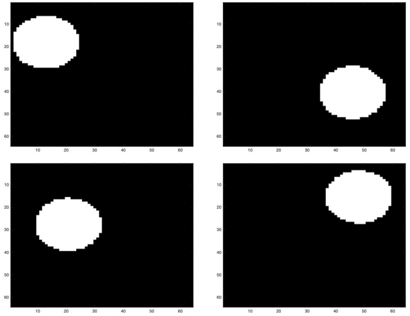
Semi-supervised Learning

Yun You, Wen Zhang, Feng Lu (Francis)

Overview

Problem Description:



Samples: $x_1, \ldots, x_n \in \mathcal{X} \subseteq \mathbb{R}^d$

Only **m<<n** samples have labels

Labels: $y_i = f(x_i) \in \mathcal{Y} = \{1, \ldots, \mathcal{C}\}$

Classifier: $f : \mathcal{X} \mapsto \mathcal{Y}$

$$\boxed{err(\hat{f}) = \frac{1}{n-m} \sum_{i=m+1}^{n} \mathbb{1}\{\hat{f}(x_i) \neq f(x_i)\}.}$$

Eigenvector Classifier

Graph Construction

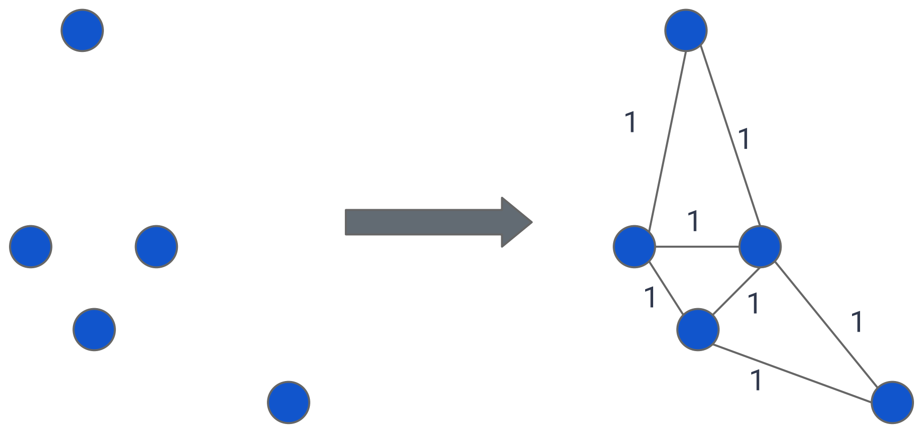
(Weight and Unweighted)

Diffusion Classifier

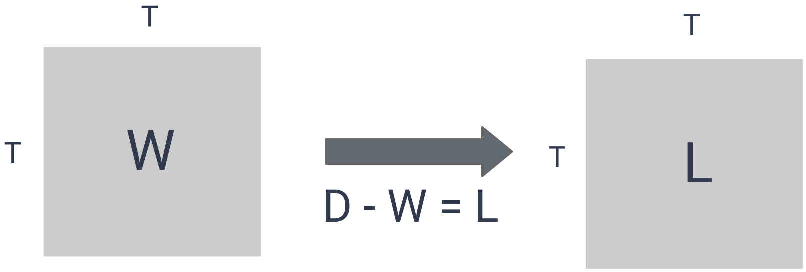
Harmonic / smoothing

Eigenvector Classifier
Simplest Construction

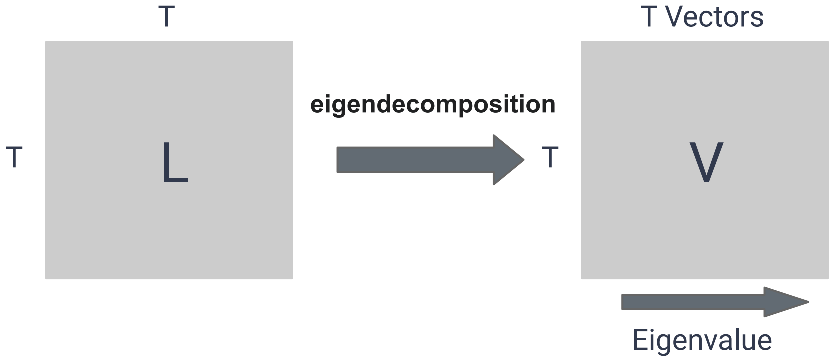
KNN of 2 on 2D:



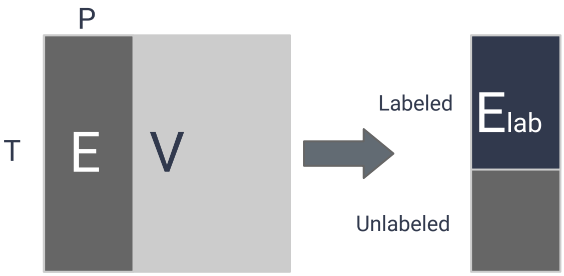
Step 1:



Step 2:

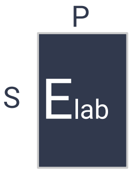


Step 3:



Step 4:

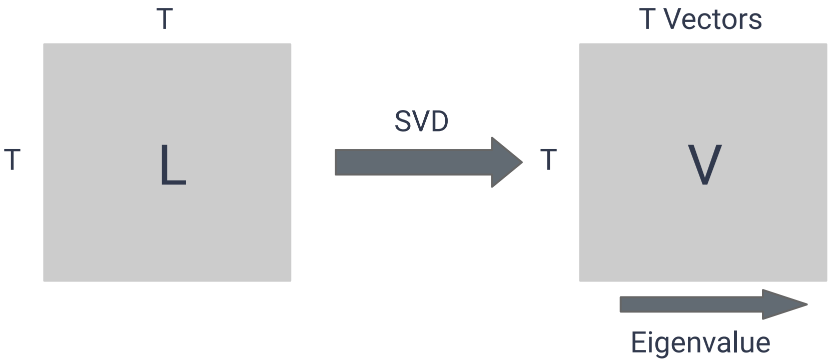
$$c_1 = \begin{bmatrix} -1 \\ 1 \\ -1 \\ -1 \\ 1 \end{bmatrix} \quad c_2 = \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \\ -1 \end{bmatrix} \quad c_3 = \begin{bmatrix} 1 \\ -1 \\ -1 \\ -1 \\ -1 \end{bmatrix}$$



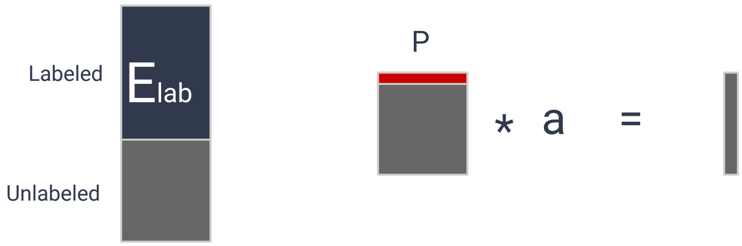
$$\mathbf{a} = \left(\mathbf{E}_{\mathrm{lab}}^{T}\,\mathbf{E}_{\mathrm{lab}}\right)^{-1}\mathbf{E}_{\mathrm{lab}}^{T}\,\mathbf{c}$$

Px1= PxS * SxP * PxS * Sx1

Recall Step 3:



Step 5:



Repeat for all:

$$\mathbf{a_1} = \left(\mathbf{E}_{\text{lab}}^{T}\mathbf{E}_{\text{lab}}\right)^{-1}\mathbf{E}_{\text{lab}}^{T}\mathbf{c_1}$$

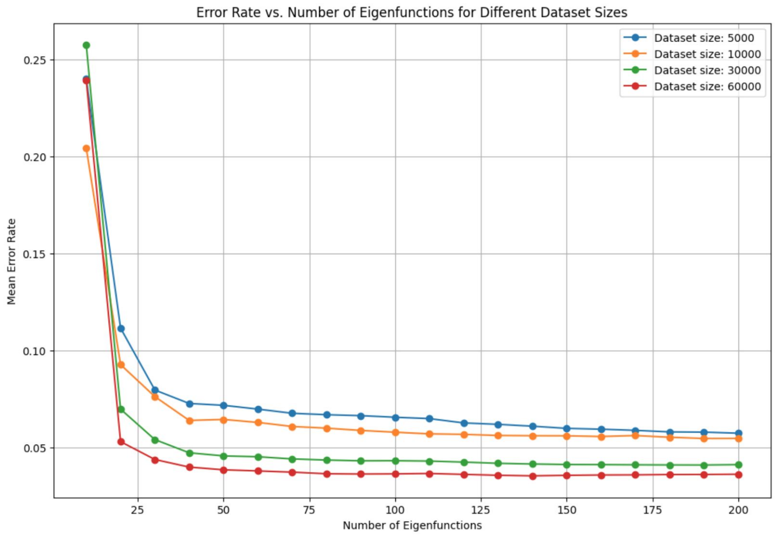
$$\mathbf{a_2} = \left(\mathbf{E}_{\text{lab}}^{T}\mathbf{E}_{\text{lab}}\right)^{-1}\mathbf{E}_{\text{lab}}^{T}\mathbf{c_2}$$

$$\vdots$$

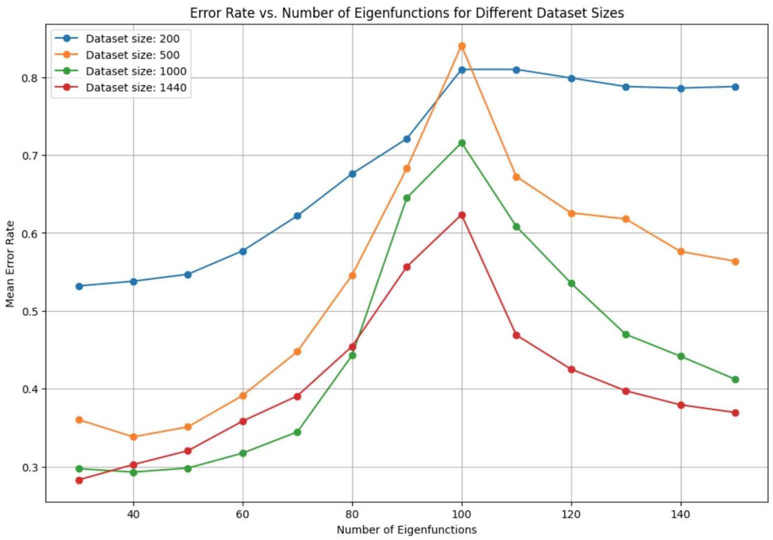
$$\mathbf{a_{classnumber}} = \left(\mathbf{E}_{\text{lab}}^{T}\mathbf{E}_{\text{lab}}\right)^{-1}\mathbf{E}_{\text{lab}}^{T}\mathbf{c_{classnumber}}$$

The Last Step:

Binary Graph Result



MNIST



COIL20

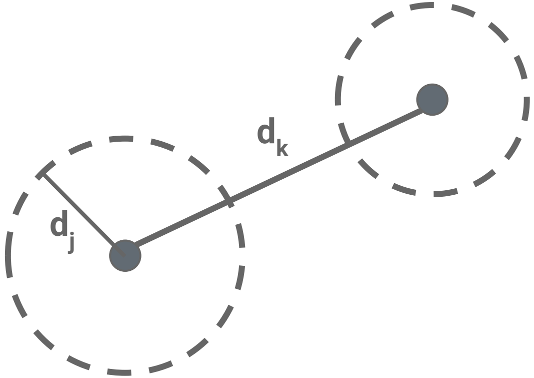
Self-tuning Weighted Matrix

Local similarity: $$W_\sigma(x,y) = h\left(\frac{\rho(x,y)^2}{\sigma}\right)$$

Normalization: $$\rho_x(z,z') = \rho_x(z,z')/\rho_x(x,x_j).$$

Self-tuning weighted matrix: $$W_\sigma(x,y) = h\left(\frac{\rho_x(x,y)\rho_y(x,y)}{\sigma}\right)$$

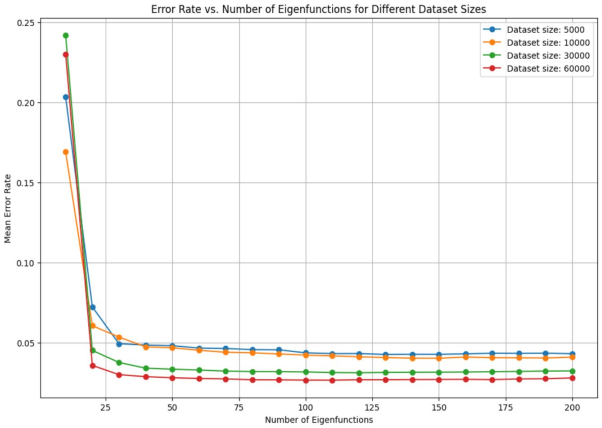
Complexity reduced: $n^2 \longrightarrow kn$

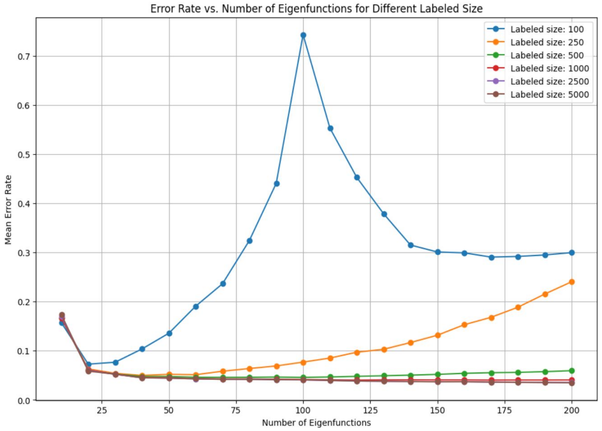


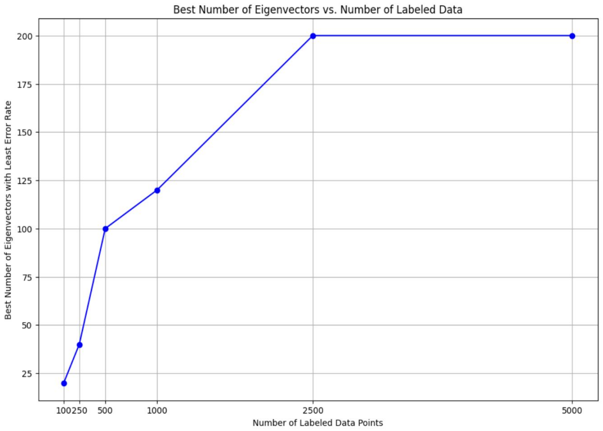
Laplacian Eigenmap-based SSL

1. Construct the **adjacency matrix** W and **degree matrix** D.($D_{ii} = \Sigma_j W_{ij}$)
2. Compute the first p eigenvectors $e_1, \ldots, e_p$ of the **Laplacian matrix** $L = W - D$, corresponding to the p **smallest** eigenvalues.
3. Represent each data point x_i in the p-dimensional Laplacian eigenmap space using $(\mathbf{e}_1(j), \ldots, \mathbf{e}_p(j))$.
4. Train a linear **classifier** using the **labeled** data point.
5. Classify the **unlabeled** data points.

MNIST Results: non-adapted(Eigenfunction)





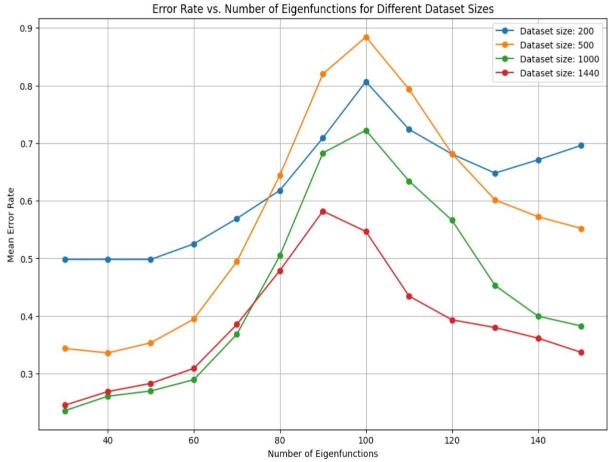


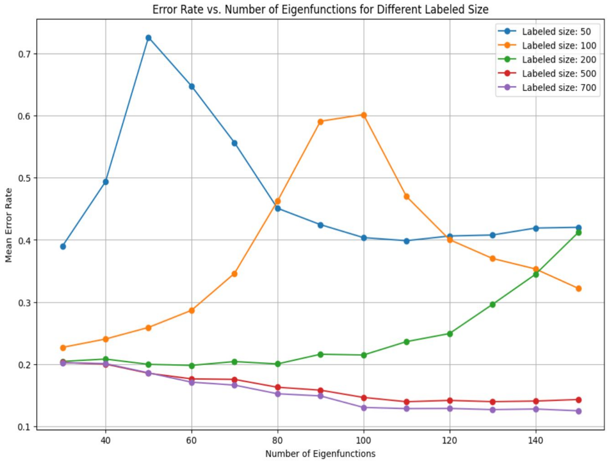
Error rate vs. number of eigenvectors with different set sizes

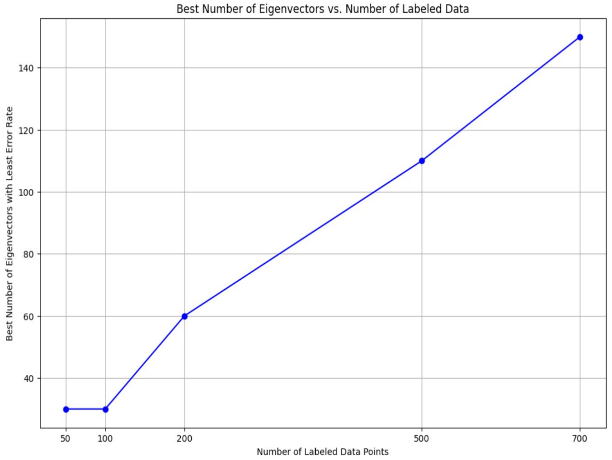
Error rate vs. number of eigenvectors with different labeled data sizes

Best number of eigenvectors with different labeled data sizes

COIL-20 Result: non-adapted







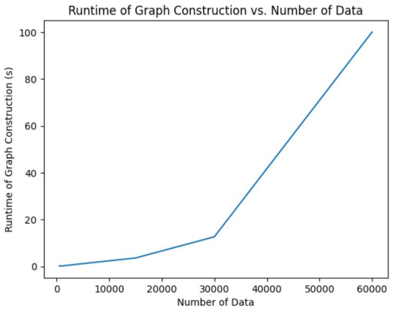
Error rate vs. number of eigenvectors with different set sizes

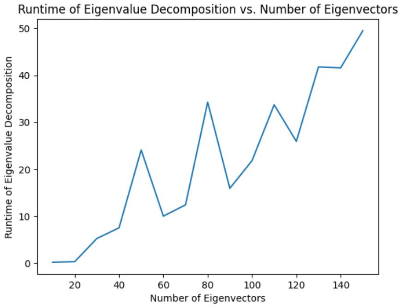
Error rate vs. number of eigenvectors with different labeled data sizes

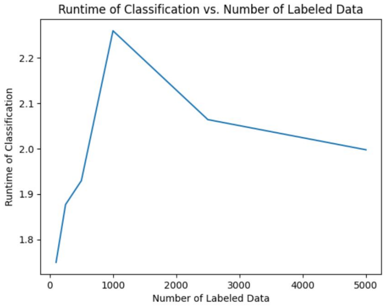
Best number of eigenvectors with different labeled data sizes

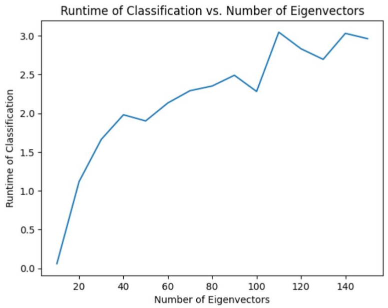
Time Complexity–non adapted

MNIST

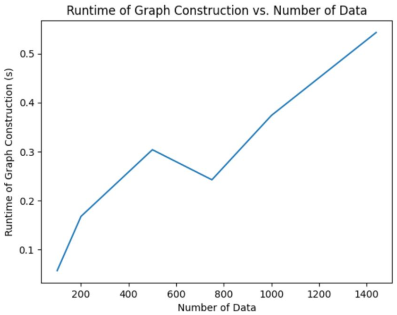


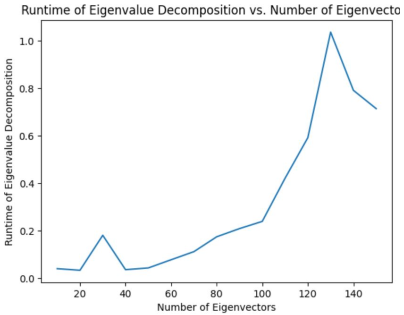


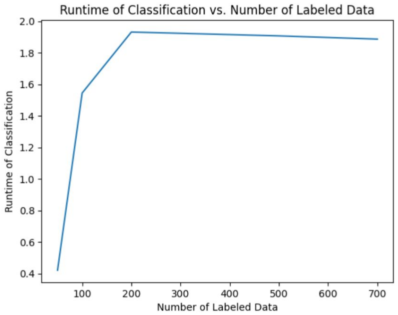


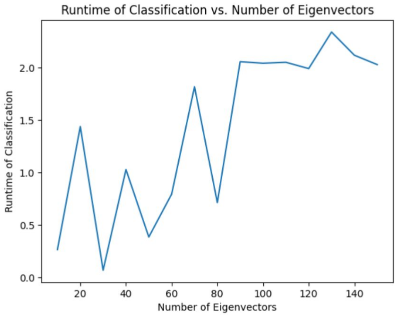


COIL20









Function-adapted Kernel

$$W_\sigma(x,y) = h\left(\frac{\rho_x(x,y)\rho_y(x,y)}{\sigma}\right)$$

$$\Rightarrow$$

$$W^f(x,y) = \exp\left(-\frac{||x-y||^2}{\sigma_1} - \frac{|\tilde{f}(x) - \tilde{f}(y)|^2}{\sigma_2}\right)$$

Physical Distance + the closeness of the estimated function values

$\boldsymbol{\sigma_1}$: control **geometric** similarity based on **distance**

$\boldsymbol{\sigma_2}$: control **functional** similarity based on function values

Strengthens connections **within** classes
maintaining class distinctions

Diffusion Process

Random Walk:

d = $\sum_{y \in V} W(x,y)$

Created normalized transition matrix to make sure the sum of probability equals one

$$K(x,y) = d^{-1}(x)W(x,y)$$

K, explain how nodes are connected, probabilities of transitioning from one node to another

Diffusion process

$$\overline{\chi_i^{lab}} = K^t \chi_i^{lab}$$

Used to smooth the function over the graph

Diffusion classifiers

Harmonic Classifier/Smoothing

Similarity

Known labels initial condition
Apply the diffusion processing

Difference

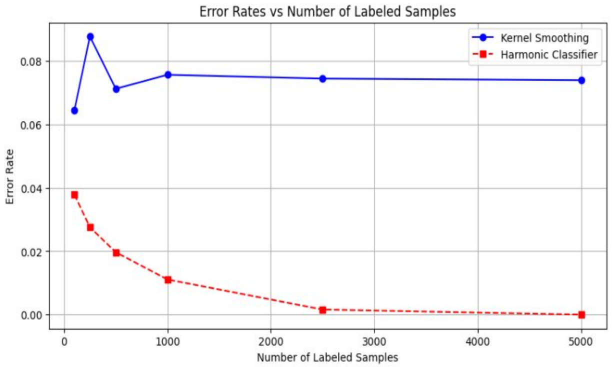
Smoothing will re-update the initial condition at each interaction

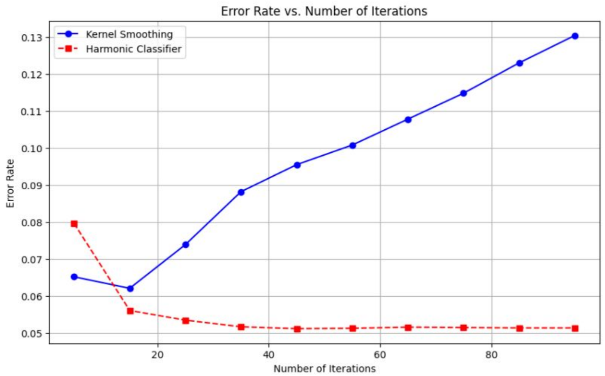
Function adaptive-

$$c_i(x) = \frac{g_i^{250}(x)}{\sum_i |g_i^{250}(x)|}$$

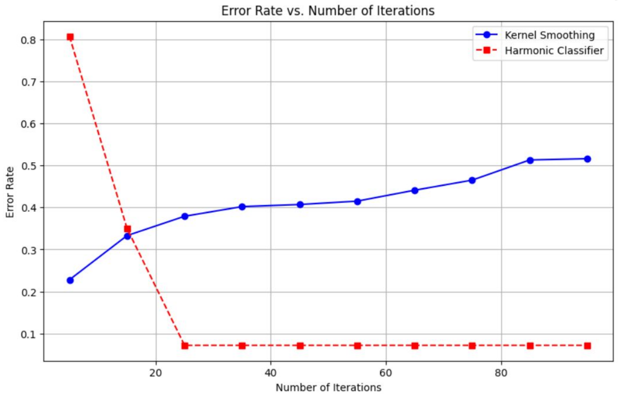
Results: non-adapted(Harmonic/Diffusion)

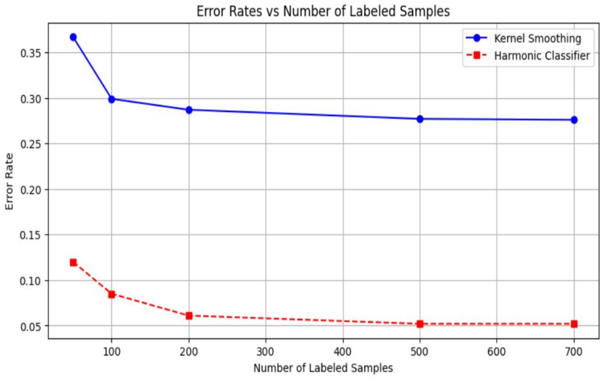
MNIST



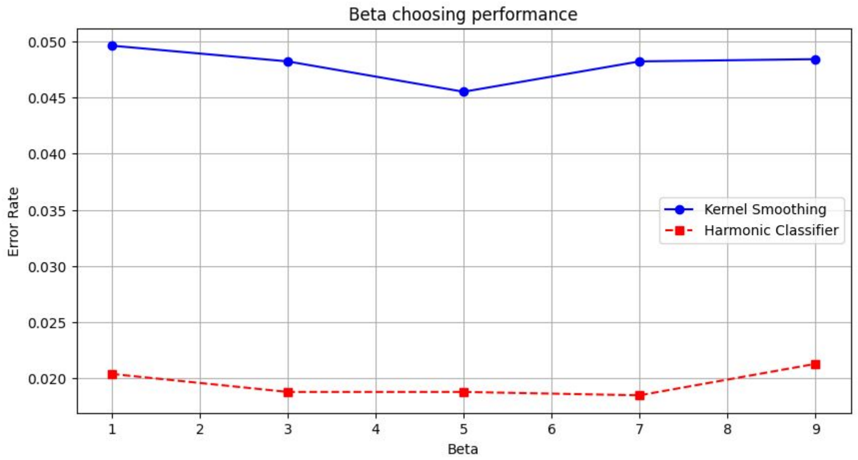


COIL20

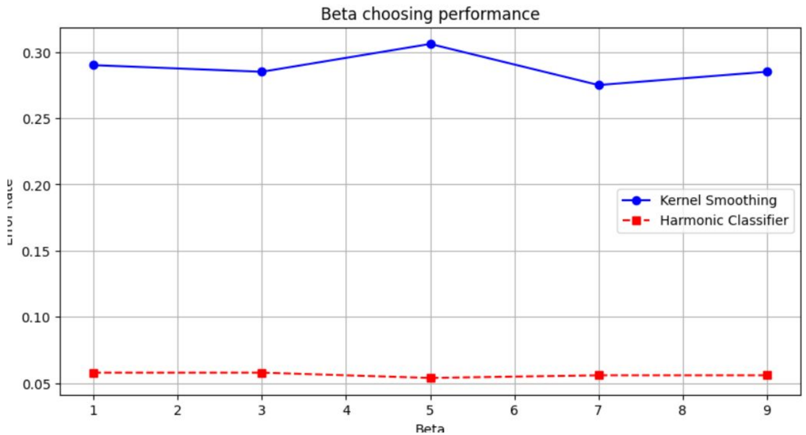




Results: Function-adapted



MNIST

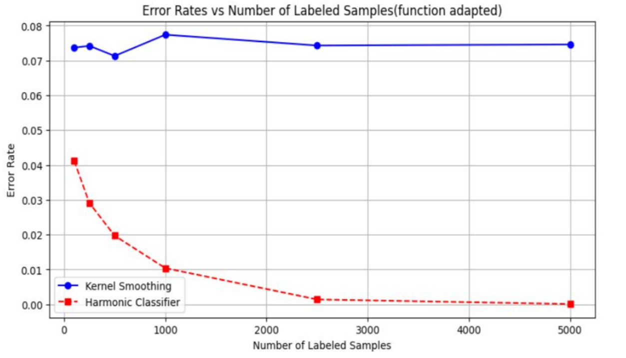


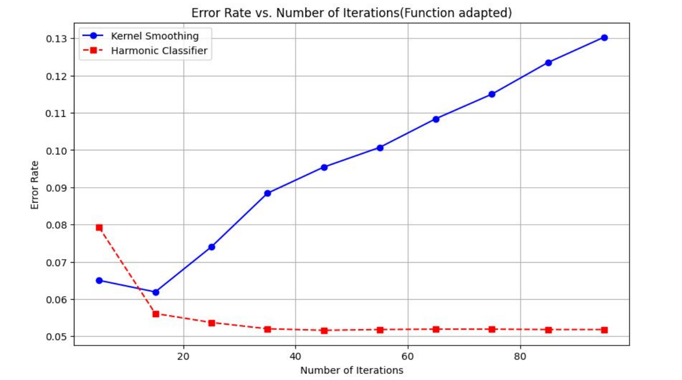
COIL20

Beta * sigma 1= sigma 2

Results: Function-adapted

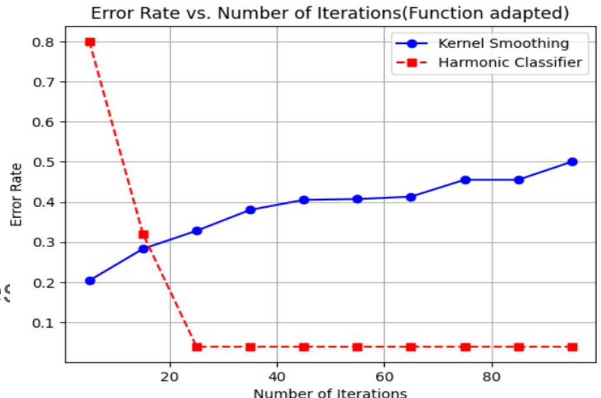
MNIST

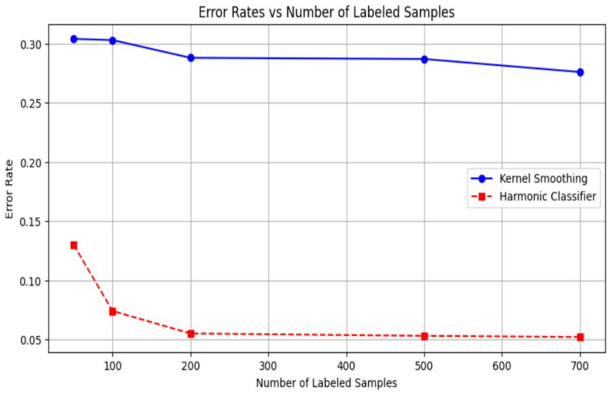




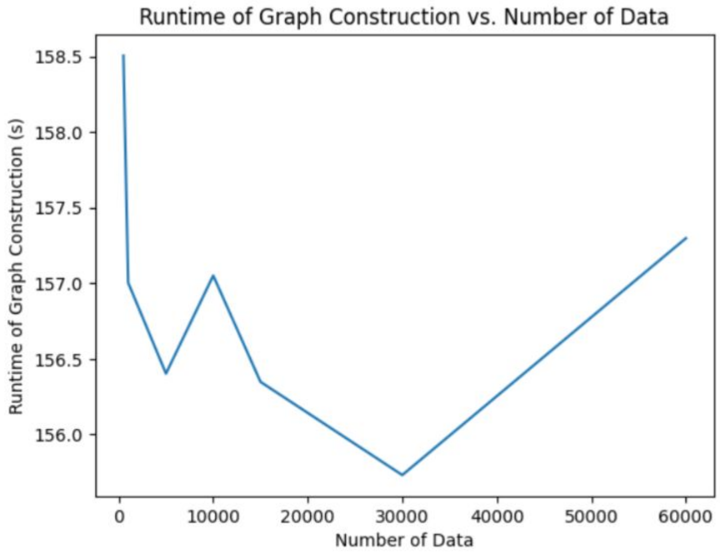
COIL20

of labeled samples
error

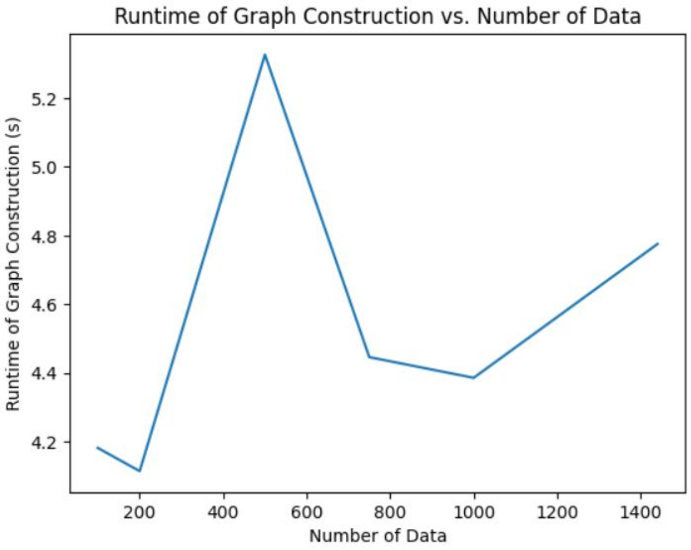




Time Complexity Function adapted



MNIST



COIL20

THANK YOU

Yun You, Wen Zhang, Feng Lu (Francis)