Survey Paper - Optimal Transport

Seminar Selected Papers from NeuroIPS 2019 - SS 2020

Fabrice beaumont

Rheinische Friedrich-Wilhelms-Universität Bonn

s6fabeau@uni-bonn.de

31.08.2020

I. Introduction

In this report three scientific papers, which all were presented at the *33rd Conference on Neural Information Processing Systems* (NeurIPS 2019 in Vancouver, Canada) are being summarized and compared. The papers are:

1. "*Wasserstein Weisfeiler-Lehman Graph Kernels*" by Togninalli, Llinares-López, Ghisu, Rieck and Borgwardt (ETH Zurich and the Swiss Institute of Bioinformatics) [1]
2. "*GOT: An Optimal Transport Framework for Graph Comparison*" by Maretic, Chierchia, EL Gheche and Frossard (Ecole Polytechnique Fédérale de Lausanne) [2]
3. "*A Graph Theoretic Additive Approximation of Optimal Transport*" by Lahn, Mulchandani and Raghvendra (Virginia Tech) [3]

All three papers relate to the topic of Optimal Transport and the Wasserstein distance (WD) in some way. The papers will be presented by summarizing the underlying motivation, the presented method and the achieved results. On top of that comments on the papers as a whole are made.
Subsequently the three papers are compared and related to each other.
If not marked differently, all explanations related to the topic originate from the three papers at hand.

Two of the papers ([1] and [2]) use similar but slightly different definitions of graphs. To avoid redundancy a single detailed definition which suffices for both papers is now prepended:
An *undirected graph* $G = (V, E)$ is a tuple of a set of nodes V and a set of edges $E \subseteq \{(v, w)| \ v, w \in V\}$. The set $N(v) := \{u \in V| \ \{u, v\} \in E\}$ is called the *neighborhood* of a node $v \in V$. Edges $e \in E$ can have weights which are defined by a *weight function* $w : E \to \mathbb{R}$. Nodes $v \in V$ can have labels which are defined by a *label function* $l : E \to \Sigma$. If Σ is a finite set the labels are called *categorical* (or discrete). Otherwise it is $\Sigma = \mathbb{R}$

and the labels are called *continuous*[1]. Note that in general multi-dimensional labels are possible as well.
By subtracting the diagonal degree matrix from the adjacency matrix one can compute the *Laplacian matrix* which will be denoted with L. The notations ν_G and μ_G refer to the $|V|$ dimensional vector containing the labels of all nodes in V (with respect to some given order of the nodes and a given label function).

In this report graphs are considered to be connected. The formal definitions of connected graphs, subgraphs, paths and complete bipartite graphs will not be formalized further in this report but instead can be found for example in [4].

Since all three papers do related to the topic of optimal transport and the WD, a quick introduction to this topic is presented in the following section.

II. Optimal Transport

The topic of Optimal Transport was formulated by Monge and Kantorovich. In general it is concerned with computing a transportation strategy to transform a given data set (supply) into another desired one (demand)[2]. WDs are suitable to compute the effort needed for such transformations. Recently they are used in more and more applications in the fields of Machine Learning and Artificial Intelligence.
The task of determining an optimal transportation plan yields in comparing transportation costs, which in turn induces a similarity measure between the given distributions.

Consider a complete bipartite graph $G(A, B)$ on two distinct sets of nodes A (demand) and B (supply). Each node $b \in B$ has a supply $s_b \in \mathbb{R}$ and each node $a \in A$ has a demand $d_a \in \mathbb{R}$. A function $c : A \times B \to \mathbb{R}$ is called *cost function*. $c(a, b)$ indicates the effort to change the value of s_b to the value of d_a.
A *transportation plan* is defined as a function $\sigma : A \times B \to \mathbb{R}_{\geq 0}$ such that

$$\sum_{b \in B} \sigma(a, b) \leq d_a \quad \text{and} \quad \sum_{a \in A} \sigma(a, b) \leq s_b.$$

These conditions formalize that a transportation plan can not oversaturate a demand or create supply. A transportation plan is *maximal*, if all given supply is transported [3]
The cost $w(\sigma)$ of a transportation plan σ is defined as

$$w(\sigma) := \sum_{(a,b) \in A \times B} \sigma(a, b) c(a, b)$$

Intuitively the cost of a minimal transportation plan can be used as similarity measure between the two inputs. This idea is utilized in [1] and [2] and it could be applied to difficult problems such as the NP-hard problem of graph alignment and graph clustering.

[1] In [1] such labels are called attributes. In [2] such labels are called signals.

[2] Example of data sets are point clouds, probability distributions, images and graphs.

[3] For this purpose it is reasonable to limit the supply such that it cannot oversaturate all demands.

Applying further constraints to this general description of the problem leads to different variations. If all supplies and demands are positive integers, the problem is called the *Hitchcock-Koopmans transportation problem*. If all supplies and demands have value one, the problem is called the *assignment problem* (where the costs encode possible assignments). If all supplies sum up to the value one, the input sets can be seen as a discrete distribution. If the cost function is a metric, the optimal transport cost is referred to as the *Earth Mover's distance*.
Finally, if the cost function is the p-th power of a metric for $1 < p \in \mathbb{N}$, the optimal transport cost is referred to as the *p-Wasserstein distance* (p-WD)[4].
Note that the WD in its general form is not isometric. This means that there exists no metric-preserving mapping to an L^2-norm.

III. Paper Presentation

III.I. Paper: "*Wasserstein Weisfeiler-Lehman Graph Kernels*"

The paper "*Wasserstein Weisfeiler-Lehman Graph Kernels*" proposes a kernel to measure similarities between two graphs. The basic idea is to first represent a given graph via a matrix which contains some level of information about the substructures of the graph (Wiesfeile Lehman graph embedding). Secondly the 1-WD between these matrices of two graphs is computed and finally used as part of a kernel function which serves as similarity function.

III.I.a. Motivation

The authors motivate their research with the observation of an increasing importance of graph-structured data and the difficult problem of classification problems on such data. They state that known similarity measures between graphs often rely on global features of the graphs (e.g. aggregated by $\mathcal{R}$-convolution kernels) and do not capture small, but possibly significant, differences. Furthermore they argue that many known approaches do not scale well with increasing dimensionality of the node label function.

The authors mention related work in which the sequence of labels of a random walk or of a shortest path through the graph were used. In contrast to this the presented new method is designed to avoid simplifications and capture more detailed information about the substructure of the graph.

III.I.b. Method

The method can be sectioned into three phases which now are summarized one by one.

Weisfeiler Lehman graph embedding scheme The information about edge weights and node labels $x_G^{h+1}(v)$ is compressed by iteratively generating new node labels depending on the weighs and labels in the neighbourhood. Define $\mathcal{N}^h(v) := \{x_G^h(u) | \, u \in N(v)\}$ as the set of labels in the neighbourhood of a node v. For the computation of

[4] Some more formal definitions of the p-WD can be found in all three of the summarized papers.

such an iterative embedding two formulas are proposed. One for categorical labelled nodes (CAT) and one for node labels with continuous labels (CON):

$$x_G^{h+1}(v) := \begin{cases} \text{hash}(x_G^h(v),\ \mathcal{N}^h(v)) & \text{(CAT)} \\ \frac{1}{2}\Big(x_G^h(v) + \frac{1}{d_v}\sum_{u\in\mathcal{N}^h(v)} w(\{u,v\})x_G^h(v)\Big) & \text{(CON)} \end{cases}$$

Repeating this iterative definitions i times yields in attributes that depend on every node that can be reached in paths of length i[5]. Thus the dependency in strongly connected subgraphs is increased. A graph embedding f is defined as the matrix of all computed embeddings for every node:

$$f(G) := \begin{bmatrix} X_G^H(v_1) \\ \dots \\ X_G^H(v_{|V|}) \end{bmatrix} := \begin{bmatrix} [x_G^0(v_1) & \dots & x_G^H(v_1)] \\ & \dots & \\ [x_G^0(v_{|V|}) & \dots & x_G^H(v_{|V|})] \end{bmatrix}$$

Wasserstein distance In the second step the 1-WD between two such graph embeddings is computed.
This requires a distance matrix M, which contains distances between the vectors containing all computed labels for each pair of nodes. Again two different approaches for both the categorical and the continuous case are presented. Either the normalized Hamming distance or the Euclidean distance are used. It is:

$$d(g,g') := \begin{cases} \frac{1}{H+1}\sum\limits_{i=1}^{H+1} \delta(g,g') & \text{(CAT)} \\ \|g-g'\|_2 & \text{(CON)} \end{cases}$$

where $\delta(x,y)$ has value 1 if $x \neq 0$ and 0 otherwise.
Since in this context, the 1-WD W_1 is computed on finite sets of node embeddings and not with continuous probability distributions, a formulation as a sum could be used. For a transportation matrix P defining a transportation plan between the two given embeddings and the distance matrix D containing all ground distances between all nodes it is:

$$W_1(f(G_1), f(G_2)) = \min_P \langle P, D\rangle$$

In the implementation this distance was computed using a network simplex method.

Kernelization Finally, in the third step the 1-WD between the graph embeddings is used in an instance of a Laplacian kernel in order to yield the main result of the paper called the *WWL-kernel* K_{WWL}:

$$K_{\text{WWL}}(f(G_1), f(G_2)) := \exp\Big(-\lambda\ W_1(f(G_1), f(G_2))\Big)$$

The paper concludes the presentation of the method by discussing the positive definiteness of the WWL kernel. At the time it could only be ensured for the categorical case. In order to "ensure the theoretical and practical correctness" ([1]) an Krein SVM is used for classifications tasks in the continuous case.

[5] $x_G^0(v)$ denotes the original label of v in the graph G.

III.I.c. Results

Some experimental results on several databases (MUTAG, PTC-MR, ...) are presented to support the claim that the presented WWL kernel is competitive with the best graph kernel for categorically labelled data[6] and that it outperforms all state-of-the art graph kernels for continuous attributed graphs[7].
The positive effects of using the WD in particular is demonstrated by replacing it with a less successful RBF kernel. The applicability for both the categorical and continuous labels is seen as and advantage on top of that.

The runtime complexity using a naive simplex network implementation for solving the optimization problem lies in $\mathcal{O}(n^3 \log(n))$ where n is the number of nodes in both graphs combined. However using approximations relying on Sinkhorn regularisation are praised to reduce the computational burden to near-linear time while preserving accuracy.

For future work the authors suggest runtime improvements and theoretical results for the positive definiteness of the WWL kernel in the continuous case.

III.I.d. Critic

The paper is clearly structured using definitions claims and case distinctions. Sources of proofs are hinted on, or the proofs are attached in the appendix. This makes it easy to focus on and understand the ideas and method itself. Nonetheless it seems quite unintuitive that in the paper first the used computation method of the 1-WD is introduced (page 3, equation 2), then the distance on some graph embedding scheme is defined and only after this the actually used graph embedding scheme is revealed. Thus the authors later come back to the preemptively given definition of the 1-WD and only then continue with the keratinization. A more linear representation of the computation could be beneficial.

Some further criticism seems suitable regarding flaws, which seem to be easily avoidable. For example the column-wise notation in the 6th equation (page 5) suggests some sort of correlation between the assigned labels between the nodes. But on the same page it is explained how the just defined matrix is processed row-wise (for the computation of the ground distance).
In the same section the number of iterative embeddings is denoted with H which later is claimed to be an input parameter of the algorithm. Since depending on de graphs, H directly influences the quality of the embedding, it seems indispensable to discuss useful settings or limits of this parameters. But this was dismissed entirely.

Another rather important step in the algorithm is the actual computation of the WD. It is barely mentioned that a network simplex was used, settings or the foundations for this decision are missing, too. Similarly the use of a Laplacian kernel seems to be

[6] WL (shervashidze and Borgwardt), WL-OA (Kriege, Giscard, Wilson)

[7] Hash graph kernels (HGK-SP, HGK-WL), GraphHopper GH

considered trivial. The use of other kernel definitions is not discussed.

Finally it is stated that the use of a KSVM for the proposed method and a SVM for all other compared methods would not influence the results. Since this is a distinctly different setting for exclusively the investigated method it seems to be reasonable on thoroughly investigate the truth of this claim.

III.II. Paper: *"GOT: An Optimal Transport Framework for Graph Comparison"*

In this paper the authors propose a similarity measure between graphs which is based on the 2-WD on their labels and computed using a stochastic gradient descent method.

III.II.a. Motivation

The motivation is similar to the motivation of the first paper [1]. One major difference is, that in this research the node labels are assumed to follow a normal distribution which can be used to represent the graph.

III.II.b. Method

According to the motivation it is assumed that the labels of the nodes in two given graphs are distributed normally. More explicitly the authors assume that the labels are distributed normally with a mean of zero and a covariance given by the pseudoinverse of the Laplacian of the respected graph. Using this assumption authors argue that the 2-WD $W_2^2(\nu^{G_1}, \mu^{G_2})$ between the label vectors $\nu^{G_1} \in \mathbb{R}^{|V_1|}$ and $\mu^{G_2} \in \mathbb{R}^{|V_1|}$ of two graphs G_1 and G_2 can be computed as:

$$W_2^2(\nu^{G_1}, \mu^{G_2}) = Tr(L_1^\dagger + L_2^\dagger) - 2Tr(\sqrt{L_1^{\dagger/2} L_2^\dagger L_1^{\dagger/2}})^8$$

By definition of the Laplacian matrices L_1 and L_2 this computation depends on the used enumeration of the nodes when defining the label vectors. To get rid of this dependency a permutation matrix P is introduced and the covariance of one of the label distributions is permuted accordingly:

$$W_2^2(\nu^{G_1}, \mu_P^{G_2}) = Tr(L_1^\dagger + P^T L_2^\dagger P) - 2Tr(\sqrt{L_1^{\dagger/2} P^T L_2^\dagger P L_1^{\dagger/2}})$$

Since the goal is to obtain an unambiguous similarity measure by minimizing this distance the resulting non convex optimization problem is further simplified. First replacing the discrete permutation parameter with a continuous version provided by the Sinkhorn operator and then by focussing on optimizing the expectation (Bayesian exploration) of the problem rather than the equation itself. Lastly a multivariate normal distribution and a parameterless distribution (product of standard normal distributions) is used to obtain an approximating gradient with can be optimized used stochastic gradient descent. As usual, this approach can not guarantee convergence towards a global minimum of the (original) optimization problem.

[8] $M^\dagger$ denotes the covariance of a matrix M

III.II.c. Results

The authors indicate that their proposed method outperforms both the use of the Gromov-Wasserstein distance and the Euclidean distance in the task of graph alignement and graph clustering.

In the paper some exemplary applications for graph alignment and graph classification on the MNIST and th Fashion MNIST database are illustrated.
The straight forward implementation results in a run time complexity for one iteration in $\mathcal{O}(n^3)$ where n is. But the authors indicate that the actual implementation is faster by using approximations of the square-roots of matrices using Newton's method and the sparsity of the matrices. On top of that the computation of the pseudo-inverses is claimed to be avoidable by diagonally shifting the Laplacian matrices and directly computing their inverse.

III.II.d. Critic

Since the paper focusses on one problem definition which is refined and transformed throughout the paper, the train of though is quite clear and easy to follow. The presented illustrations do support the explanations.

However some claims are not proved or the argumentation of their correctness seems to be entirely left on other papers. For example the discussion of the runtime complexity (page 7) basically leaves the reader with the claim that the algorithm can be implemented better than in a naive way (which has cubic runtime complexity).

Furthermore the experimental results are compared using different error measurements, suited for different methods which were actually used in these experiments. The comparability of these error measures, in particular without giving their definitions, is implied unfounded.
Similarly the assumption about the distribution of the nodes labels indeed founded on other papers, but the performance of the method on other distributions or examples is not discussed.

At last one can notice that the paper lacks the usual suggestions for future work.

III.III. Paper: *"A Graph Theoretic Additive Approximation of Optimal Transport"*

In this paper the authors present an approximation method to solve find a maximal transportation plan in the optimal transport problem as defined above with no restrictions regarding the cost, demand and supply.

III.III.a. Motivation

The motivation for the research is to find approximate solutions of the optimal transport problem fast and without restricting the cost, demand and supply.

III.III.b. Method

A *δ-close solution* to the given problem is a transportation plan σ which costs at most δ-percent of the total demand more than the optimal transport plan σ^* (i.e. $w(\sigma) \leq w(\sigma^*) + \delta \sum_{b \in B} s_b$).
The proposed method uses and algorithm by Gabow and Tarjan which was defined for integral inputs (demands, supplies and costs). The contribution of the authors is to scale and round arbitrary inputs in order to obtain a similar integer formulation which can be processed similarly like in the algorithm of Gabow and Tarjan, and then using "backwards-rounding" to obtain a transportation plan that is suitable for the initial inputs.
The algorithm used on the scaled and rounded inputs is a primal-dual-algorithm and computes a δ-close solution. Since the computation of only an approximation is targeted, the approach uses a relaxed version of dual feasibility and derivates a primal feasible formulation. The dual variables of the linear program are updated greedily by using Dijkstras algorithm. The primal variables are updated using a partial DFS routine.

III.III.c. Results

The worst-case time complexity of the presented algorithm is bounded by $\mathcal{O}(n^2(C/\delta) + n(C/\delta)^2)$ where n is the sum of nodes in both compared graphs. Note that it is $C = \max_{e \in E} c(e)$.

The researchers conclude their paper with two experiments on data from the dataset MNIST. First they illustrate with an application in mapping pixel colours from one image to another, that the worst case runtime is not heavily reflected in practice.
Second they argue that the proposed method is superior to the former state-of-the art implementations (especially for small values of δ, which results in better approximations)[9].

Future work could consider using a nearest neighbor structure to improve the runtime. Performing the update and query steps in poly-logarithmic time in n would lead to a runtime complexity with linear and logarithmic terms only.
Another approach could be to replace the use of Dikstras algorithm and allow easier parallelization of the procedure.

III.III.d. Critic

The formulations in the paper are rather technical and heavily focusses on statements and proofs. Although proofs are indispensable for the credibility of the research, they often do not help the reader to understand the main ideas and techniques behind the proven statements.
It could be preferable to allow access to the proofs in some other form (e.g. appendix) and shift the focus of the paper towards illustrating and visualizing the ideas, advantages and disadvantages of the method.

[9] By Sinkhorn, Greenkhorn and APDAGD5

IV. Paper comparison and relation

Clearly [1] and [2] are more similar since they both apply a WD to obtain some sort of similarity measure between graphs. On the other hand [3] is focusses one a more general solution of the WD itself. On a methodical level [1] and [2] are also more similar, since both papers focus on explaining the general ideas and concepts behind the methods, rather than immediately proving some theoretical results as it is done in [3].

Nonetheless it is possible to compare the style of the papers and to argue how they can benefit from each other.

As mentioned, [1] uses the 1-WD on real valued vectors representing layers of labels in neighbourhoods in two given graphs. [2] on the other hand uses the 2-WD on assumed distributions on the labels in two given graphs. Both the assumption on present distributions and the compressing of information with respect to neighbourhoods could be tried in the other approach.

Regarding the style of the papers, the illustrations in [2] seem more practical and informative. Especially regarding the applied experimental results. [1] only compares the performance of the presented method in a statistical manner. This in turn could be improved in [2].

Naturally the results in [3] could be used to replace the network simplex method in computing the WD in [1]. Since the main effort of [2] is to suitable use stochastic gradient descent to compute a WD for the presented specialized case, it is evidently how the approximation from [3] could improve the results in [2].
All applications in the other way round seem illogical since this would mean to apply a concrete solution to a more abstract formulation.

To conclude, the three different approaches towards similarity measures for graphs, all using some instance of the WD, indicate, that there are many more possible strategies one could investigate. Relaxing or improving the computation of the WD itself, constructing other graph representations which can be compared using a WD, or combining several of such approaches - this seems to be just a glimpse on how to utilize methods from Optimal Transport in understanding and processing graphs.

References

[1] Matteo Togninalli, Elisabetta Ghisu, Felipe Llinares-López, Bastian Rieck, and Karsten Borgwardt. Wasserstein weisfeiler-lehman graph kernels. In *Advances in Neural Information Processing Systems*, pages 6439–6449, 2019.

[2] Hermina Petric Maretic, Mireille El Gheche, Giovanni Chierchia, and Pascal Frossard. Got: an optimal transport framework for graph comparison. In *Advances in Neural Information Processing Systems*, pages 13876–13887, 2019.

[3] Nathaniel Lahn, Deepika Mulchandani, and Sharath Raghvendra. A graph theoretic additive approximation of optimal transport. In *Advances in Neural Information Processing Systems*, pages 13836–13846, 2019.

[4] Bernhard Korte, Jens Vygen, B Korte, and J Vygen. *Combinatorial optimization*, volume 2. Springer, 2012.