Graph Filtration Kernels

Till H. Schulz [1] Pascal Welke [1] Stefan Wrobel [1,2,3]

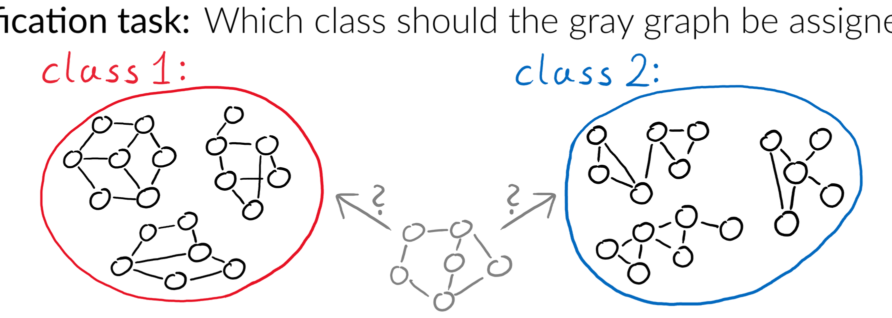
[1]Universität Bonn [2]Fraunhofer IAIS [3]Fraunhofer Center for Machine Learning

Background

What are graph kernels?

Graph kernels are functions defining similarities between graphs. They allow for applications of machine learning methods such as Support Vector Machines. One of the most successful graph classification methods relies on graph kernels.

Graph classification task: Which class should the gray graph be assigned to?



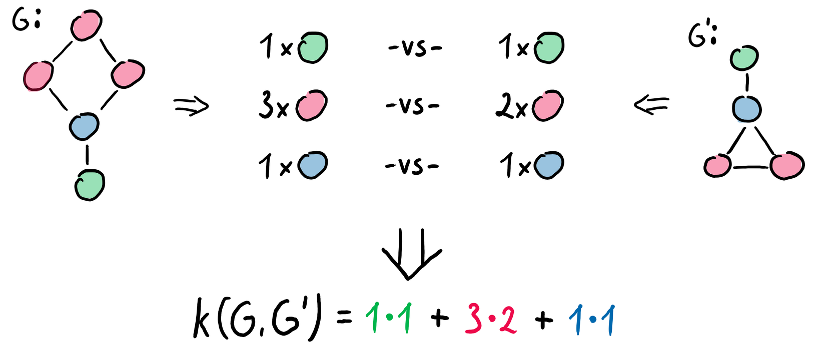
Traditional graph kernels

The majority of traditional graph kernels are based on Haussler's $\mathcal{R}$-convolution kernel and define graph similarity by comparing counts of mutual features. They are of the form

$$k(G, G') = \sum_{f \in \mathcal{F}} \text{count}(G, f) \cdot \text{count}(G', f)$$

with feature domain $\mathcal{F}$, and count(G, f) denoting the frequency of feature f in G. Thus, they compute the dot product between explicit feature vectors.

Example: Below, features correspond to node degrees where green is degree 1, red is degree 2, and blue is degree 3.



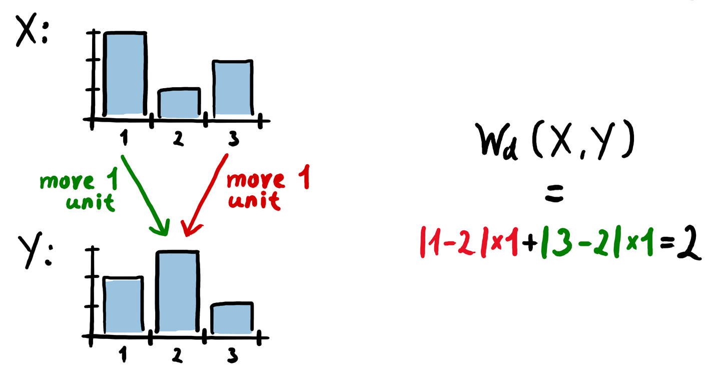
Optimal transport distance

The optimal transport distance is a distance function between probability distributions based on the concept of optimal mass transportation. Intuitively speaking, it can be viewed as the minimum effort necessary to transform one pile of earth into another.

Whereas, the optimal transport distance has cubic complexity in general, its complexity is linear for the 1-dimensional ground distance. The ground distance defines the cost for shifting mass from one point to another.

More formally, for distributions X and Y of equal mass and a ground distance d defining pairwise distances between entries of X and Y, the optimal transport distance is denoted by $\mathcal{W}_d(X, Y)$.

Example: Below, the cost for moving mass from index i to index j is equal to their absolute difference, i.e., $d(i, j) = |i - j|$. As the displayed transport plan is optimal, the optimal transport distance is simply the sum of the red and green colored costs.



The Graph Filtration kernel - A concept overview

The Graph Filtration kernel is a graph similarity measure which considers graphs at multiple granularities. This is achieved by comparing feature occurrence distributions over sequences of such graph resolutions.

From graph filtrations to kernels:

Graph filtrations — We view graphs at increasing resolution levels by considering only subsets of all edges according to their relevance.

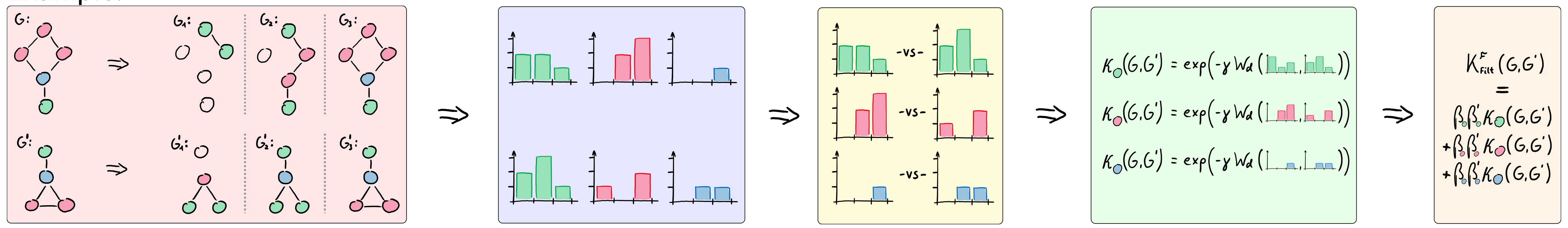
→ **Filtration histograms** — The number of feature occurences over the course of filtration sequences is tracked and expressed in form of feature distribution histograms.

→ **Histogram distance** — For each feature, the feature distribution histograms are compared using optimal transport concepts.

→ **Base kernels** — The histogram distance is "transformed" into a kernel function which compares graphs w.r.t. a specific feature.

→ **Graph Filtration kernel** — The Filtration Graph kernel is a linear combination of base kernels. It compares graphs based on different resolutions of graphs.

Example:



Graph filtrations

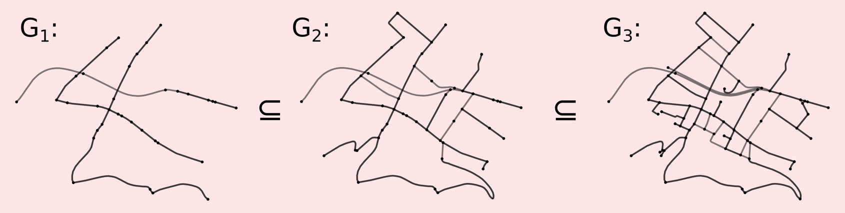
Graph filtrations view graphs at different resolutions. A graph filtration is a nested sequence of subgraphs which describes how a graph is constructed by gradually adding sets of edges.

Formally, for a weighted graph $G = (V, E, w)$, a filtration $\mathcal{A}(G)$ is a sequence

$$G_1 \subseteq G_2 \subseteq \ldots \subseteq G_k = G$$

where subgraph $G_i = (V, E_i, w)$ contains only edges exceeding threshold value α_i, i.e., $E_i = \{e : w(e) \geq \alpha_i\}$. Thus, filtration function $\mathcal{A}$ is determined by values $\{\alpha_1, \ldots, \alpha_k\}$.

Example: In street maps, it is often useful to consider subgraphs containing only roads of specific relevance. Such subgraphs highlight crucial infrastructure.

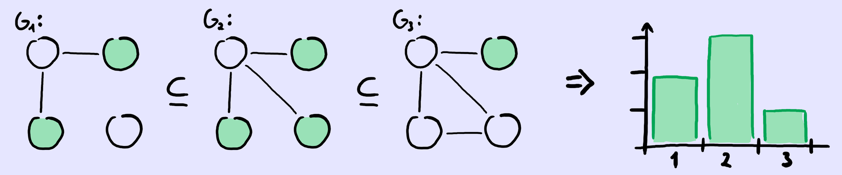


Filtration histograms

Graph filtration histograms record the number of feature occurences over filtration graphs. For every feature, a histogram displays the counts of features that appear in each filtration graph.

Formally: Given a graph G together with a length-k filtration $\mathcal{A}(G)$ and a feature f, the function $\phi_f^{\mathcal{A}} : G \to \mathbb{R}^k$ maps G to its *filtration histogram*.

Example: The highlighted feature corresponds to vertices with degree one. It is counted across all filtration graphs. This information is stored in a histogram.



Histogram distance

The Graph Filtration kernel compares feature distributions. This comparison is done by computing the optimal transport distance between filtration histograms. Roughly speaking, the optimal transport distance is the minimum cost necessary to transform one histogram into another. Since we would like to compare feature occurrences in a sequence, the ground distance needs to be 1-dimensional. This ground distance describes the cost for shifting mass from one point in the histogram to another.

The filtration histogram distance between the feature-f histograms $\phi_f(G)$ and $\phi_f(G')$ is given by the optimal transport distance $\mathcal{W}_d(\phi_f(G), \phi_f(G'))$ employing the 1-dimensional ground distance $d(\alpha_i, \alpha_j) = |\alpha_i - \alpha_j|$.

Base kernels

The filtration histogram distance gives rise to proper kernel functions. This is achieved by "transforming" the distance measure into a similarity, that is, a kernel. Such a kernel $\kappa_f(G, G')$ compares graphs G and G' w.r.t. their feature distributions of feature f over graph filtrations $\mathcal{A}(G)$, resp. $\mathcal{A}(G')$.

These so-called base kernels are of the form

$$\kappa_f(G, G') = e^{-\gamma \mathcal{W}_d(\phi_f(G), \phi_f(G'))}.$$

Graph Filtration kernel

The final Graph Filtration kernel is a linear combination of base kernels. Each such base kernel is concerned with a single feature $f \in \mathcal{F}$. Hence, an aggregation of base kernels yields a graph similarity over all considered features in $\mathcal{F}$.

The Filtration Graph kernel is defined as

$$K_{\text{Filt}}^{\mathcal{F}}(G, G') = \sum_{f \in \mathcal{F}} \beta \beta' \kappa_f(G, G').$$

Details: Computing the optimal transport distance requires equal mass of histograms. Thus, a mass-normalisation is necessary as a first step. This, however, results in a loss of feature frequency information. In order to "reverse" this disadvantage, each $\kappa_f(G, G')$ is weighted by the original histogram masses $\beta = ||\phi_f(G)||_1$ and $\beta' = ||\phi_f(G')||_1$.

The Weisfeiler-Lehman Filtration kernel

The Weisfeiler-Lehman Filtration kernel is an instance of the Graph Filtration kernel. It employs the well-known Weisfeiler-Lehman features and, hence, compares graphs based on the Weisfeiler-Lehman feature distribution over graph filtrations.

Weisfeiler-Lehman features are generated by an iterative node relabeling procedure which compresses a node's label and that of its neighbors into a new label.

Theoretical results for the WL Filtration kernel

For the Weisfeiler-Lehman Filtration kernel, we show results about its **linear complexity** as well as its **expressive power**.

Theorem

The Weisfeiler-Lehman filtration kernel $K_{\text{Filt}}^{\mathcal{F}_{WL}}(G, G')$ on graphs G, G' can be computed in time $O(hkm)$, where

- h is number of Weisfeiler-Lehman iterations,
- k is the length of the filtration sequence, and
- m is the number of edges in G and G'.

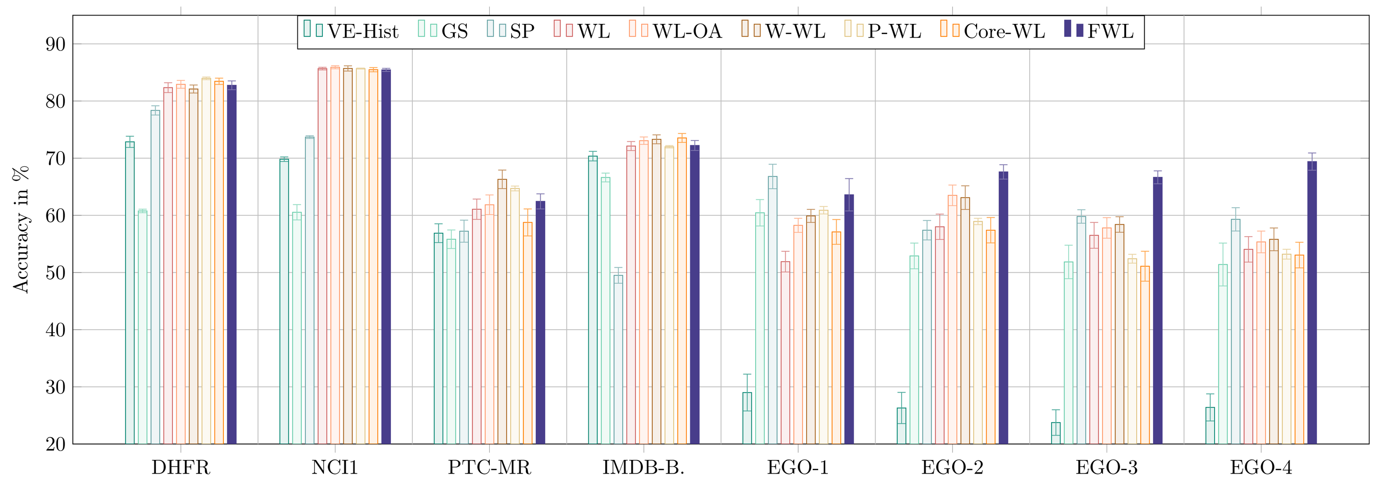
Theorem

There exists a filtration function $\mathcal{A}$ such that $\phi_f^{\mathcal{A}}(G) = \phi_f^{\mathcal{A}}(G')$ for all WL features $f \in \mathcal{F}_{WL}$ if and only if G and G' are isomorphic.

Furthermore, for such a filtration function $\mathcal{A}$, the kernel $K_{\text{Filt}}^{\mathcal{F}_{WL}}$ is *complete*, i.e., it can differentiate between *all* non-isomorphic graphs.

Experimental evaluation of the WL Filtration kernel

The Weisfeiler-Lehman Filtration kernel significantly outperforms other graph classification methods on several real-world benchmark datasets.



For the EGO datasets, filtrations of length at most $k = 3$ are sufficient to obtain overall top performing results.