Medical Data Science
University of Tuebingen
WS 2023/2024
Atahan Özer - 6317973
Karahan Sarıtaş - 6661689

Assignment III

Problem 1

(a) In a medical context, the concept of *dual-task learning* entails treating both the source domain and the target domain as individual tasks. Unique models are trained for each task, and they are linked under the assumption that these tasks share similarities, therefore the learned weight vectors shouldn't differ too much. In *multi-task learning*, pairwise distances between each pair of vectors are regularized. In theory, with more source domains available for training, we can anticipate better performance from multi-task learning on the target domain, as long as these domains exhibit similarity to each other – thus, our assumption stands. Conversely, the major benefit of dual-task learning lies in its simplicity compared to multitask learning. This makes it a preferred choice, especially when concentrating on just two tasks that are highly similar. If we are aware that the target domain closely resembles a particular source domain in our possession, it might be more advantageous to employ dual-task learning specifically with those domains.

(b) MHC is a set of cell surface proteins essential for the acquired immune system to recognize foreign molecules in vertebrates, which in turn determines histocompatibility. In humans, MHC (major histocompatibility complex) is also called human leukocyte antigen (HLA) molecule. The primary distinction between the two categories is that MHC (major histocompatibility complex) is commonly present in vertebrates, whereas HLA (human leukocyte antigen) is exclusive to humans. [1] *MHC I* plays a crucial role in presenting peptide fragments derived from intracellular proteins to cytotoxic T-cells, provoking an immediate immune response against a specific foreign antigen.

(c) Leveraging is the encoding of the data to enable data sharing. Heckerman *et al.* [2] were able to leverage data across all HLA alleles and/or their supertypes, automatically discerning what information to share and how to adeptly combine allele-specific, supertype-specific, and global information in a principled way.

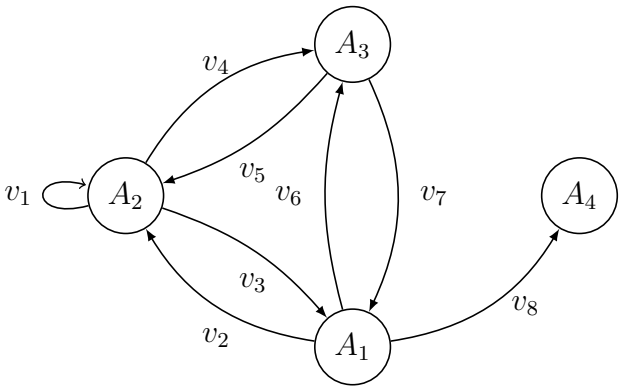
(d) In the biomedical field, we encounter graphs across various domains, including protein 3D structures, protein-protein interaction networks, drug-target interactions, and gene prioritization. Identifying similarities among graphs involves recognizing commonalities among the entities they depict, whether it's proteins or knowledge networks. This enables us to extrapolate information to other entities with analogous graph structures, assuming comparable features. As a specific example, knowledge graphs can be employed to investigate associations between genes and diseases.

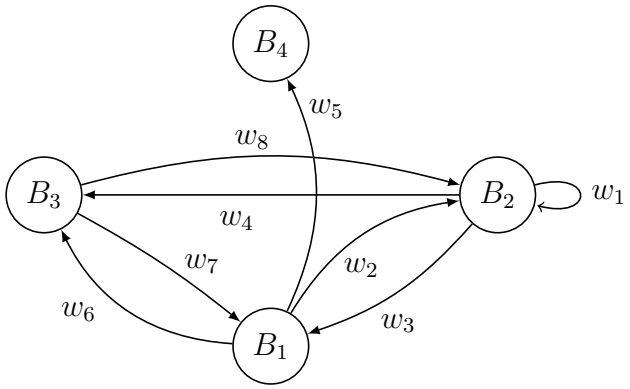
In such networks, the nodes represent proteins (or genes), and connections are established between them based on existing interactions. To facilitate gene discovery, additional information on disease associations is incorporated into these networks. In such a network, clustering can be beneficial for gaining insights into which proteins are more likely to contribute to specific types of diseases. Metabolic pathways can also be represented with graphs to extract phylogenetic trees using similarity [3].

(e) Identifying similarities between graphs involves an exponential operation based on the number of nodes in the graph, making a brute-force approach necessary. Subgraph isomorphism, the subgraph isomorphism problem involves evaluating two input graphs, G and H, to ascertain whether G encompasses a subgraph that is *isomorphic* to H. This problem is known to be NP and it can be reduced to the Hamiltonian Cycle problem in polynomial time, which means it is an NP-complete problem. Currently, this problem is addressed through various techniques that aim to optimize graph similarity computations using heuristics and certain assumptions. Two possibilities for comparing two graphs can be utilizing graph kernel methods and graph edit distance. In simple terms, graph kernels assess computable substructures of graphs—like walks—within polynomial time, while Graph Edit Distance (GED) is a well-known technique in the field of Graph Matching used to calculate the dissimilarity between two graphs. GED represents the cost of the best set of edit operations needed to transform one graph into another [4]. Efficient approximate methods for Graph Edit Distance (GED) exist to compute it within a reasonable time frame, as discussed in [5]. While the performance of these approaches may vary depending on the specific case, it is reasonable to anticipate that GED surpasses graph kernels. This is because GED examines the entire graph to calculate the number of deletion/insertion operations, unlike kernels which rely on sub-properties.

(f) Two graphs G_1 and G_2 are isomorphic if there exists a matching between their vertices so that two vertices are connected by an edge in G_1 if and only if corresponding vertices are connected by an edge in G_2. To put it into layman's words, if we change the name of the nodes in the graph G_1 to graph G_2 carefully, we can get G_2 with corresponding edges between the vertices.

Let's examine the following two isomorphic graphs A and B:





By simply changing each A_i to B_i and v_i to w_i, it can be seen that these graphs are isomorphic to each other. (Actually, I created the second graph by simply tweaking the names and rearranging the positions of the nodes in the image.)

(g) In the most formal sense, graphlets are induced subgraph isomorphism classes in a graph, two graphlet occurrences are isomorphic, whereas two graphlets are non-isomorphic [6]. In order to create a graphlet from a given graph G, we should select a subset of vertices of the graph G and all edges connecting pairs of vertices in that subset [7]. For similarity, we can assert that both graphlet kernels and previously discussed kernels are non-negative, real-valued, integrable functions, satisfying the conditions of symmetry and normalization across the entire range. As a difference, graphlet kernels necessitate the generation of graphlets to compute the k-spectrums of the graph. To expedite this process, random sampling or a specialized technique for bounded degree graphs is employed [8]. In contrast, the preceding kernels did not require such approximations. Yet, when it comes to execution time, graphlet kernels outshine graph kernels, primarily because graph kernels struggle to scale effectively with larger graphs.

(h) Shortest path kernel [9]: In the particular case of graphs with unweighted edges, let us consider the base kernel k_{SP} of the form $k_{SP}\ (G, G') =< \phi_{SP}(G), \phi_{SP}(G') >$, where $\phi_{SP}\ (G)$ (resp. $_{SP}(G')$) is a vector whose components are numbers of occurrences of triplets of the form (a, b, p) in G (resp. G'), where a, b are ordered start and end node labels of the shortest path and $p \in N_0$ is the shortest path length. Then:

$$k^{(h)}_{WL\ \text{shortest path}} = k_{SP}(G_0, G'_0) + k_{SP}(G_1, G'_1) + ... + k_{SP}(G_h, G'_h).$$

Random walk kernel [10]: When presented with a pair of graphs, execute random walks on each, and quantify the occurrences of congruent walks. This fundamental concept serves as the underlying principle for both random walk and marginalized graph kernels. To elaborate on this, the introduction of direct product graphs is a prerequisite. As illustrated by the findings presented in the slides (Multitask and Graph Kernels pg. 53), the shortest path consistently outperforms random walk across all the datasets in terms of prediction accuracy.

Problem 3

1. In the context of undirected graphs, the degree of a vertex is defined as the cardinality of the set of edges incident upon that vertex.

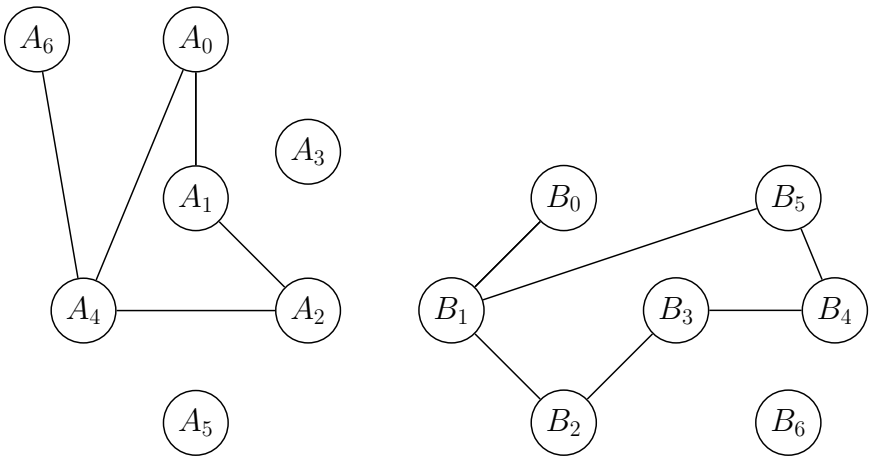
Vertex	A_0	A_1	A_2	A_3	A_4	A_5	A_6
Degree	2	2	2	0	3	0	1

Table 1: Vertex degrees in Graph A

Vertex	B_0	B_1	B_2	B_3	B_4	B_5	B_6
Degree	1	3	2	2	2	2	0

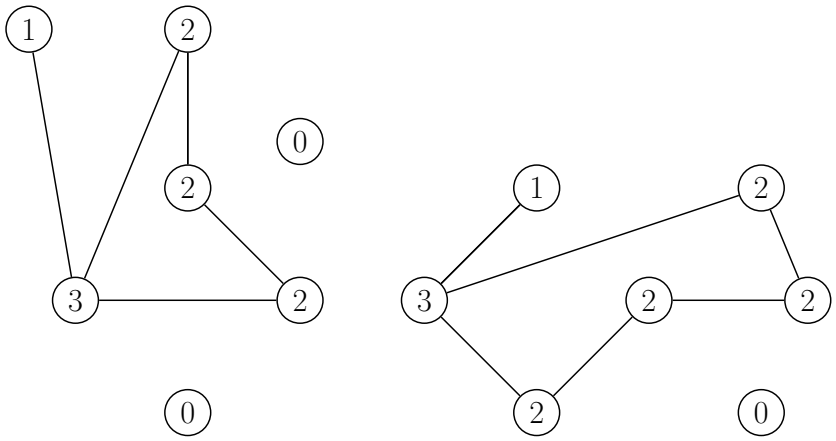
Table 2: Vertex degrees in Graph B

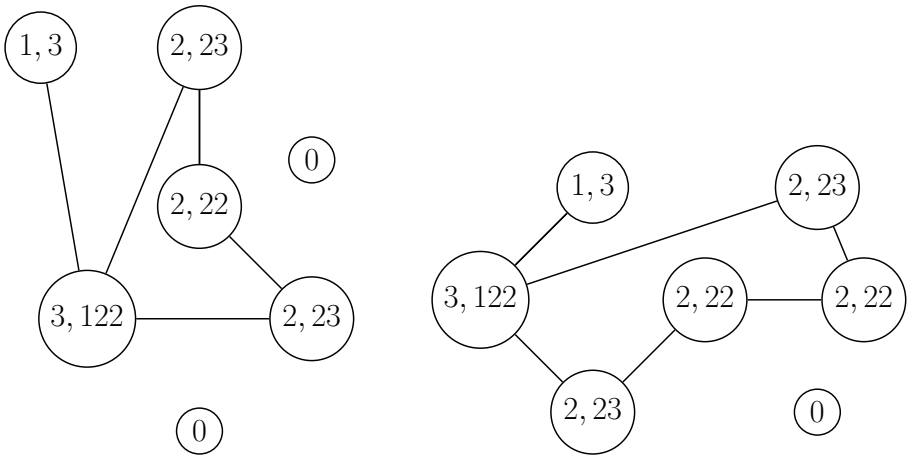
G_1 (A) and G_2 (B) can be visualized as follows:



2. Application of the one-dimensional Weisfeiler-Lehman test for graph isomorphism is as follows:

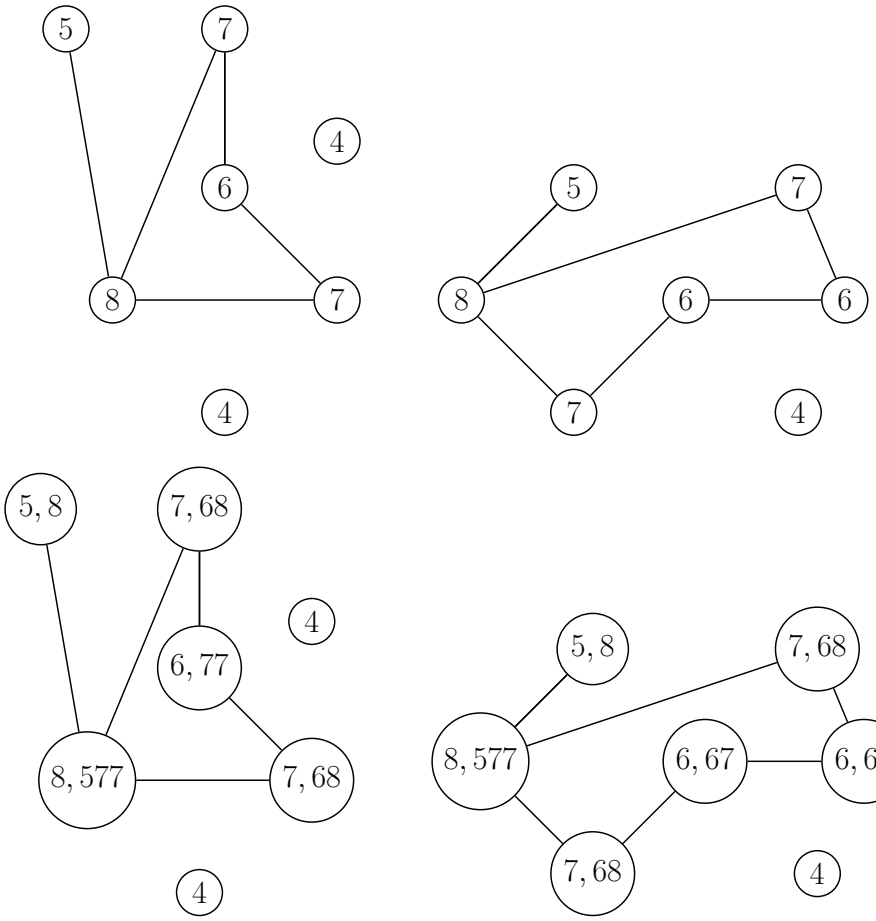
   (a) Multi-set label determination and sorting:



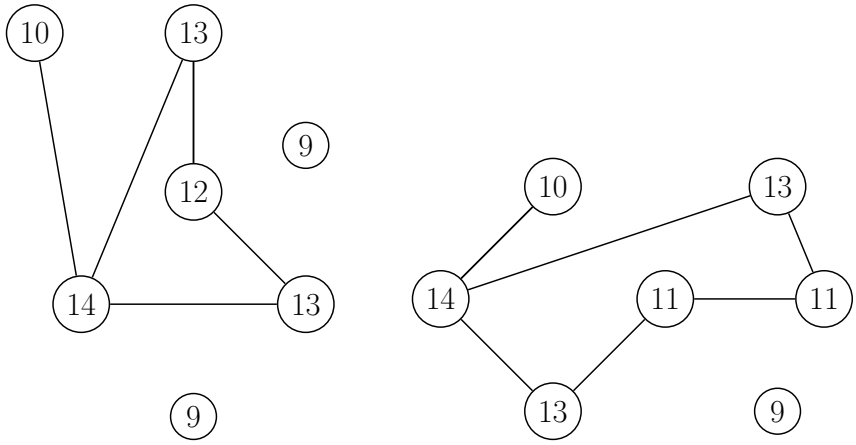


(b) Label compression and relabelling:

0	4
1, 3	5
2, 22	6
2, 23	7
3, 122	8

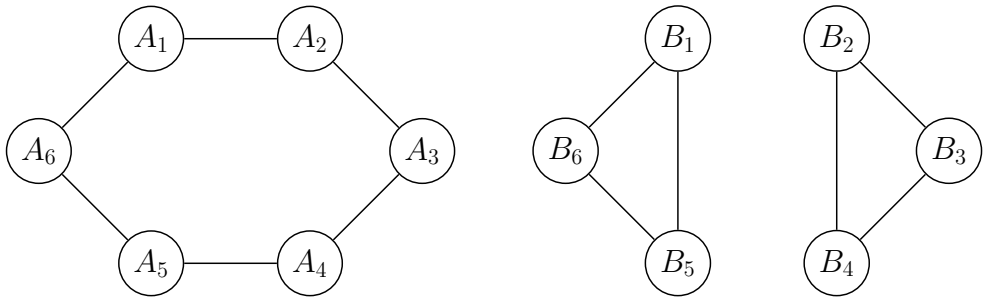


$$
\begin{aligned}
4 &\longrightarrow 9 \\
5, 8 &\longrightarrow 10 \\
6, 67 &\longrightarrow 11 \\
6, 77 &\longrightarrow 12 \\
7, 68 &\longrightarrow 13 \\
8, 577 &\longrightarrow 14
\end{aligned}
$$



Given that the label set of graph A differs from that of graph B (specifically, the label 12 is absent in B), we can deduce that these graphs are **not isomorphic**.

3. With the Weisfeiler-Lehman test, our aim is to assess isomorphism based on degrees. However, this approach makes the test susceptible to failure in straightforward cases. For instance, it fails to differentiate between two regular graphs with identical node counts and degrees, even when one is connected and the other isn't:



In this example, following the initial iteration, none of the nodes switch groups, prompting us to halt the iteration. Unfortunately, our algorithm fails to correctly classify them as non-isomorphic, resulting in a false positive.

References

[1] Anaya JM, Shoenfeld Y, Rojas-Villarraga A, *et al.*, editors. Bogota (Colombia): El Rosario University Press; 2013 Jul 18.

[2] Heckerman, David & Kadie, Carl & Listgarten, Jennifer. (2006). Leveraging Information Across HLA Alleles/Supertypes Improves Epitope Prediction. 296-308.

[3] Heymans, M., & Singh, A. (2003). Deriving phylogenetic trees from the similarity analysis of metabolic pathways. Bioinformatics, 19 Suppl 1, i138-46.

[4] Sanfeliu, Alberto; Fu, King-Sun (1983). "A distance measure between attributed relational graphs for pattern recognition". IEEE Transactions on Systems, Man, and Cybernetics. 13 (3): 353–363. doi:10.1109/TSMC.1983.6313167.

[5] Adel Dabah, Ibrahim Chegrane, Saïd Yahiaoui, Efficient approximate approach for graph edit distance problem, Pattern Recognition Letters, Volume 151,2021, Pages 310-316, ISSN 0167-8655, https://doi.org/10.1016/j.patrec.2021.08.027. (https://www.sciencedirect.com/science/article/pii/S0167865521003251)

[6] Graphlets. (2023, November 21). In Wikipedia. (https://en.wikipedia.org/wiki/Graphlets)

[7] Induced Subgraphs. (2023, November 21). In Wikipedia. (https://en.wikipedia.org/wiki/Induced_subgraph)

[8] Shervashidze, N., Vishwanathan, S., Petri, T., Mehlhorn, K., Borgwardt, K. (2009). Efficient graphlet kernels for large graph comparison. *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, in *Proceedings of Machine Learning Research* 5:488-495 Available from https://proceedings.mlr.press/v5/shervashidze09a.html.

[9] Shervashidze, N., Schweitzer, P., van Leeuwen, E. J., Mehlhorn, K. & Borgwardt, K. M. (2011). Weisfeiler-Lehman Graph Kernels. J. Mach. Learn. Res., 12, 2539-2561.

[10] Vishwanathan, S. V. N., Schraudolph, N. N., Kondor, R., & Borgwardt, K. M. (2010). Graph kernels. Journal of Machine Learning Research, 11, 1201-1242.