, m, t, η.

Estimators	Time	Space
RLS-Direct	$\mathcal{O}(m^3)$	$\mathcal{O}(m^2)$
LapRLS-Direct	$\mathcal{O}(n^3)$	$\mathcal{O}(n^2)$
LapRLS-CG	$\mathcal{O}(n^{2.5})$	$\mathcal{O}(n^2)$
LapRLS-PCG	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2)$
Nyström-Direct	$\mathcal{O}(n^2)$	$\mathcal{O}(n)$
Nyström-CG	$\mathcal{O}(n^{1.75})$	$\mathcal{O}(n)$
`Nyström-PCG`	$\mathcal{O}(n^{1.5})$	$\mathcal{O}(n)$

Table 1: Summary of time complexity and space complexity in terms of various methods. Here, we omit logarithmic terms.

The above result provides the desired bound and proof details are deferred in the appendix. Under basic assumptions, the obtained learning rate is the same as the full KRR estimator and is optimal in a minmax sense [Caponnetto and De Vito, 2007]. Theorem 1 shows that $\mathcal{O}(\sqrt{m})$ Nyström labeled points and $\mathcal{O}(\log m)$ iterations can obtain $\mathcal{O}(1/\sqrt{m})$ learning rate. In practical, we sampled uniformly $\mathcal{O}(\sqrt{m})$ labeled points and $\mathcal{O}(\sqrt{n-m})$ unlabeled points, thus the size of sampled examples is $\mathcal{O}(\sqrt{n})$. With omitting logarithmic terms, the approach needs $\mathcal{O}(n^{1.5})$ time and $\mathcal{O}(n)$ space.

6 Compared methods

In this part, we introduce compared methods including the standard RLS, LapRLS and Nyström LapRLS. Results in Table 1 show that the proposed approach Nyström LapRLS with PCG (`Nyström-PCG`) can remarkably reduce memory requirements and improve computational efficiency.

Standard RLS (RLS)
Consider the standard RLS only using labeled data

$$f^*(\mathbf{x}) = \sum_{i=1}^m \alpha_i^* \mathbf{K}_{mm}(\mathbf{x}_i, \mathbf{x}),\ \boldsymbol{\alpha}^* = (\mathbf{K}_{mm} + \lambda\mathbf{I})^{-1}\mathbf{y},$$

where $\mathbf{K}_{mm}$ is the empirical kernel matrix on labeled data and $\mathbf{y}$ is corresponding labels. The computation cost is standard, requiring $\mathcal{O}(m^3)$ time and $\mathcal{O}(m^2)$ memory.

Laplacian Regularized Least Squares (LapRLS)
The linear system of LapRLS has been derived in (3) is

$$\widehat{f}_\lambda(\mathbf{x}) = \sum_{i=1}^n \widehat{\alpha}_i K(\mathbf{x}_i, \mathbf{x}),\ \widehat{\boldsymbol{\alpha}} = (\mathbf{JK} + \lambda_A\mathbf{I} + \lambda_I\mathbf{LK})^{-1}\mathbf{y}_n.$$

In terms of LapRLS, the space complexity is $\mathcal{O}(n^2)$, but time complexity depends on optimization algorithms, $\mathcal{O}(n^3)$ for matrix inversion and $\mathcal{O}(n^2 t)$ for gradient methods. Specifically, the number of iterations is $t = \mathcal{O}(\sqrt{n}\log n)$ for CG and $t = \mathcal{O}(\log n)$ for PCG [Rudi *et al.*, 2017].

LapRLS with Nyström subsampling (Nyström)
The linear system of LapRLS with Nyström subsampling to overcome memory bottleneck has been given in (4). According to Theorem 1, we set the number of Nyström centers be $s = \mathcal{O}(\sqrt{n})$. Space complexity of Nyström LapRLS is $\mathcal{O}(s^2)$ by matrix blocks multiplications, that is $\mathcal{O}(n)$. For direct methods, time complexity depends on computing $\mathbf{H}$ with $\mathcal{O}(ns^2)$, that is $\mathcal{O}(n^2)$. For CG methods, time complexity is $\mathcal{O}(nst)$ where $t = \mathcal{O}(\sqrt{s}\log s)$, that is $\mathcal{O}(n^{1.75})$.

dataset	sample size	RLS-CG	LapRLS-CG	LapRLS-PCG	Nyström-CG	Nyström-PCG
space_ga	3107	1.251±0.004	**1.210±0.004**	**1.210±0.004**	**1.210±0.004**	**1.210±0.004**
phishing	11055	0.426±0.049	0.294±0.005	**0.273±0.007**	0.295±0.005	0.275±0.008
a8a	22696	0.702±0.002	**0.664±0.002**	**0.664±0.002**	0.664±0.002	0.664±0.002
w7a	24692	0.291±0.002	**0.283±0.002**	**0.283±0.002**	0.284±0.002	0.284±0.002
a9a	32561	0.698±0.005	**0.664±0.000**	**0.664±0.002**	**0.664±0.000**	**0.664±0.002**
ijcnn1	49990	0.434±0.005	**0.389±0.002**	**0.389±0.002**	0.393±0.001	0.463±0.001
cod-rna	59535	0.686±0.002	/	/	0.614±0.001	0.614±0.001
connect-4	67757	0.781±0.015	/	/	**0.739±0.002**	**0.739±0.002**
skin_nonskin	245057	3.119±0.023	/	/	**2.620±0.043**	**2.620±0.043**
YearPrediction	463715	0.198±0.001	/	/	**0.187±0.001**	**0.187±0.001**

Table 2: Comparison of average root mean square error between Nyström-PCG and RLS-CG, LapRLS-CG, LapRLS-PCG, Nyström LapRLS-CG. We bold the best results and underline the results of the other methods which are not significantly worse than the best one.

	RLS-CG		LapRLS-CG		LapRLS-PCG		Nyström-CG		Nyström-PCG	
	iter	time	iter	time	iter	time	iter	time	iter	time
space_ga	11	0.004	23	1.220	5	0.569	23	0.113	**2**	**0.016**
phishing	74	0.031	300	24.20	56	8.210	300	2.470	**3**	**0.045**
a8a	100	0.068	50	189.1	3	20.98	50	44.71	**1**	**4.370**
w7a	13	0.072	32	143.2	2	9.683	213	107.7	**1**	**2.252**
a9a	300	0.529	64	1699	3	30.30	65	70.40	**1**	**4.034**
ijcnn1	242	8.204	57	2154	9	72.41	53	108.8	**5**	**4.186**
cod-rna	96	7.178	/	/	/	/	55	134.6	**7**	**8.154**
connect-4	103	11.07	/	/	/	/	154	186.5	**10**	**4.220**
skin_nonskin	43	91.39	/	/	/	/	65	1490	**3**	**40.05**
YearPrediction	37	236.5	/	/	/	/	94	2479	**2**	**116.1**

Table 3: Comparison of average number of iterations and running time (seconds).

7 Empirical Study

In this section, we conduct empirical experiments of some SSL methods on a range of datasets. We compare five approaches, including standard RLS using CG, LapRLS and Nyström LapRLS in terms of CG and PCG. Nyström LapRLS using PCG is our proposed method, named Nyström-PCG.

Scalable datasets from thousand to hundreds of thousand are used. We apply 8-NN to construct adjacency matrix with weight $W_{ij} = \exp^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2/4}$ as using in many literatures. For each dataset, we use Gaussian kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|/2\sigma^2)$. We choose kernel parameter σ and regular parameters (λ in standard RLS and λ_A, λ_I in LapRLS methods) in $2^i, i \in \{-15, -14, \cdots, 14, 15\}$, by minimizing test error via 10-folds cross-validation.

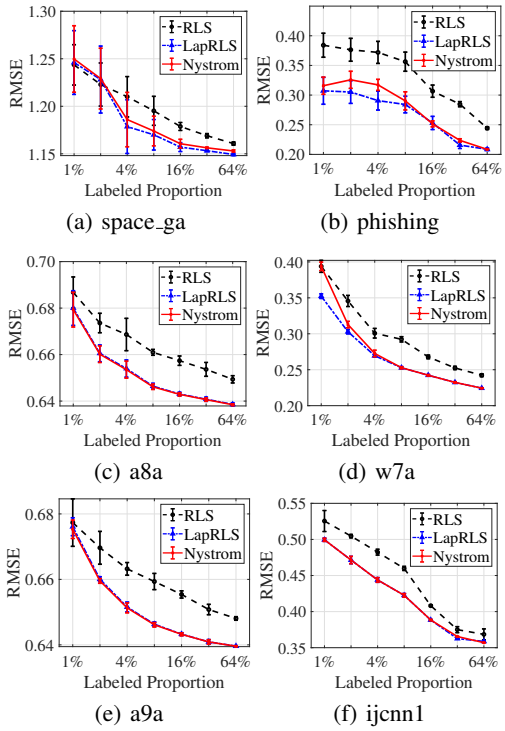
7.1 Accuracy and Speed

Using the chosen parameters determined by 10-folds cross-validation, we run all methods 30 times with randomly select 70% for training and 30% for testing on each dataset. Meanwhile, we randomly select 10% samples ($m = 0.1n$) as labeled data and 10% samples ($s = 0.1n$) as Nyström centers. The use of multiple training/testing partitions allows an estimate of statistical significance between best one and the remainder referring to 95% level of significance under t-test.

Table 2 reports the average root mean square error (RMSE) and Table 3 reports the number of iterations and running time, can be summarized as follows: (1) RLS is defended by other methods using Laplacian regularization on all datasets, while LapRLS-CG or LapRLS-PCG gives the best results almost on all datasets. (2) There is no significant difference in error rate between LapRLS methods and Nyström LapRLS methods. (3) LapRLS-CG and LapRLS-PCG failed on large datasets because of memory limitation. (4) CG and PCG always result in the same accuracy, but PCG need much smaller iterations thus cost less time. (5) Our method Nyström-PCG achieves similar accuracy as LapRLS methods with tens to hundreds of times speeding up.

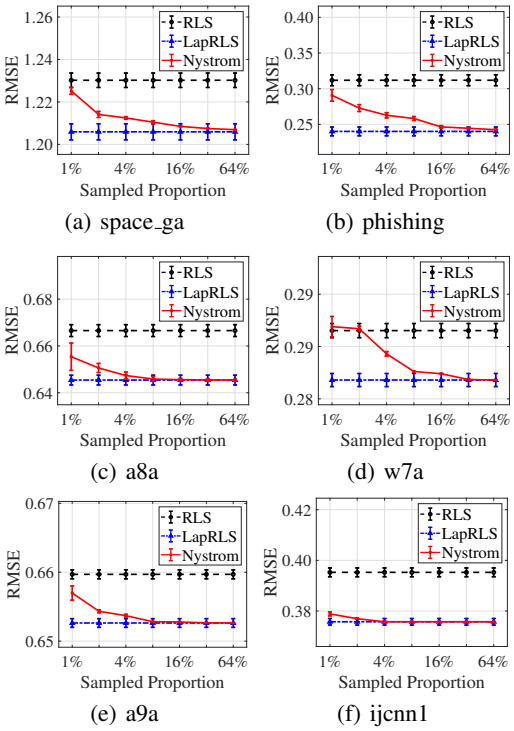
7.2 Influence of label proportion

To explore influence of different label proportions, we let m vary from $n \times \{1\%, 2\%, 4\%, 8\%, 16\%, 32\%, 64\%\}$ and fix sample size as $s = 0.1n$. For better presentation, we use LapRLS represents primal LapRLS approaches, Nyström represents approximate LapRLS using Nyström. By repeating the procedure 20 times for different labeled/unlabeled splits on eight datasets, we report results in Figure 1. We can see that: (1) LapRLS methods are always better than the RLS method, but the difference becomes smaller when labeled partition m becomes larger. (2) Average accuracies of LapRLS and Nyström LapRLS are close when the partition of labeled data is big enough. (3) The standard deviation of all methods becomes smaller as the increase of labeled samples.



(a) space_ga (b) phishing (c) a8a (d) w7a (e) a9a (f) ijcnn1

Figure 1: Average RMSE for different labeled data proportion.



(a) space_ga (b) phishing (c) a8a (d) w7a (e) a9a (f) ijcnn1

Figure 2: Average RMSE for different sample proportion.

7.3 Influence of sample proportion

To explore influence of different sample proportions, we let s vary from $n \times \{1\%, 2\%, 4\%, 8\%, 16\%, 32\%, 64\%\}$ and fixed labeled sample as $m = 0.1n$. After repeating experiments on three different optimizations 20 times on different sample size but same labeled/unlabeled splits on eight datasets, we report results in Figure 2. We can see that: (1) RLS and LapRLS methods always give the same results because they are run on the same labeled/unlabeled splits. (2) Average accuracies Nyström LapRLS become closer to LapRLS as increase with sample size. (3) Nyström methods can achieve good approximation when the sampled proportion is larger than 10%.

8 Conclusion

In this paper, with sound theoretical guarantee, we use Nyström subsampling on LapRLS and solve the linear systems with PCG, substantially reducing memory and computational costs. More precisely, the theoretical analysis provides good convergence rates for Nyström subsampling and Nyström subsampling with PCG, suggesting that $\mathcal{O}(\sqrt{m})$ sample size and $\mathcal{O}(\log m)$ iterations can achieve $\mathcal{O}(1/\sqrt{m})$ error bound. Then complexity analysis show our method achieve good statistical accuracy with $\mathcal{O}(n\sqrt{n})$ time and $\mathcal{O}(n)$ space. Empirical results show our method achieves similar prediction accuracy to LapRLS with higher computational efficiency and less memory requirement.

9 Proof

Based on integral operator framework and common assumptions, we firstly derive the *excess risk* bound for LapRLS with Nyström $\mathcal{E}(\widetilde{f}^s_\lambda) - \mathcal{E}(f_\mathcal{H})$. And then we prove the *excess risk* bound $\mathcal{E}(\widehat{f}^s_{\lambda,t}) - \mathcal{E}(f_\mathcal{H})$ for `Nyström-PCG` in Theorem 1.

Firstly, we introduce definition of the integral operator and standard assumptions in statistical learning [Caponnetto and De Vito, 2007; Liu *et al.*, 2014; Li *et al.*, 2018] and approximation theory [Rudi *et al.*, 2015; Rastogi and Sampath, 2017; Liu *et al.*, 2018].

Definition 1 (Integral operator and the effective dimension)**.** *Let $L_K : L^2(\mathcal{X}, \rho_X) \to L^2(\mathcal{X}, \rho_X)$ be integral operator*

$$(L_K g)(\mathbf{x}) = \int_\mathcal{X} K(\mathbf{x}, \mathbf{z}) g(\mathbf{z}) d\rho_X(\mathbf{z}), \quad \forall g \in L^2(\mathcal{X}, \rho_X),$$

where $L^2(\mathcal{X}, \rho_X) = \{f : \mathcal{X} \to \mathbb{R} \mid \|f\|_\rho^2 = \int |f(\mathbf{x})|^2 d\rho_X < \infty\}$ and the effective dimension is defined as

$$\mathcal{N}(\lambda) = Tr\left((L_K + \lambda I)^{-1} L_K\right).$$

The effective dimension $\mathcal{N}(\lambda)$ measures the size of $\mathcal{H}$ and there exists $Q > 0$ such that always holds $\mathcal{N}(\lambda) \leq Q^2 \lambda^{-1}$. The following Assumption 2 is satisfied when y is bounded, sub-gaussian or sub-exponential.

Assumption 2 (Moment assumption)**.** *Assume there exists $M > 0$ and $\sigma > 0$, such that for all $l \geq 2$ with $l \in \mathbb{N}$,*

$$\int_\mathbb{R} |y|^l d\rho(y|\mathbf{x}) \leq \frac{1}{2} l! M^{l-2} \sigma^2.$$

Assumption 3 (Regularity assumption)**.** *Assume there exists $r \in [1/2, 1]$ and $g \in L^2(\mathcal{X}, \rho_X)$, such that*

$$f_\mathcal{H}(\mathbf{x}) = (L^r g)(\mathbf{x}) \quad \textit{and} \quad \|g\|_\mathcal{H} \leq R.$$

Assumption 3 controls the regularity of $f_{\mathcal{H}}$ and is common in statistical learning [Caponnetto and De Vito, 2007]. It is always satisfied with $r = 1/2$.

We consider a more general multi-penalty regularization scheme for Nyström subsampling LapRLS:

$$\widetilde{f}^s_\lambda = \arg\min_{f\in\mathcal{H}_s} \frac{1}{m}\sum_{i=1}^{m}||f(\mathbf{x}_i)-y_i||^2_Y + \lambda||f||^2_{\mathcal{H}} + \sum_{j=1}^{p}\lambda_j||B_j f||^2_{\mathcal{H}},$$

where $B_j : \mathcal{H} \to \mathcal{H}$ $(1 \le j \le p)$ are bounded operators, $\lambda > 0$, λ_j $(1 \le j \le p)$ are non-negative real numbers.

Theorem 2 (Excess risk bound for Nyström LapRLS)**.** *Let $\eta \in [0,1]$. Assume there exists some constants M, σ. Under Assumption 1, 2 and 3, for sufficiently large sample according to $\lambda \ge \frac{8\kappa^2}{\sqrt{m}}\log\left(\frac{4}{\eta}\right)$ and for subsampling according to $s \ge \max\left\{67\log\left(\frac{12\kappa^2}{\lambda\eta}\right), 2\kappa^2\log\left(\frac{12\kappa^2}{\lambda\eta}\right)\right\}$, the convergence rate of $\widetilde{f}^s_\lambda$ holds with probability at least $1-\eta$*

$$\left[\mathcal{E}(\widetilde{f}^s_\lambda)-\mathcal{E}(f_{\mathcal{H}})\right]^{1/2} \le c_1\lambda^r + c_2\frac{\mathcal{B}_\lambda}{\lambda} + \left\{\frac{8\kappa M}{m\sqrt{\lambda}} + 8\sigma\sqrt{\frac{\mathcal{N}(\lambda)}{m}}\right\}\log\left(\frac{6}{\eta}\right), \tag{8}$$

where the constant c_1 and c_2 do not depend on λ, s, m, t, η, $\mathcal{B}_\lambda = ||\sum_{j=1}^{p}\lambda_j B_j^* B_j||$, s is the sample size on labeled data.

Proof. Following proof details of Theorem 3.1 in [Rastogi and Sampath, 2017], there holds

$$||\psi(L_K)(f^s_{\mathbf{z},\lambda} - f_{\mathcal{H}})||_{\mathcal{H}} \le \psi(\lambda)\{c_1\phi(\lambda) + c_2\frac{\mathcal{B}_\lambda}{\lambda^{3/2}} + (\frac{8\kappa M}{m\lambda} + 8\sigma\sqrt{\frac{\mathcal{N}(\lambda)}{m\lambda}})\log(\frac{4}{\eta})\},$$

where $c_1 = 6R + (5+c_\psi)(3+c_\phi)R$, $c_2 = (5+c_\psi)||f_{\mathcal{H}}||_\rho$ and $\mathcal{B}_\lambda = ||\sum_{j=1}^{p}\lambda_j B_j^* B_j||$. Assumption 3 is always satisfied with $f_{\mathcal{H}}(\mathbf{x}) = (L^r g)(\mathbf{x})$ thus $\phi(\lambda) = \lambda^{r-1/2}$, $\psi(\lambda) = \lambda^{1/2}$ and $\psi(L_K) = L_K^{1/2}$. There holds

$$||L_K^{1/2}(f^s_{\mathbf{z},\lambda} - f_{\mathcal{H}})||_{\mathcal{H}} \le c_1\lambda^r + c_2\frac{\mathcal{B}_\lambda}{\lambda} + \{\frac{8\kappa M}{m\sqrt{\lambda}} + 8\sigma\sqrt{\frac{\mathcal{N}(\lambda)}{m}}\}\log(\frac{6}{\eta}).$$

Using the fact $\varepsilon(f)-\varepsilon(f_{\mathcal{H}}) = ||L_K^{1/2}(f-f_{\mathcal{H}})||^2_{\mathcal{H}}$ for any $f \in \mathcal{H}$ [Caponnetto and De Vito, 2007], we obtain the result. □

Proof of Theorem 1. The proof is mainly following proof techniques of Theorem 8 and Theorem 3 in [Rudi *et al.*, 2017]. Let $\widetilde{f}^s_{\lambda,t}$ be the estimator in Nyström LapRLS optimization after $t \in \mathbb{N}$ iterations. Theorem 8 of [Rudi *et al.*, 2017] gives the connection between $\mathcal{E}(\widetilde{f}^s_{\lambda,t}) - \mathcal{E}(f_{\mathcal{H}})$ and $\mathcal{E}(\widetilde{f}^s_\lambda) - \mathcal{E}(f_{\mathcal{H}})$. Following proof steps in Theorem 8 of [Rudi *et al.*, 2017], there holds

$$\begin{aligned}&\left[\mathcal{E}(\widetilde{f}^s_{\lambda,t}) - \mathcal{E}(f_{\mathcal{H}})\right]^{1/2}\\ \le& c_3\lambda^r + c_2\frac{\mathcal{B}_\lambda}{\lambda} + \left\{\frac{8\kappa M}{m\sqrt{\lambda}} + 8\sigma\sqrt{\frac{\mathcal{N}(\lambda)}{m}}\right\}\log\left(\frac{24}{\eta}\right)\end{aligned}$$

when the Nyström centers are uniformly sampled and

$$s \ge 70\left[1+\frac{\kappa^2}{\lambda}\right]\log\frac{48\kappa^2}{\lambda\eta},\ \ t \ge 2\log\frac{8(b+\kappa||f_{\mathcal{H}}||_{\mathcal{H}})}{R\lambda^r}$$

where $c_3 = 10R + (5+c_\psi)(3+c_\phi)R$, $c_2 = (5+c_\psi)||f_{\mathcal{H}}||_\rho$, $\mathcal{B}_\lambda = ||\sum_{j=1}^{p}\lambda_j B_j^* B_j||$.

Assumption 3 is satisfied with $r = 1/2$. From Theorem 3.2 of [Rastogi and Sampath, 2017], we set $\lambda_j = \lambda^{r+1}$, thus $\mathcal{B}_\lambda \le c_p\lambda^{r+1}$, where $c_p > 0$ only depends on the number of penalty terms. Applying $\mathcal{N}(\lambda) \le Q^2\lambda^{-1}$, $r = 1/2$ and $\lambda = c_\lambda m^{-1/2}$ where $c_\lambda = 8\kappa^2\log\left(\frac{4}{\eta}\right)$, we have

$$\begin{aligned}&\left[\mathcal{E}(\widetilde{f}^s_{\lambda,t}) - \mathcal{E}(f_{\mathcal{H}})\right]^{1/2}\\ \le& c_3\lambda^r + c_2\frac{\mathcal{B}_\lambda}{\lambda} + \left\{\frac{8\kappa M}{m\sqrt{\lambda}} + 8\sigma\sqrt{\frac{\mathcal{N}(\lambda)}{m}}\right\}\log\left(\frac{24}{\eta}\right)\\ \le& c_3 c_\lambda^{1/2}m^{-1/4} + c_2 c_p c_\lambda^{1/2}m^{-1/4}\\ &+(8\kappa M c_\lambda^{-1/2}m^{-3/4} + 8\sigma Q c_\lambda^{-1/2}m^{-1/4})\log\left(\frac{24}{\eta}\right)\\ \le& (c_3 c_\lambda^{1/2} + c_2 c_p + 8\kappa M + 8\sigma Q)\log\left(\frac{24}{\eta}\right)m^{-1/4}.\end{aligned}$$

Therefore, the *excess risk* bound of *Nyström-PCG* holds

$$\mathcal{E}(\widetilde{f}^s_{\lambda,t}) - \mathcal{E}(f_{\mathcal{H}}) \le \frac{c_0\ \log^2\frac{24}{\eta}}{\sqrt{m}}, \tag{9}$$

where $c_0 = (c_3 c_\lambda^{1/2} + c_2 c_p + 8\kappa M + 8\sigma Q)^2$. With conditions

$$\begin{aligned}&m \ge \max\left(\frac{1}{\|L_K\|} + 82\kappa^2\log\frac{373\kappa^2}{\sqrt{\eta}}\right)^2,\\ &s \ge 5(67+20\sqrt{m})\log\frac{48\kappa^2 n}{\eta},\ \lambda = \frac{8\kappa^2}{\sqrt{m}}\log\left(\frac{4}{\eta}\right),\end{aligned} \tag{10}$$

satisfying conditions required by Theorem 8 of [Rudi *et al.*, 2017] and Assumption 2 is satisfied by $M = \sigma = 2b$, thus

$$t \ \ge\ \frac{1}{2}\log m + 2\log(2b+3\kappa) + 5. \tag{11}$$

Combining (9), (10) and (11), we complete the proof. □

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (No.61703396, No.61673293), the Youth Innovation Promotion Association CAS, the National Key Research and Development Program of China (No.2018YFC0823104, No.2016YFB1000604), the Science and Technology Project of Beijing (No.Z181100002718004) and the Excellent Talent Introduction of Institute of Information Engineering of CAS (Y7Z0111107).

References

[Belkin *et al.*, 2006] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7(Nov):2399–2434, 2006.

[Camps-Valls *et al.*, 2007] Gustavo Camps-Valls, Tatyana V Bandos Marsheva, and Dengyong Zhou. Semi-supervised graph-based hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 45(10):3044–3054, 2007.

[Caponnetto and De Vito, 2007] Andrea Caponnetto and Ernesto De Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368, 2007.

[Cesa-Bianchi *et al.*, 2013] Nicolo Cesa-Bianchi, Claudio Gentile, Fabio Vitale, and Giovanni Zappella. Random spanning trees and the prediction of weighted graphs. *Journal of Machine Learning Research*, 14(1):1251–1284, 2013.

[Chang *et al.*, 2017] Xiangyu Chang, Shao-Bo Lin, and Ding-Xuan Zhou. Distributed semi-supervised learning with kernel ridge regression. *Journal of Machine Learning Research*, 18(1):1493–1514, 2017.

[Cutajar *et al.*, 2016] Kurt Cutajar, Michael Osborne, John Cunningham, and Maurizio Filippone. Preconditioning kernel matrices. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 2529–2538, 2016.

[Jiang *et al.*, 2017] Bingbing Jiang, Huanhuan Chen, Bo Yuan, and Xin Yao. Scalable graph-based semi-supervised learning through sparse bayesian model. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2758–2771, 2017.

[Joachims, 1999] Thorsten Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of the 16th International Conference on Machine Learning (ICML)*, volume 99, pages 200–209, 1999.

[Li *et al.*, 2018] Jian Li, Yong Liu, Rong Yin, Hua Zhang, Lizhong Ding, and Weiping Wang. Multi-class learning: From theory to algorithm. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*, pages 1591–1600, 2018.

[Liu *et al.*, 2010] Wei Liu, Junfeng He, and Shih-Fu Chang. Large graph construction for scalable semi-supervised learning. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pages 679–686, 2010.

[Liu *et al.*, 2012] Wei Liu, Jun Wang, and Shih-Fu Chang. Robust and scalable graph-based semisupervised learning. *Proceedings of the IEEE*, 100(9):2624–2638, 2012.

[Liu *et al.*, 2014] Yong Liu, Shali Jiang, and Shizhong Liao. Efficient approximation of cross-validation for kernel methods using Bouligand influence function. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, pages 324–332, 2014.

[Liu *et al.*, 2018] Yong Liu, Hailun Lin, Li-Zhong Ding, Weiping Wang, and Shizhong Liao. Fast cross-validation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2497–2503, 2018.

[Liu *et al.*, 2019] Yong Liu, Shizhong Liao, Shali Jiang, Lizhong Ding, Hailun Lin, and Weiping Wang. Fast cross-validation for kernel-based algorithms. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[McWilliams *et al.*, 2013] Brian McWilliams, David Balduzzi, and Joachim M Buhmann. Correlated random features for fast semi-supervised learning. In *Advances in Neural Information Processing Systems 26 (NIPS)*, pages 440–448, 2013.

[Melacci and Belkin, 2011] Stefano Melacci and Mikhail Belkin. Laplacian support vector machines trained in the primal. *Journal of Machine Learning Research*, 12(Mar):1149–1184, 2011.

[Rastogi and Sampath, 2017] Abhishake Rastogi and Sivananthan Sampath. Manifold regularization based on nystr {\" o} m type subsampling. *arXiv preprint arXiv:1710.04872*, 2017.

[Rudi *et al.*, 2015] Alessandro Rudi, Raffaello Camoriano, and Lorenzo Rosasco. Less is more: Nyström computational regularization. In *Advances in Neural Information Processing Systems 28 (NIPS)*, pages 1657–1665, 2015.

[Rudi *et al.*, 2017] Alessandro Rudi, Luigi Carratino, and Lorenzo Rosasco. Falkon: An optimal large scale kernel method. In *Advances in Neural Information Processing Systems 30 (NIPS)*, pages 3888–3898, 2017.

[Talwalkar *et al.*, 2013] Ameet Talwalkar, Sanjiv Kumar, Mehryar Mohri, and Henry Rowley. Large-scale svd and manifold learning. *Journal of Machine Learning Research*, 14(1):3129–3152, 2013.

[Wang *et al.*, 2012] Zhuang Wang, Koby Crammer, and Slobodan Vucetic. Breaking the curse of kernelization: Budgeted stochastic gradient descent for large-scale svm training. *Journal of Machine Learning Research*, 13(Oct):3103–3131, 2012.

[Williams and Seeger, 2001] Christopher KI Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems 14 (NIPS)*, pages 682–688, 2001.

[Zhang *et al.*, 2016] Yan-Ming Zhang, Xu-Yao Zhang, Xiao-Tong Yuan, and Cheng-Lin Liu. Large-scale graph-based semi-supervised learning via tree laplacian solver. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI)*, pages 2344–2350, 2016.

[Zhu *et al.*, 2003] Xiaojin Zhu, Zoubin Ghahramani, and John D Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International conference on Machine learning (ICML)*, pages 912–919, 2003.