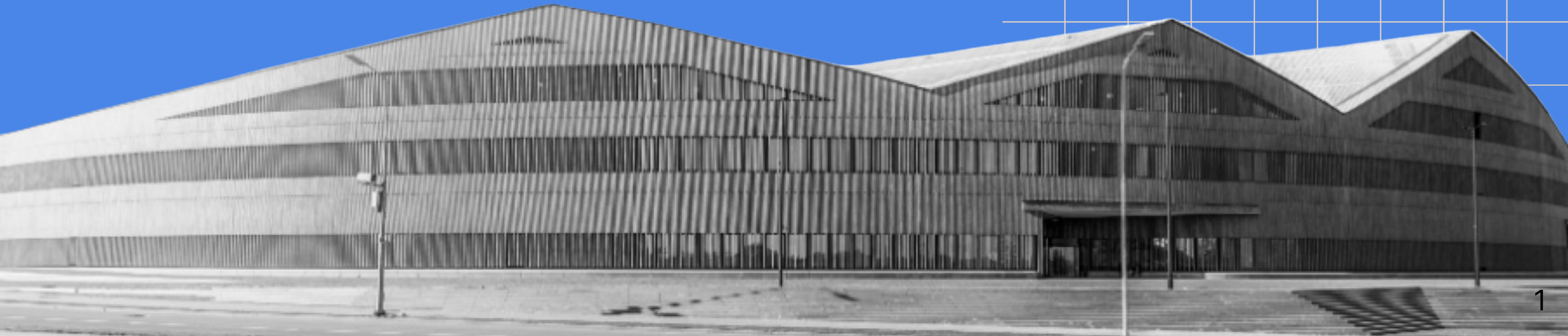
Hyperbolic Spectral Embeddings

Skoltech



Structure

1. Motivation
2. Overview
3. Problem Statement
4. Implementation
5. Findings
6. Discussion
7. Team Members
8. Link to the Repository

Motivation

Intuition. Usual graph Laplacian is symmetric nonnegative-definite matrix and it easily can be interpreted as matrix of some Euclidean metric (for connected graph we even have only one kernel vector). So we add some "negative" vertex and connect it with others by "negative" edges. So, the degree of this additional vertex is negative and for others we have some decreasing of their original degrees. We get some quasi-Laplacian matrix with signature $(V(G) - 1, 1, 1)$ for connected graph (where $V(G)$ is a number of vertices).

Overview

Definition. For graph G denote by $L(G)$ matrix of the form $V(G) \times V(G)$ with elements of the form $L(G)_{ij} = -1$ iff vertices v_i and v_j share edge (otherwise $L(G)_{ij} = 0$) and $L(G)_{ii} = \deg v_i$. By *hyperbolic Laplacian* $HypLap(G)_{\alpha_1, \alpha_1, \ldots, \alpha_V}$ for positive α_i's with $\sum_i \alpha_i = 1$ we mean matrix of the form $(V(G)+1) \times (V(G)+1)$ and with entries like

$$HypLap(G)_{\alpha_1, \alpha_2, \ldots, \alpha_V} = \begin{pmatrix} -1 & \alpha_1 & \ldots & \alpha_i & \ldots & \alpha_V \\ \alpha_1 & L(G)_{11} - \alpha_1 & \ldots & L(G)_{1i} & \ldots & L(G)_{1V} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \alpha_i & L(G)_{i1} & \ldots & L(G)_{ii} - \alpha_i & \ldots & L(G)_{iV} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \alpha_V & L(G)_{V1} & \ldots & L(G)_{Vi} & \ldots & L(G)_{VV} - \alpha_V, \end{pmatrix}$$

where $v = V(G)$

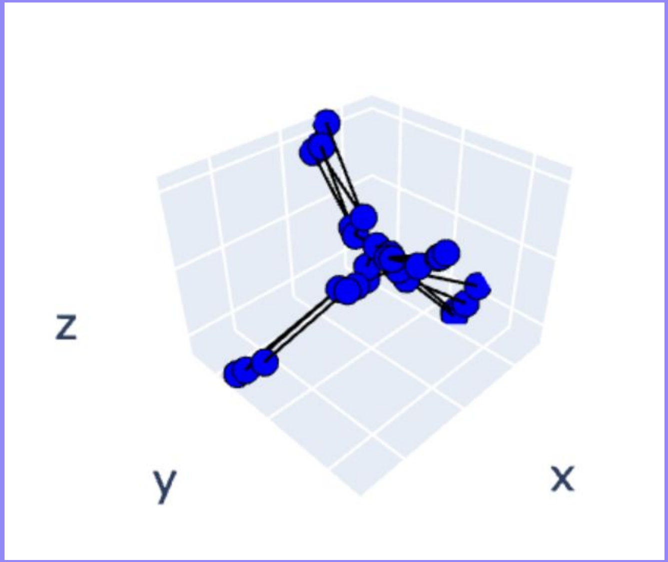
Remark. Weights α_i may depend on properties of corresponding vertices v_i. For our first research purposes we will assume equal weights $\alpha_1 = \ldots = \alpha_V = k$ (however, some experiments with adjusting of weights is also interesting).

Problem Statement

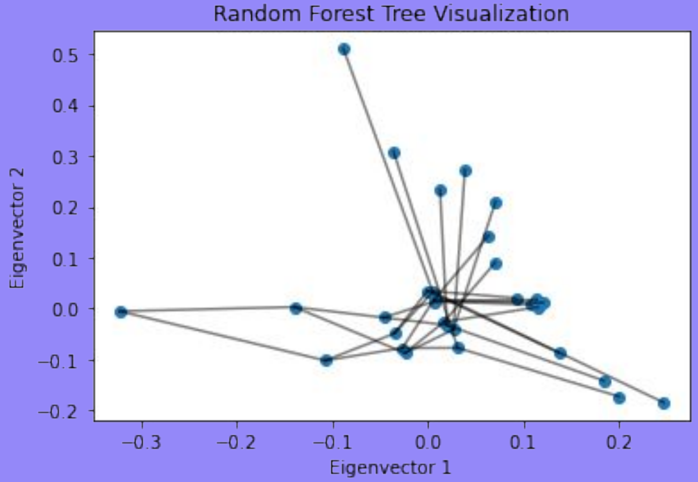
The hypothesis that projecting binary graphs into a suitable two-dimensional subspace in hyperbolic space results in minimal or no edge crossings has significant theoretical implications.

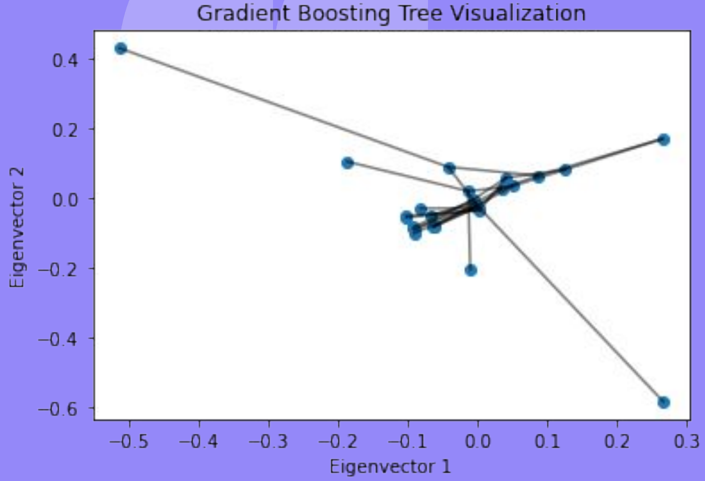
Implementation

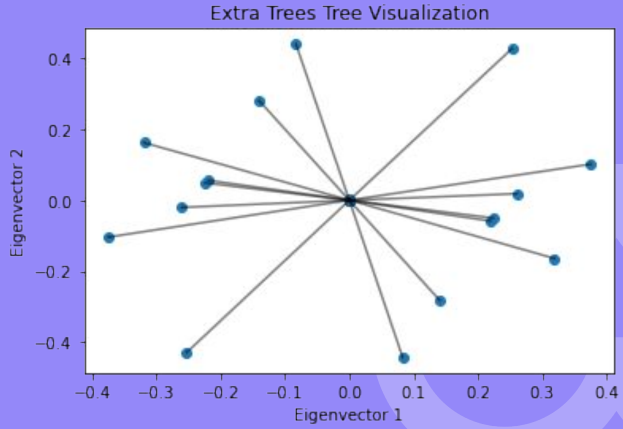
Findings



Random Forest Embedding projection in 3d


Random Forest Tree Visualization
Eigenvector 1
Eigenvector 2


Gradient Boosting Tree Visualization
Eigenvector 1
Eigenvector 2


Extra Trees Tree Visualization
Eigenvector 1
Eigenvector 2

Discussion

TBD by 21 Dec.

Team Members

Oleg Malchenko
Theoretical support, presentation

Iaroslav Gusev
Embedding of a decision tree

Jingtao Xu
Implementing projections and embeddings of other tree types

Fedor Belolutskiy
Improving efficiency of the code

Link to the GitHub repo