Kernel Methods: Kaggle Challenge (MVA)*

*Code: https://github.com/jpaillard/KernelChallenge.git

Joseph PAILLARD
joseph.paillard@etu.minesparis.psl.eu

Pierre-Adrien PLESSIX
pierre-adrien.plessix@etu.minesparis.psl.eu

I. Context

The challenge at hand involves predicting the function of molecules, specifically through the classification of graphs containing labeled edges and vertices. The dataset used for this challenge comprises a set of 8000 molecules, with a total of 50 different atoms serving as vertex labels and 3 different molecular bindings serving as edge labels. Among the **8000 molecules**, labels are provided for **6000** molecules, while the remaining **2000** molecules are reserved for testing and accessible only to the challenge organizers. The task involves predicting a binary molecular function, where labels correspond to 1 if the molecule has the function and 0 otherwise. It is worth noting that the function is unknown, making it challenging to implement chemical domain knowledge into the method. In this report, we will delve into the details of this challenge and discuss the methodologies employed to tackle it effectively.

II. Kernel methods for graphs

Graph kernel methods have been widely used for graph classification tasks in predicting the function of molecules. These methods involve mapping each molecule to a feature vector in a Reproducing Kernel Hilbert Space (RKHS), which is then used for classification or logistic regression. Many well-known graph kernel methods, such as the graphlet kernel, shortest path kernel, subtree kernel, and random walk kernel, have been extensively studied in the literature and have been improved in terms of computational efficiency [1]. **However, these methods may not inherently capture the labels of edges and vertices in their kernel computation, and may require additional modifications or extensions to do so.** Considering the labels of vertices and edges is one of the most viable ways to incorporate domain knowledge into the graph classification task. As such, our approach explores the use of the **Weisfeiler-Lehman (WL) kernel** [2], known for its ability to capture the labels of atoms and provide a more expressive representation of molecular structures.

III. The Weisfeiler-Lehman kernel [2]

A. Original Concept

The algorithm starts by initializing each vertex with a unique label, typically based on its original label (e.g., atom type in the case of molecules). Then, in each iteration, the labels of vertices are updated by considering the labels of their neighbors, and a new label is assigned to each vertex based on a combination of its own label and the labels of its neighbors. This process is repeated for a fixed number of iterations, and the **final vertex labels are used to compute a feature vector for the graph.**

The iterative updating of vertex labels based on their neighbors enables the algorithm to identify common **functional groups** or motifs that contribute to the same molecular function. This makes the WL kernel algorithm well-suited for capturing local neighborhood information and identifying relevant features in graph classification tasks for predicting molecular function.

B. Enhancement proposition

Unlike what was presented previously in the litterature, the dataset of this challenge contains graphs with **annotated edges**. We therefore propose a modification of the WL algorithm to take into account this specific feature.

This particularity is essential given the fundamental role that bond types play in the **functionality of the chemical groups.** To do so, at each step of the algorithm, for a given node, the label update concatenates the information of the neighboring atoms' type and the bond that connects them (instead of only considering the atom type). This method will be refered to as *edge embedding.*

IV. Classifier implementation

To solve the classification task in the RKHS, two methods have been implemented: kernel Support Vector Classification (SVC) and Kernel Logistic Regression (KLR). The code used for this project and briefly described in this section is available on `github` [1].

a) KLR: We implemented the solving of the KLR problem using Iteratively Reweighted Least-Square (IRLS), algorithm that is described in the lectures slides.

b) SVC: After initially using SciPy to solve the optimization problem for support vector classification (SVC) we turned to the quadratic programming solver from cvxopt[2], which significantly improved computation time. This improvement can be attributed to cvxopt's specialized solvers for convex optimization problems like SVC, while SciPy is a more general-purpose library.

V. Methodology

Given the data set structure we propose the following training method :

[1]https://github.com/jpaillard/KernelChallenge.git
[2]cvxopt.org

TABLE I
RESULTS OF TRAINING AND SUBMISSION ON KAGGLE

		Edge embedding				None			Edge embedding				None
		$h=1$	$h=2$	$h=3$	$h=4$	$h=2$			$h=1$	$h=2$	$h=3$	$h=4$	$h=2$
KLR							**SVC**						
$c=0.0001$	train		0.99			0.98	$c=0.001$	train	**0.9**				0.93
	valid		0.8452			0.76		valid	**0.901**				0.83
	test							test	**0.87**				
$c=0.001$	train	0.87	0.95	0.99	1	0.93	$c=0.01$	train	**0.89**		0.99		
	valid	0.86	0.84	0.83	0.82	0.78		valid	**0.9**		0.879		
	test	0.86	0.85					test	**0.868**		0.811		
$c=0.01$	train		0.87			0.86	$c=0.1$	train	0.9				0.95
	valid		0.81			0.76		valid	0.8983	0.88	0.89		0.795
	test							test	0.864	0.824	0.819		
							$c=1$	train	0.91				
								valid	0.89				
								test	0.836				

A. Training Phase

The purpose of the training phase is to train the model and determine the optimal hyper parameters. The following hyper-parameters have been tuned:

- *Model* (SVC or KLR)
- *Regularization parameter* C
- *Depth of WL Kernel embedding*
- *Edge embedding* (Yes or No)

To carry out the training of our methods with kernels and the selection of hyperparameters, we use the *train-valid-test method* using 80 % of the shuffled dataset for training.
To evaluate our models we use the ROC AUC metric since it is the one used in the evaluation of our model on kaggle.

B. Testing phase

Before testing, the model was re-trained on all available labeled data (train + valid). The output of the model are float numbers that were then submitted on the kaggle platform.
The resulting public score is only partial and cannot be used to make decisions about the training method
One of the major drawback remains the training/test time which is considerably high.

VI. RESULTS

Results are presented in the table Table I. It can be seen that as the depth of the WL algorithm increases, the performance of the algorithm becomes worse, mainly in terms of overlearning. This does not seem to us to be an aberration given the characteristic distance of the interactions at the molecular scale which is not more than one or two atoms, except in the case of the mesomeric effects which appear at great distance.
The results obtained show unequivocally the importance of the embedding of the edges in the encoding of the neighborhood of the atoms. Indeed, we obtain much better results with this method than with the original method.
We observe that KLR seems to yield better results than SVC for classification.
We observe that our training and test score values on the public game consider the same set of hyperparameters as more promising. This reassures us on the basis of our choices.

VII. DISCUSSION

To continue the improvement of the implementation, we think that it is necessary to continue to refine the structure of the WL kernel to allow it to better represent the structures of the molecules. Indeed, results presented in table I show that representation limits of kernel have been reached.
The knowledge of the mechanisms at work in the construction of the **chemical reactivity of the functional groups** indicates us several tracks to follow to improve a priori **the quality of the representativeness** of the spaces in which we project our graphs.
1. To take into account the ***chirality***[3], i.e. the intrinsic difference of two groups which are image of each other in a mirror. Thus the spatial distribution of the atoms is fundamental and can lead to very different reactivities (cf. thalidomide[4]). However the data set here does not allow access to this geometrical information.
2. To take into account potential long distance electron interaction via ***mesomery***[5]. This would correspond to a the use of a selective depth-first algorithm instead of a Breadth-first search.

REFERENCES

[1] SVN Vishwanathan, Karsten M Borgwardt, and Nicol N Schraudolph. "Fast computation of graph kernels". In: *NIPS*. 2006.

[2] Nino Shervashidze et al. "Weisfeiler-lehman graph kernels." In: *Journal of Machine Learning Research* (2011).

[3] https://en.wikipedia.org/wiki/Chirality(chemistry)
[4] https://en.wikipedia.org/wiki/Thalidomide_scandal
[5] https://en.wikipedia.org/wiki/Resonance(chemistry)