

Graph Kernels and Support Vector Machines for Pattern Recognition

Léo Andéol*

Master DAC, Sorbonne Université
Paris, France

May 2019

Abstract

Nowadays the world is turning to a new era where data is the most valuable resource, and we are working on algorithms to exploit those the best way possible. Most research is currently focused on the so-called "big data", huge quantities of data which are very sparse and of poor quality. However, there also exist structured databases of good quality, in the form of graphs, which is the one we will be studying. Graphs can be very useful in various fields, but is mostly known for its use in biology, as it can be used to represent proteins or other types of molecules.

I chose to work on Support Vector Machines(cite qui/quand), a well known and powerful algorithm of machine learning. It is a perfect fit for studying graphs and learning to compare them because the algorithm allows through an elegant trick the use of several types of structured data. The goal of this project is to implement a working algorithm and its improvements, test it on different types of data and try to improve it.

reecrire une
fois rapport
fini : Ra-
jouter ap-
port projet
et facteur
acceleration

*leo.andéol@gmail.com

Contents

1	Introduction	3
2	Methodology	4
2.1	Background	4
2.1.1	Graphs	4
2.1.2	Support Vector Machines	5
2.1.3	Kernels	6
2.2	State of the Art	7
2.2.1	Introduction	7
2.2.2	Graphlets	7
2.2.3	Random walks	7
2.2.4	Kernel definition	8
2.2.5	Sylvester Equation	8
2.2.6	Conjugate Gradient Method	8
2.2.7	Fixed-Point Iterations	8
2.3	Contributions	9
3	Experiments	9
3.0.1	Spectral Decomposition	9
3.0.2	Nearest Kronecker Product	9
3.1	Implementation	10
3.1.1	Graph Database	10
3.1.2	Kernels	11
3.2	Test of the random walk kernel	11
3.2.1	Accuracy comparison between labelled and unlabelled graphs	11
3.2.2	Efficiency of alternate methods	11
3.2.3	Experiments on a biology dataset	11
3.3	Improvements	12
4	Conclusion and Future Work	12
A	Appendix	12
B	Annex 1	12
C	Acknowledgements	12

1 Introduction

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque

introduction
générale plus
étendue : 1
et 1/2 page
:donner un
aperçu/résumé
du rapport
: un para-
graphe par
élément -
ECRIRE A
LA FIN

egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetur at, consectetur sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

2 Methodology

2.1 Background

sous section 1 : background : graphe, svm, noyaux, donner toutes les définitions.

2.1.1 Graphs

A graph(1) is a type of mathematical structure that represents the connection between object. It is thus composed of vertices (or nodes) and edges. Vertices represent objects and are usually depicted as circles or spheres whereas edges link pairs or vertices.

Definition 2.1. A graph is an ordered pair $G = (V, E)$ where :

- V is a set of nodes
- $E \subseteq (u, v) | (u, v) \in V^2$

Est-ce que c'est la bonne expression en anglais ?

Reference pour graphes

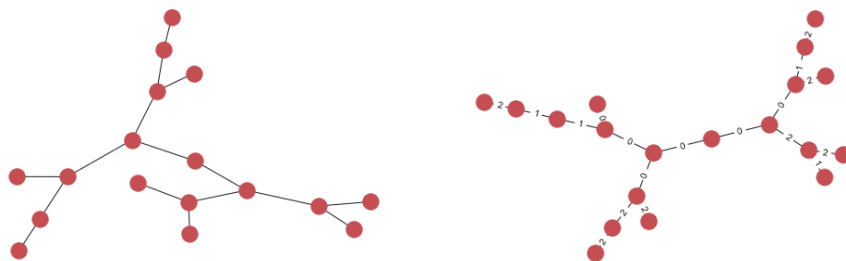


Figure 1: Two "tree" graphs, resp. unlabeled and labeled

The edges of the graph can be weighted and can be either oriented or not. We will be focusing on undirected unweighted graphs, thus we have :

Definition 2.2. $\forall (u, v) \in E \implies (v, u) \in E$

Both vertices and edges can have labels, which can take various forms : they can be elements of a finite or infinite set (such as integers but also strings), or even continuous (such as $\in \mathbb{R}$). Graphs were first used in their modern form to represent the problem of Seven Bridges of Königsberg (cite), and have been ever since used to represent maps, and thus path-finding algorithms were developed. Graphs can also be used, with labels, to represent different type of molecules and interactions between them.(parler aussi des réseaux ? des problèmes de flots etc ?)

During this project, we will focus on undirected graphs which can have any type of labels on their edges only (if it is only on their nodes, the complementary graph can be studied).

We will be comparing graphs in order to find similarities between them, and classify them in different groups. It has been shown(cite) that it can be used effectively on proteins and (?).

2.1.2 Support Vector Machines

Support Vector Machines (SVM) are a type of machine learning algorithms discovered in the early 90s(cite). It was originally a classification algorithm however it has been expanded since to regression and clustering too. It is especially powerful and widely used for two main reasons :

Margin and support vectors The Support Vector Machines were based on the model of the Perceptron (cite), another classification algorithm that tried to find an hyperplane that discriminates the two sets.

The main issue with the Perceptron is that it had no formal guarantee to find an optimal hyperplane, and more often than not would not find it. Moreover, if the data weren't linearly separable, the algorithm would not converge.

In order to tackle these issues, the SVM offered three fixes.

First, they added a margin in the loss function in order to not only classify well the data samples, but also with the maximum certainty, i.e. as far from the decision boundary as possible.

$$L(y, x, w) = \max(0, -y * x \cdot w) \implies L(y, x, w) = \max(0, 1 - y * x \cdot w) \quad (1)$$

Then, seeing this solution wasn't enough, as there was still an infinity of possible solutions, it was proven that the margin between points and the decision boundary was inversely proportional to the norm of the weights.

$$\gamma = \frac{y_i(x_i \cdot w + w_0)}{\|w\|} \quad (2)$$

These two fixes gave us the first version the SVM, the hard-margin SVM :

$$\min \frac{1}{2} \|w\|^2 \quad \text{s.t.} \quad y_i(x_i \cdot w + w_0) \geq 1 \quad \forall i \in \{1..n\} \quad (3)$$

However, if the data weren't linearly separable the algorithm wouldn't be optimizable since the quadratic programming problem requires all points to be correctly classified. Then, a new term ξ was introduced as an error tolerance, as well as a factor C that would determine the balance between error tolerance and weights minimization.

$$\begin{aligned} \min \quad & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i \\ \text{s.t.} \quad & \forall i \in \{1..n\} \quad y_i(x_i \cdot w + w_0) \geq 1 - \xi_i \end{aligned} \quad (4)$$

2.1.3 Kernels

In its dual form, the SVM problem only requires a dot product between the sample vectors. Then, we can use a nonlinear transformation $\phi(x)$ to replace the dot-product $\mathbf{x}_i \cdot \mathbf{x}_j$ by $\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$, effectively augmenting the dimensions of our vectors and thus the expressiveness of our model by enhancing the separability of the data. However, for complex transformations, such as in infinite dimensions, the function ϕ isn't defined. But it was found(cite) that the standard dot product can be replaced by another function as long as it is a bilinear positive semi definite form.

$$\sum_{i=1}^n \sum_{j=1}^n K(\mathbf{x}_i, \mathbf{x}_j) c_i c_j \geq 0 \quad \forall i \in \{1..n\} \quad c_i \in \mathbb{R} \quad (5)$$

Thus, the dot product can be replaced by a function $K(x_1, x_2)$ which will indirectly (?) transform the data in larger dimensions, even infinite with the Gaussian kernel(cite) and return the dot product of those transformations, while remaining computable in all cases. This replacement is usually called the "Kernel Trick".

Ever since the foundation of SVMs, the kernel trick became a big focus of the

machine learning community as it naturally fits in the algorithm and allows supervised learning on very complex data, and enjoying greater accuracy than most algorithms.

2.2 State of the Art

sous section 2 : state of the art : graph kernels et acceleration des random walk, donner les complexité, donner des exemples, discuter avantages et inconvenients chacun

2.2.1 Introduction

In this chapter we will discuss the advance of research on graph kernels. Graph kernels have been studied since Support Vector Machines started getting popular(cite). Since then, a lot of progress has been made, and several types of kernels have been discovered, such as random walk and graphlet kernels, the two we will be stuyding, but not the only ones. There are also graph kernels that use spanning trees, and several others that are unfortunately usually not semi definite(cite nino). **R-convolutionkernels,rationalkernels**
source:https://publikationen.uni-tuebingen.de/xmlui/bitstream/handle/10900/49731/pdf/thesis_all.pdf?sequence=1&isAllowed=y

2.2.2 Graphlets

A graphlet(cite) is a subgraph of 3,4 or 5 nodes taken from a much larger graph. The idea of a graphlet kernel is to count all types of graphlets that appear in a graph, and use that count to compare to another graph.
insert list of graphlets

2.2.3 Random walks

The state of the art of this type of kernel is mainly defined in Graph Kernels(cite) although there have been several other developments recently (cite). Random walks are a type of rather intuitive algorithms : as its name says, it randomly walks through a graph for a certain time, and we can use these walks to compare graphs as it has been shown(cite). Actually, we will walk through all possible paths in the graph we will compare using a simple trick : by making the adjacency matrix stochastic, we can use it as a markov chain. Moreover, it has also been shown(cite) that performing a walk on two separate graphs at the same time is the same as performing a walk on the product graph. The product graph is in computer science terms the Cartesian product of the two graphs, taking all possible combinations of edges and nodes, but mathematically, the adjacency matrix of a such graph is the result of the Kronecker product of the two graphs(cite) (expliquer?) :

$$W_{\times} = A \otimes A_2 \quad (6)$$

2.2.4 Kernel definition

2.2.5 Sylvester Equation

A Sylvester equation, also sometimes called Lyapunov equation (a generalized version of it) is a matrix equation of the following shape :

$$AX + XB = C \quad (7)$$

However, there also exist a discrete-time version of it :

$$AXB + C = X \quad (8)$$

TODO solvable how, what efficiency

2.2.6 Conjugate Gradient Method

Note : Preconditioner random vector : $x \cdot x^T$ fonctionne bien

The Conjugate Gradient Method is an optimization algorithm used to find approximate solutions of linear systems that have a positive semi definite matrix. As its name suggests, it is a gradient based (usually) iterative algorithm. The Main idea is

2.2.7 Fixed-Point Iterations

si lambda trop proche de l'inverse de la valeur propre peut etre tres lent à converger quand la matrice d'adjacence est très dense

Fixed point consists in ...

proof that $\lambda < |\xi|^{-1}$

Proof. Let $x_0 = p_\times$ and $t \gg 0$

$$\begin{aligned}
x_{t+1} - x_t &= p_\times + \lambda W_\times x_t - x_t \\
&= p_\times + (\lambda W_\times - 1)(p_\times + \lambda W_\times x_{t-1}) \\
&= p_\times - p_\times + \lambda W_\times p_\times + (\lambda W_\times)^2 x_{t-1} - \lambda W_\times x_{t-1} \\
&= \lambda W_\times (p_\times - x_{t-1} + \lambda W_\times x_{t-1}) \\
&= \lambda W_\times (p_\times - \lambda W_\times x_{t-2} - p_\times + \lambda W_\times (\lambda W_\times x_{t-2} + p_\times)) \\
&= \lambda W_\times (\lambda W_\times (\lambda W_\times x_{t-2} + p_\times - x_{t-2})) \\
&= (\lambda W_\times)^2 (\lambda W_\times x_{t-2} + p_\times - x_{t-2})
\end{aligned}$$

We find a pattern

$$\implies x_{t+1} - x_t = (\lambda W_\times)^t (\lambda W_\times x_0 + p_\times - x_0)$$

And because $x_0 = p_\times$

$$= (\lambda W_\times)^{t+1} p_\times$$

Which decreases only when

$$\xi_1 < 1$$

The largest magnitude eigenvalue of λW_\times

$$\implies \lambda < |\xi_1|^{-1}$$

□

Rapidité de la convergence : géométrique avec un ratio λ_1/λ_2 ???

2.3 Contributions

en anglais ça donne ?

3 Experiments

experiences : détailler db, tests, méthodes, les parametres, construction label, donner tableaux, resultats, teps de calculs, précision, discuter tout cela

3.0.1 Spectral Decomposition

can't inverse Left eigenvectors : <https://arxiv.org/pdf/1708.00064.pdf>

3.0.2 Nearest Kronecker Product

[https://www.sciencedirect.com/science/article/pii/S0377042700003939?](https://www.sciencedirect.com/science/article/pii/S0377042700003939?via%3Dihub)
[via%3Dihub https://www.imageclef.org/system/files/CLEF2016_Kronecker_Decomposition.pdf](https://www.imageclef.org/system/files/CLEF2016_Kronecker_Decomposition.pdf) http://dx.doi.org/10.1007/978-94-015-8196-7_17 <https://mathematica.stackexchange.com/questions/91651/nearest-kronecker-product>

3.1 Implementation

3.1.1 Graph Database

The first step was to create a graph database generator that we would use to test our algorithms on simple cases with expectable behaviour. I have chosen to focus on 3 types of graphs (todo expand) : ring, star and tree graphs.

The graph generator can create graphs that are either labelled or unlabelled. As for labelled ones, it will simply assign to all edges a random integer between 1 and the given limit during the creation of the database. Moreover, the database

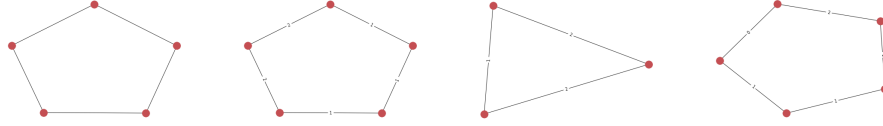


Figure 2: Ring graphs, resp. unlabelled, labelled, altered structured, altered labels

generator will also alter all generated graphs both structurally and on labels. Concerning labels, some edges are picked in the graph following a limit given by the user and the new label values of those edges are randomly generated in the same set as the original ones. Finally, the structure of generated graphs will

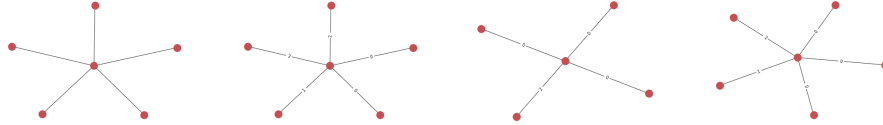


Figure 3: Star graphs, resp. unlabelled, labelled, altered structured, altered labels

be altered depending on their type, but will always keep the same set of labels (except for new edges).

1. Ring graphs are simply regenerated slightly longer or shorter randomly.
2. Star graphs are also regenerated either with more or less nodes, randomly.
3. Tree graphs are either expanded by adding randomly leaves anywhere in the graph, or reduced by randomly removing leaves.

After having generated the whole database, graphs will be transformed to adjacency matrices since we will only be handling graphs under this form. For unlabelled graphs, it simply returns the adjacency matrix (expliquer ce que c'est ?) and for labelled graphs it returns an array of adjacency graphs, only considering the partial graph of the edges of a particular label at a time.

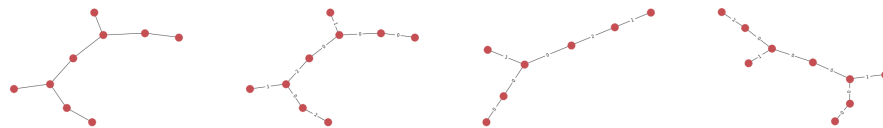


Figure 4: Tree graphs, resp. unlabelled, labelled, altered structured, altered labels

3.1.2 Kernels

3.2 Test of the random walk kernel

3.2.1 Accuracy comparison between labelled and unlabelled graphs

Même score, car donne même matrice ???

	Unlabelled	Labelled
Accuracy	?	?
Time	?	?

Table 1: Time and Accuracy of learning resp. for unlabelled and labelled graphs

3.2.2 Efficiency of alternate methods

	Raw kernel	Inverse Kernel	Sylvester Equation	Conjugate Gradients	Fixed points	Spectral Decomp.	Nearest Kronecker Product
Accuracy	67%	67%	?	67%	67%	?	?
Time	18.40s	5.72	?	7.62s	5.94s	?	?

Table 2: Time and Accuracy of learning for the raw kernel and other methods

3.2.3 Experiments on a biology dataset

<https://ls11-www.cs.tu-dortmund.de/staff/morris/graphkerneldatasets>
<http://members.cbio.mines-paristech.fr/~nshervashidze/code/enzymes>

3.3 Improvements

4 Conclusion and Future Work

experiences : détailler db, tests, méthodes, les parametres, construction label, donner tableaux, resultats, teps de calculs, précision, discuter tout cela section 4 : 1 page ou page et demi : conclusion et discussion

A Appendix

B Annex 1

C Acknowledgements

This work was done during the first year of my master.

List of Figures

1	Two "tree" graphs, resp. unlabeled and labeled	5
2	Ring graphs, resp. unlabelled, labelled, altered structured, altered labels	10
3	Star graphs, resp. unlabelled, labelled, altered structured, altered labels	10
4	Tree graphs, resp. unlabelled, labelled, altered structured, altered labels	11

List of Tables

1	Time and Accuracy of learning resp. for unlabelled and labelled graphs	11
2	Time and Accuracy of learning for the raw kernel and other methods bibliographie et index	11

References

- [1] Wikipedia, "Graph theory - Wikipedia," 2019.
- [2] Y. Nesterov, *Lectures on Convex Optimization*. Springer Optimization and Its Applications, Springer International Publishing, 2 ed., 2018.