

Diffusion Fingerprints

Jimmy Dubuisson
Jean-Pierre Eckmann
Andrea Agazzi

*Département de Physique Théorique and Section de Mathématiques
Université de Genève*

JIMMY.DUBUISSON@UNIGE.CH
JEAN-PIERRE.ECKMANN@UNIGE.CH
ANDREA.AGAZZI@UNIGE.CH

Abstract

We introduce, test and discuss a method for classifying and clustering data modeled as directed graphs. The idea is to start diffusion processes from any subset of a data collection, generating corresponding distributions for reaching points in the network. These distributions take the form of high-dimensional numerical vectors and capture essential topological properties of the original dataset. We show how these diffusion vectors can be successfully applied for getting state-of-the-art accuracies in the problem of extracting pathways from metabolic networks. We also provide a guideline to illustrate how to use our method for classification problems, and discuss important details of its implementation. In particular, we present a simple dimensionality reduction technique that lowers the computational cost of classifying diffusion vectors, while leaving the predictive power of the classification process substantially unaltered. Although the method has very few parameters, the results we obtain show its flexibility and power. This should make it helpful in many other contexts.

Keywords: graph theory, complex network, machine learning, natural language processing, classification, authorship detection, random walk, pagerank, dimensionality reduction, feature extraction, metabolic pathway, subgraph extraction, graph mining

1. Introduction

Our method combines several ideas that can be applied in a flexible way, either for classification or graph mining. By using a graph modeling of associations within a data collection, we generate “fingerprints” for any subset of data items in the form of high-dimensional distribution vectors of a random walk diffusion process. When such a graph is not provided explicitly, we first compute association matrices for each subset of the data collection, and then merge them to form a directed graph, after a threshold is applied. Since the diffusion vectors are in a high-dimensional space (the dimension being, for example, the number of different tokens in a corpus of texts), we propose a simple dimensionality reduction that allows for reasonable computational cost associated with the classification of these vectors.

Various related methods start to build a weighted undirected graph in the form of a similarity matrix, and make use of the spectrum of a derived Laplacian matrix for dimensionality reduction or clustering [Shi and Malik \(2000\)](#); [Belkin and Niyogi \(2001\)](#); [Coifman et al. \(2005\)](#).

Diffusion kernels [Kondor and Lafferty \(2002\)](#) propose a general method for constructing kernels on undirected graphs, so that such discrete structures can be used with classical kernel-based learning algorithms [Schölkopf and Smola \(2002\)](#). Unfortunately, this approach remains in general computationally expensive for large graphs.

Random walks over undirected graphs have also been used as a similarity measure for collaborative recommendation [Fouss et al. \(2007\)](#), community detection [Andersen et al. \(2006b\)](#); [Pons and Latapy](#)

(2006), or relational classification Perozzi et al. (2014). The *ItemRank* algorithm Gori et al. (2007), on the other hand, uses biased random walks over a directed graph to compute a score vector for each user, within the framework of a recommender system.

Although related to these previous works, our method combines a set of specific features. First, by operating preferably on directed graphs, it enables to get deeper insights into the topology of the dataset under consideration. Moreover, it can potentially use many types of biased random walks, and does not need the diffusion processes to reach full convergence (this may help to quickly compute estimates of the fingerprints, or to get “snapshots” of the dataset at different times). Finally, it can be used to compute numerical vectors (one may think of “feature vectors”) for any subset of data items, including overlapping ones for example.

In the present paper, we start by presenting the basic formalism of our method in section 2, then show how it can be successfully applied to the problem of extracting relevant pathways from metabolic networks in section 3. In section 4, we provide a guideline to illustrate how to use our method for classification problems. Finally, we discuss a few important details more thoroughly in section 5.

2. Formalism

2.1. The association matrices

We consider a *data collection* $\Sigma = \{\sigma(1), \sigma(2), \dots\}$ of “documents”, where each *document* is viewed as a set of data items whose pairs get assigned a value of association. Some data collections such as the USF Free Associations dataset Nelson et al. (2004); Dubuisson et al. (2013) already provide explicit association weights while for others, one needs to define a way to compute these association values. The set of different items appearing in Σ is called \mathcal{T} . We denote $|\mathcal{T}|$ the cardinality of \mathcal{T} .

2.1.1. AN EXAMPLE: COMPUTING WORDS COLLOCATION

To illustrate our terminology and give an example of how to compute such association values when necessary, we consider a corpus Σ of text documents. A document $\sigma(k)$ will consist here of $N(k)$ tokens, which are typically stemmed words, with some stop words omitted. We define $I(k) = \{I_k(1), \dots, I_k(N(k))\}$ as the list of tokens of $\sigma(k)$ in the order in which they appear. Each $I(k)$ is thus a map from positions in $\sigma(k)$ to tokens in \mathcal{T} .

We next define, for each k , the association matrix $K(k)$, which is a $|\mathcal{T}| \times |\mathcal{T}|$ matrix. We fix k and omit the index k for the moment. The matrix K measures the association of pairs of tokens $u, v \in \mathcal{T}$. For every $u \in \mathcal{T}$, we let $p_u(i)$ be the position of the i^{th} occurrence of token u . For every ordered pair (u, v) with $u \neq v$ of tokens we look for occurrences of the form $p_u(i) < p_v(j) < p_u(i + 1)$, that is, occurrences of token v between two successive occurrences of token u (or after the last occurrence of u). We let s_{uv} be the set of all such pairs (i, j) . Still omitting the index k , the matrix K is defined by

$$K_{uv} = g(h(u, v)) \sum_{(i,j) \in s_{uv}} f(p_u(i), p_v(j)),$$

where

$$h(u, v) = \frac{|s_{uv}|}{\sum_{\substack{u', v' \in \mathcal{T}(k) \\ u' \neq v'}} |s_{u'v'}|},$$

$\mathcal{T}(k)$ are the tokens appearing in $I(k)$ and the functions $g(\cdot)$ and $f(\cdot, \cdot)$ are defined below. Note that $K_{uv} = 0$ if s_{uv} is the empty set.

We conduct our experiments with the following families of functions:

$$f(i, j) = \exp\left(-\frac{(j - i - 1)^\beta}{\sigma}\right) \text{ and } g(x) = -\log(x) .$$

The rationale behind the use of the function f is that the collocation measure should decrease exponentially with the distance between any two tokens. $g(h(\cdot))$ is a function of the relative frequency of each pair s_{uv} and serves as a normalizing function whose goal is to correct the influence of very frequent pairs. (These functions might be changed somewhat depending on the study one wants to perform.)

2.2. The domain graph

Having generated, for each document k , the association matrix $K(k)$ as described above, we next define a matrix $K(\Sigma)$ for the whole data collection Σ by

$$K(\Sigma)_{uv} = \sum_k K(k)_{uv} .$$

We now introduce a density parameter γ and define with it an *adjacency matrix* $A(\gamma)$ as follows: we replace the $N \equiv \gamma|\mathcal{T}| \cdot (|\mathcal{T}| - 1)$ largest elements of $K(\Sigma)$ by 1, and the others by 0. This means that the matrix elements of $K(\Sigma)$ above a certain threshold are replaced by 1 and the others by 0.¹ Note that our method does not specifically require the use of a binary adjacency matrix, and can easily be adapted to work with a weighted one, if it brings a clear benefit to do so.

The *domain graph* $G(\gamma)$ is the directed graph whose nodes are the elements of \mathcal{T} and whose adjacency matrix is $A(\gamma)$. The topology of $G(\gamma)$ reflects the N strongest associations in Σ for a given density γ .

2.3. The diffusion fingerprints

Having determined the directed graph $G(\gamma)$, we now consider a diffusion process on it. In particular, we are interested in how a given document $\sigma(k)$ fits into this graph.

For a fixed k , there is a set $\mathcal{T}'(k) \subset \mathcal{T}$ of data items which are nodes of the domain graph and which appear in $\sigma(k)$. We want to know how the set $\mathcal{T}'(k)$ diffuses into the domain graph.²

We call *diffusion fingerprint* (DF) of document $\sigma(k)$ the distribution vector of the diffusion process started from the subset $\mathcal{T}'(k)$ of nodes in $G(\gamma)$. Note that the smaller the set $\mathcal{T}(k) \setminus \mathcal{T}'(k)$, the better the generated fingerprint represents $\sigma(k)$ within the context of the domain graph.

Let P be the *probability matrix* defined by

$$P(\gamma) = D^{-1}(\gamma)A(\gamma) ,$$

where D is the diagonal matrix of the degrees of G . We compute the DF of document $\sigma(k)$ as the *personalized Pagerank* Page et al. (1999); Andersen et al. (2006a) ppr_k defined recursively by

$$\text{ppr}_k(t + 1) = \alpha v_k + (1 - \alpha) \text{ppr}_k(t)P , \tag{1}$$

where $\alpha \in (0, 1]$ is called the *jumping constant* and $\text{ppr}_k(0) = v_k$. The vector v_k is the *personalized vector* given by

$$v_k(u) = \begin{cases} f_k(u) & \text{if } u \in \mathcal{T}(k) \\ 0 & \text{otherwise} \end{cases} ,$$

1. In case of multiplicities (for example if all matrix elements of $K(\Sigma)$ are equal), we perform a random choice of the required number of elements.

2. $\mathcal{T}'(k)$ might be smaller than the set $\mathcal{T}(k)$ of tokens in $\sigma(k)$.

with $f_k(u)$ the frequency of data item u in document $\sigma(k)$. Note that the parameter $(1 - \alpha)$ is the inverse of the expected path length of a random walker before being projected back to $\mathcal{T}'(k)$. In principle, we define

$$\pi(k) = \lim_{t \rightarrow \infty} \text{ppr}_k(t),$$

and call $\pi_t(k)$ the *DF at time t* of document $\sigma(k)$, and $\pi(k)$ its *stationary DF*.³ Note that by default, we use the personalized Pagerank algorithm for computing the diffusion vectors as its properties are well-studied and understood, but nothing prevents our method to be used with other types of biased random walks. Considering the DFs at different times may also permit to construct derived feature vectors taking into account the dynamics of the diffusion processes.

3. Application to metabolic pathway inference

We will now apply the general method sketched out in section 2 to the problem of extracting metabolic pathways from metabolic networks. After describing what the question is, we proceed with the application of the general ideas, adapting them, where needed, to the specificities of the task.

3.1. Description of the problem

The understanding of the dynamics underlying the set of metabolic reactions in a living cell has been pushed a step further by the extensive application of high-throughput analysis techniques in cell biology. In particular, the quantity and accuracy of data produced with these methods has dramatically increased in the last decade, and new computational techniques are required in order to interpret the set of these experimental results as a whole.

The metabolomic datasets can be represented by simple or bipartite graphs. In the first case, the nodes of the graphs are the metabolites, and a (directed or undirected) link is drawn between two molecules if there exists a chemical reaction having one of them as substrate and the other as product. In this case, we refer to the graph as the species-species graph (SSG). In the second case, the nodes of the graphs representing chemical interactions are both chemical molecules and reactions, interlinked by in- and out-flow relations. This kind of representation is referred to as species-reactions graph (SRG). An interesting problem in this context is the one of predicting metabolic pathways (chains of interlinked reactions, transforming a set S of source metabolites into a set T of targets) in metabolic databases.

Different approaches have been proposed for solving this problem: Zien et al. (2000) have developed a method that ranks paths connecting a source and a target node in a SRG, according to gene expression levels corresponding to enzymes catalyzing reactions on the possible paths. Approaches based on the same intuition have been developed for pathway discovery in protein-protein interaction networks, by applying breadth-first search-based and Steiner tree problem solving algorithms Scott et al. (2005); Rajagopalan and Agarwal (2005).

Other algorithms for metabolic pathway extraction based on shortest-path finding have been developed in Antonov et al. (2009); Faust et al. (2010). In particular, a method based on random walks on weighted metabolic graph is used in Faust et al. (2010) to extract metabolic pathways connecting a given set of nodes in a SRG extracted from the *MetaCyc v11.0* database. This is done by searching for the shortest paths interlinking the set of terminal nodes of a given pathway, where by shortest path one means the path with minimal summed weights.

The weighting algorithm used in this case is based on the expected number of times a random walker transits through a given edge before reaching a terminal node of the pathway. The computational cost of this algorithm is $O(sm^3)$ Dupont et al. (2006), whereby m is the total number of

3. Of course, we just compute $\text{ppr}_k(t)$ for some sufficiently large t .

edges of the domain graph and s the cardinality of $S \cup T$). An arbitrary limit on the path length must be given in order to reduce the complexity of the algorithm. In the following we propose an alternative to this metabolic pathway-finding algorithm, based on the DF method explained above.

3.2. Description of the algorithm

The graph G here is the directed domain graph modeling the metabolic network we want to analyze and P is the corresponding transition probability matrix. No filtering procedure has been applied on the domain graph in this case. We call *reverse graph* of G and denote G^* the graph having the same set of vertices as G , with all its edges reversed. We thus have $P^* = P^t$.

The domain graph G can be either simple or bipartite according to the representation of the metabolic network we choose to apply (SSG or SRG respectively).

Let $R \subset V$ be the set of species that participate in a given annotated metabolic pathway. The nodes in R are weakly connected in the domain graph G . We also denote $S \subset R$ the set of sources (*i.e.*, nodes with null in-degree), and $T \subset R$ the set of sinks (*i.e.*, nodes with null out-degree) of the pathway.

For a general pathway, we want to reconstruct the set R of nodes participating in it starting from the sets S of sources and T of sinks.

We start by computing $\pi(S)$, the stationary DF of S in G , and $\pi^*(T)$, the stationary DF of T in G^* . In order to exhibit the set of nodes that are highlighted by both of the fingerprints, we consider the *combined diffusion fingerprint*

$$\pi_{\triangleright}(S, T) = \pi(S) \times \pi^*(T) ,$$

where \times represents the Hadamard Product (component-wise multiplication).

3.3. Pagerank boosting

Because of the presence of hub nodes (such as H_2O , ADP , $NADH$) in the corresponding graphs, a direct application of the algorithm described above would result in pathways connecting source and target nodes only through such highly connected compounds, since they effectively represent the shortest path to connect any given set of two nodes in the graph. This problem is well known and different methods have been developed to overcome it [Faust and van Helden \(2012\)](#). We propose here a method which is softer than the direct elimination of hub metabolites having a total degree above an arbitrarily fixed threshold.

In order for the algorithm to find relevant metabolites, *i.e.*, those that characterize a particular pathway and belong to as few other paths as possible, we renormalize our results keeping the centrality of the different nodes into account. In particular, we rescale the values of the combined diffusion fingerprint vector $\pi_{\triangleright}(S, T)$ resulting from the algorithm described above using the Pagerank vector $\pi(G)$ of the full graph G , and the Pagerank vector $\pi(G^*)$ of the full reverse graph G^* in the following way:

$$\pi_{\triangleright}(S, T)^{boosted} = \frac{\pi_{\triangleright}(S, T)}{\pi(G) \times \pi(G^*)} ,$$

where both multiplication and division are intended in the component-wise sense.

This boosting procedure effectively penalizes metabolic "hubs" such as H_2O and ATP which have high in- and out-degrees.

3.4. Pathway selection

In order for the algorithm to detect a pathway connecting a set S of source nodes to a set T of sink nodes, we consider the n largest entries of the vector $\pi_{\triangleright}(S, T)^{boosted}$, and increase n until the subgraph resulting from the first n^w compounds connects all the elements of S to all the elements

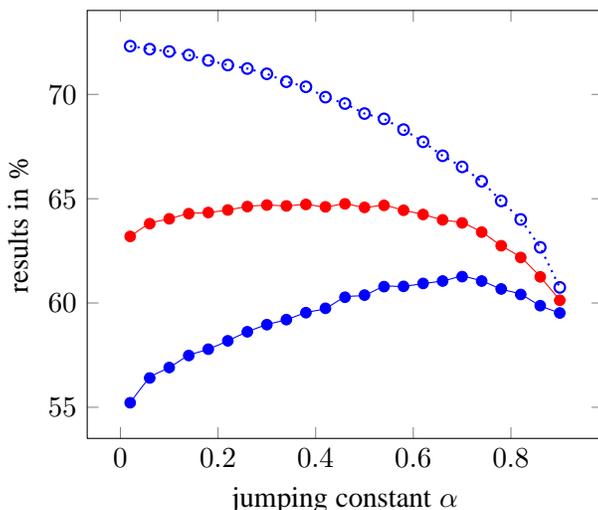


Figure 1: Precision (filled blue circles) and recall (empty blue circles) of pathway inference averaged over 1981 experiments as a function of the jumping constant α . The geometric mean (red filled circles) of these two quantities, later also referred to as the *base score geometric accuracy*, is very weakly affected by the choice of α in the interval $\alpha \in (0.1, 0.6)$.

of T in the weak sense. The extracted subgraph (the inferred pathway) is then given by the subset of the n^w largest entries of $\pi_{\triangleright}(S, T)^{boosted}$ that belong to the weakly connected component connecting S to T .

3.5. Application of the algorithm

We applied the algorithm described above to the problem of finding the set of known pathways in the domain graph extracted from the annotated chemical reactions in the *MetaCyc v18.5* database. The domain graph is an SSG of 9 553 nodes and 75 078 edges. In order to have realistic results, we have applied the algorithm to the search of pathways of length $l \geq 3$, where l is the minimal shortest path length between any source and target node in the annotated pathway. In total, the algorithm has been applied to the reconstruction of 1 981 pathways.

We have tuned the only parameter of our algorithm to the value $\alpha = 0.15$, as generally assumed in the Pagerank literature [Brin and Page \(1998\)](#). However, our results are surprisingly invariant with respect to variations of this parameter. This behavior can be explained as follows. For a jumping constant $\alpha \approx 1$, the expected path length approaches the value 0 and no pathway can be inferred. On the opposite case, for $\alpha \approx 0$, one approaches the limit of a standard random walk, and all pathways, independently on their length, will be indistinguishably found. However, when we are in neither of these two cases, since the node weights in the $\pi_{\triangleright}(S, T)$ vector exponentially decrease as a function of distance from the source and target sets, shorter pathways will *always* be preferred by our algorithm over longer ones. After that, applying pagerank boosting, hub nodes are penalized and only the relevant shortest paths are evidenced. The behavior explained above is displayed in [Fig. 1](#). There we see that the geometric mean of precision (*PPV*) and recall (*TPR*) of our results is very weakly influenced by the choice of the parameter α , the latter two measures being defined as

$$PPV = \frac{TP}{TP + FP}, \quad TPR = \frac{TP}{TP + FN}$$

where TP is the number of non-seed nodes that are present both in the inferred and in the annotated pathway. FP is the number of nodes that are present in the inferred but not in the annotated pathway, and FN is the number of nodes present in the annotated but not in the inferred pathway. This weak dependency makes the highly parallelizable algorithm presented above virtually parameter-free.

3.6. Results and discussion

To our knowledge, the strategy appearing in the bioinformatics literature that relates most closely to ours is the one presented in Faust et al. (2010), where a random walk approach is used to infer metabolic pathways in metabolomics databases. There, the parameter chosen to quantify the quality of the results is the *geometric accuracy*, defined as

$$acc_g = \sqrt{PPV \cdot TPR}.$$

In the following we will use the same parameter in order to simplify the comparison of the results, which is nonetheless difficult for several reasons.

The first is that even though the metabolomic database on which the algorithm has been applied is the same, *MetaCyc* has been updated multiple times since 2010, doubling in size. Therefore, the graph on which we have applied our algorithm is different from the one used in the article to which we are comparing.

The second reason is that the research of pathways in the article mentioned above is carried out by diffusion starting from source subgraphs S of increasing cardinality until the whole annotated pathway is covered. The result for the geometric accuracy is then given by the average of the geometric accuracies for each diffusion. In our case the set S is given by the terminal nodes of the pathway only.

Noting these differences, we have compared our results to those obtained in Faust et al. (2010) selecting out of all the pathways analyzed therein those that are still present in the latest version of the *Metacyc* database. The comparison has been carried out by applying our DF-based algorithm to the inference of all pathways contained in the *Metacyc v18.5* database.

We can reconstruct the value of the geometric accuracy for the algorithm we are comparing to in the case only terminal nodes are taken as seed nodes. We define this parameter as the *base score geometric accuracy*: acc_g^b . In this case, for the pathways that are still present in the updated version of the database, we find a value of $acc_g^b = 0.61 \pm 0.05$. The uncertainty of $\pm 5\%$ is due to difficult decoding of the published data. The geometric accuracy of the inference of metabolic pathways using the DF-based algorithm described above is $acc_g^b = 0.66$.

4. A guideline to use DF for classification

This section is dedicated to further illustrating our method within the context of two classical textual classification problems, and provide a guideline to apply DF to classification tasks.

4.1. Gender detection

We first apply our method to a binary textual classification problem. Given a set of about 20 000 blogs Schler et al. (2006), our goal is to determine the gender of the authors Dubuisson (2014).

We randomly select a subset of 1 000 blogs with an equal proportion of male and female authors and use it as a training set. In both examples, we follow the procedure described in section 2 for generating the association matrices. Setting the density parameter $\gamma = 10^{-2}$ and the function f parameters $\beta = \sigma = 1$, we compute the domain graph G and the set of fingerprints for this subset. The graph G has 23 629 nodes and 5 583 061 edges. At this specific density, the graph forms a single strongly connected component and has a directed diameter equal to 6.

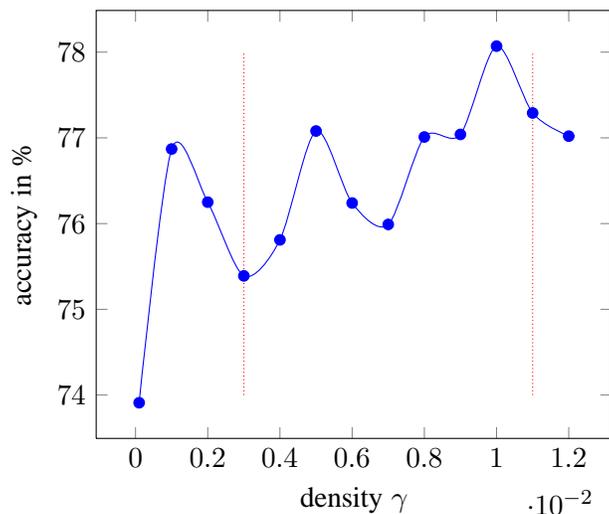


Figure 2: Accuracy of gender detection as a function of the density parameter γ . The left red vertical line marks the minimum density above which the domain graph becomes a strongly connected component (*i.e.*, the diffusion process can reach all parts of the graph). The right red vertical line marks a density above which the accuracy starts decreasing and the computation of fingerprints becomes very costly.

We then perform 10-fold cross validation on 10 random shufflings of the 1000 fingerprints. We get an average accuracy of 79.1% with the AdaBoost meta-algorithm using Decision Tree classifiers [Freund and Schapire \(1995\)](#). By comparison, we get an average accuracy of 74.8% when we use simple bag-of-words (BOW) vectors on the same set of blogs. Note that we use a BOW model not as a model to compete with, but rather as a null model to compare the effects of our dimensionality reduction heuristic (see details in section 5).

Moreover, when we apply our *OPC* dimensionality reduction heuristic to the fingerprint vectors, we observe that the accuracy remains almost constant until we reach a dimension equivalent to 10% of the size of the domain graph. For instance, the accuracy still only reduces slightly to 77.85% when the dimension d is reduced from 23 629 to 3 000.

4.2. Authorship attribution

By using the same set of blogs, we also apply our method to the problem of authorship attribution [Koppel et al. \(2011\)](#); [Seroussi et al. \(2012\)](#). We start by selecting at random 500 blogs containing at least 16 posts of more than 8 tokens each, and we split each of them in two equal number of posts.

For the 500 selected blogs (*i.e.*, 500 classes), we get 11 993 posts containing more than 8 tokens. We first use the aggregated list of tokens of the first halves for generating the domain graph. By choosing $\gamma = 10^{-2}$ and setting $\beta = \sigma = 1$, we get a graph with 17 036 nodes and 2 902 681 edges. We then generate the DFs for each post, and use the fingerprint vectors of the first halves for training 500 ‘one-vs-all’ Random Forest binary classifiers [Breiman \(2001\)](#).

We finally use the fingerprint vectors of the posts in the second parts for testing our classifier, which gives us an accuracy of 27.6%. By comparison, the accuracy is reduced to 24.2% when we use simple BOW vectors.

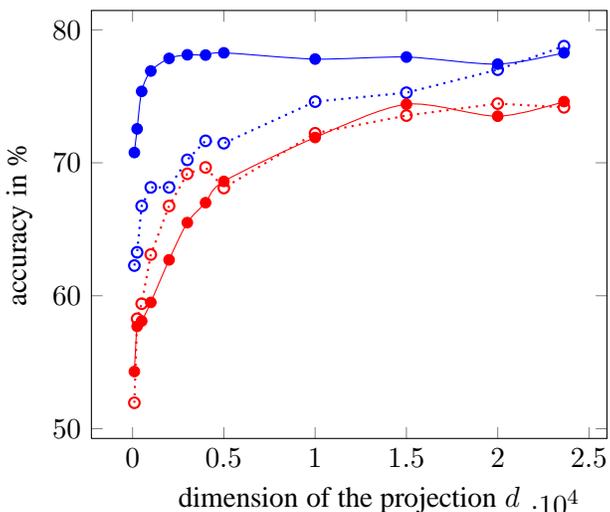


Figure 3: Accuracy of gender detection as a function of the reduced dimension d for $\gamma = 10^{-2}$. The curves are: the diffusive fingerprint method with OPC (solid blue) and the diffusive fingerprint method with random projection (dotted blue), bag of words with OPC (solid red), and bag of words with random projection (dotted red). Note the stability of the fingerprint-OPC result when the dimension d is lowered.

It is to be noted that as the number of classes increases, the computation needed to train the binary classifiers can become very costly. One way to alleviate this problem is to compute instead the mean of the training fingerprint vectors, and use it as a reference fingerprint for each author (*i.e.*, class). When the Manhattan distance is used for classifying the test vectors, we get an accuracy of 22.25% for the DFs and 9.36% for the corresponding BOW vectors.

4.3. Results and discussion

We emphasize that we use text classification problems as a mere illustration of our method. While this does not lead to state-of-the-art accuracies, it shows that the diffusion fingerprints method is generic enough to achieve quite decent results.

4.3.1. GENDER DETECTION

In [Schler et al. \(2006\)](#), the authors use a dataset of 37 478 blogs with an equal number of male and female bloggers to perform gender classification. When applying 10-fold cross validation, they report an accuracy of 72% with content based features only, and 80.1% when stylistic features are also included.

By using many more features (*i.e.*, F-measure, stylistic features, gender preferential features, factor analysis, word classes and POS sequence patterns) and a custom *Ensemble Feature Selection* (EFS) algorithm, [Mukherjee and Liu \(2010\)](#) report a maximum accuracy of 88.56% over a set of 3 100 blogs.

Thus, the accuracy of 79.1% we report seems acceptable, considering that we use no specific tools like n-grams and no additional features.

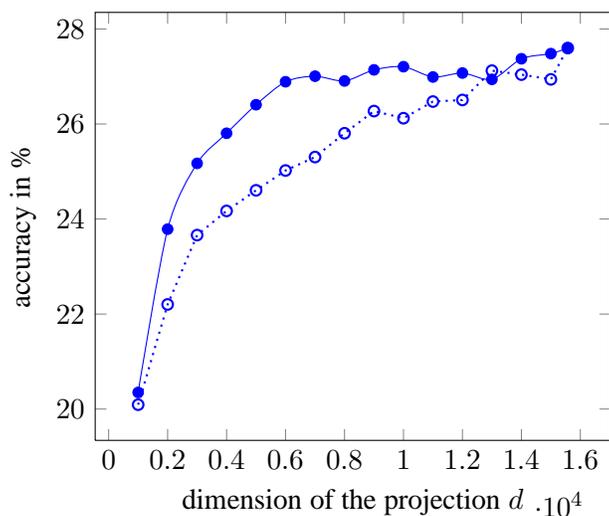


Figure 4: Accuracy of authorship attribution as a function of the reduced dimension d . The curves are: the diffusive fingerprint method with OPC (solid blue) and the diffusive fingerprint method with random projection (dotted blue). Note that BOW results are not indicated here for sake a clarity.

4.3.2. AUTHORSHIP ATTRIBUTION

In [Koppel et al. \(2011\)](#), a dataset of 10 000 blogs is used for testing a method of authorship attribution over a large set of classes. For their first experiment, each blog is split in a reference text of 2 000 words and a snippet of 500 words, both represented as vectors of space-free character 4-gram frequencies. Using cosine similarity, 46% of 1 000 snippets selected randomly get assigned correctly. They also propose an improved algorithm consisting in repeatedly selecting a random fraction of the full feature set, in a similar way to classifier ensemble methods [Bryll et al. \(2003\)](#).

In our experiment, we obtain a lower accuracy of 27.6%. However, it is to be noted that in our case, the blog posts to be classified can contain as few as 8 tokens, which makes the classification task somewhat harder.

5. General discussion

By contrast to some non-linear dimensionality reduction techniques which aim at discovering the structure of a manifold from a set of feature vectors [Roweis and Saul \(2000\)](#); [Tenenbaum et al. \(2000\)](#); [Belkin and Niyogi \(2003\)](#); [Nadler et al. \(2005\)](#), DFs start by extracting the latent topological properties of the provided data under the form of a directed graph. The distributions of the diffusion processes started from the data subsets are then computed as numerical vectors which can subsequently be fed to any classification algorithm.

5.1. Choosing the density parameter γ

We discuss here how to determine the density that provides the best accuracy.

We note that when $\gamma = 0$, there is no diffusion and our method amounts to classifying the personalized vectors associated to each subset in a high dimensional space. In this case, our method simply corresponds to using a bag-of-words model.

On the other hand, when $\gamma = 1$ (then the graph is complete), the initial distributions (*i.e.*, personalized vectors) get flattened, whereas all the other coordinates of the high dimensional fingerprint vectors get assigned the same value. Thus, in this case, the fingerprint stationary distributions resemble an attenuated form of the initial distributions (*i.e.*, the personalized vectors), except that the variance of the data is much lower.

Diffusing in the domain graph from the initial distributions enables to grasp the generic characteristics of the data subsets at the scale of the whole domain graph, and improves average accuracy. A first requirement is thus that the density of the domain graph is large enough (*i.e.*, greater than a critical value γ_c) to enable a diffusion process starting from any subset of nodes to reach all the other parts of the domain graph. This means that γ needs to be chosen to enable the emergence of a giant strongly connected component (SCC). We see for example that the domain graph used previously for gender detection forms a single SCC for $\gamma \geq 3 \cdot 10^{-3}$.

We observe that the average entropy of the fingerprint distributions increases monotonically with the graph density, whereas, at the same time, the average variance decreases. Moreover, the average accuracy appears to be approximately a concave function of the domain graph density. In order to reach the best possible accuracy, we thus need to choose the density parameter $\gamma \in [\gamma_c, 1)$ so that the generated fingerprint distributions retain a high variance, but also exhibit a high average entropy: this corresponds to seeking a trade-off between the expressiveness and the genericity of the generated vectors.

5.2. Dimensionality reduction

After the construction of the diffusion vector, we use the following procedure for dimensionality reduction to lower the computational cost of classifying the generated DF.

Fix some d and let $\Phi(d) : \mathbb{R}^{|\mathcal{T}|} \rightarrow \mathbb{R}^d$ be the orthogonal projection of the $|\mathcal{T}|$ -dimensional fingerprint vectors onto the d -dimensional node subspace \mathbb{R}^d ($d \ll |\mathcal{T}|$) spanned by the d most central nodes of the domain graph $G \equiv G(\gamma)$. To find this projection, we apply the *Pagerank* centrality metric [Brin and Page \(1998\)](#) to G in order to determine the set of the d most central nodes, corresponding a set \mathcal{T}_d of tokens. The projection $\Phi(d)$ is then the $d \times |\mathcal{T}|$ -matrix defined by

$$\Phi(d)_{uv} = \begin{cases} 1 & \text{if } v \in \mathcal{T}_d \\ 0 & \text{otherwise} \end{cases}, \quad u \in \mathcal{T}_d, v \in \mathcal{T}.$$

The intuition is that projecting the fingerprint vectors onto the d most central nodes amounts to embedding the data in a d -dimensional hyperplane of maximum variance, as we will see next.

We call *OPC* (Orthogonal Projection on Central nodes) the projection we just described.

5.3. OPC dimensionality reduction heuristic

Applying dimensionality reduction by projecting orthogonally on the hyperplane spanned by the set of most central nodes limits the decrease of classification accuracy and is computationally very efficient.

By using the *Pagerank* metric as a measure of centrality, this heuristic works well because the highest components of the *Pagerank* vector are highly correlated with the fingerprint coordinates of maximum variance. In the case of the domain graph applied above for gender detection, the Spearman correlation between the *Pagerank* distribution and the vector of fingerprint variance per coordinate is for example equal to 0.92 for $\gamma = 10^{-2}$. This means that projecting on the most central nodes amounts to projecting orthogonally on the hyperplane of maximum variance [Pearson \(1901\)](#); [Hotelling \(1933\)](#).

The diffusion process plays a fundamental role in allowing a graceful decay of the classification accuracy when the dimension of the fingerprint vectors is reduced by applying the heuristic we just described. By extracting the generic characteristics of the data subsets at the scale of the whole

data collection, diffusion fingerprints are in this case far more resilient to dimensionality reduction compared to a bag-of-words model.

5.4. Computational considerations

The first step of our method consists in generating the association matrices for the set of documents $\sigma(k)$. Here, the amount of computation highly depends on whether the association weights are provided in the data (e.g., USF Free Associations dataset Nelson et al. (2004)) or not.

The generation of the domain graph is then straightforward, but we may face a problem if the number of data items is too large for the domain matrix to fit into memory. One potential way to alleviate the problem and to avoid explicitly computing the domain matrix is to generate for each association matrix a labeled unweighted directed graph of density γ in the form of a sparse binary adjacency matrix, and to generate the domain graph by taking the edge-union of the set of labeled subgraphs (i.e., $G = \bigcup E(G_k)$). Note that in this case, it is more difficult to control the overall density γ of the resulting domain graph, as the intersection of the edge sets $E(G_k)$ is not empty.

One may wonder at this point why we ignore the edge weights when computing the DFs. The reason is simply that experimentally, we observed in our examples that it leads to a decrease in accuracy for a higher computing cost. Thus, it seems sufficient to consider only the topological properties of the domain graph.

Computing the DFs amounts to computing Pagerank vectors and many different efficient algorithms Berkhin (2005) have been developed after the Pagerank algorithm was first described Brin and Page (1998). One of the fastest algorithm currently known uses a Monte Carlo-based incremental approach and has a complexity of $\mathcal{O}(\frac{n \ln m}{\epsilon^2})$, where n is the number of nodes, m the number of selected edges and ϵ the desired precision Bahmani et al. (2010). We also observe that it may not be necessary to reach full convergence, as we experimentally get a quasi-maximal accuracy when diffusing for a limited number of steps approximately equal to the directed diameter of the domain graph (e.g., 6 in the case of the domain graphs we used for our text classification experiments).

Once the Pagerank vector of the domain graph has been computed, the heuristic we use for reducing the dimension of the fingerprint vectors is very fast, since it only consists in selecting a subset of the vector indices. We note that there is a very high correlation between the Pagerank vector of the domain graph and the component-wise average of the fingerprint vectors (e.g., Spearman correlation of 0.92 for $\gamma = 10^{-2}$ in the graph used for gender detection). We can thus get a good approximate of the Pagerank vector by computing the average of the fingerprint vectors. Incidentally, we also find a nearly perfect correlation between the component-wise average and component-wise variance of the fingerprint vectors in the studied graphs. This suggests that it may not be necessary after all to compute the Pagerank vector of the domain graph, in order to quickly identify the most central nodes.

6. Conclusion

We have presented a method to generate fingerprint vectors for data exhibiting associative properties. It is based on diffusion processes over a domain graph and shows how to make dimensionality reduction efficient and robust. The numerical vectors that get generated can subsequently be used for classification or clustering.

We applied our method to two classical text classification problems with the same set of blogs. We showed that in both the case of gender detection and of authorship attribution, *Diffusion Fingerprints* provide a better accuracy and a greater resilience to dimension reduction than equivalent bag-of-words vectors.

The adaption and application of our method to the problem of metabolic subgraph extraction led to results that favorably compare with the state of the art in the field, establishing the power

and flexibility of this elegant and conceptually simple method. The hybridization of this algorithm could in principle improve the quality of the results obtained above.

In the context of classification, we believe that DFs may prove useful not just for authorship detection but in many other domains, provided that a domain graph has been constructed. In the case of Free Association datasets for example Nelson et al. (2004); Dubuisson et al. (2013), DF could be used to detect cultural shifts or psychological disorders of certain individuals. When studying social networks (e.g., friendship networks, co-author networks, online social networks, ...), DF could be used to compare the social profiles of different members of a group. DF could also find fruitful application to Word Sense Disambiguation by enabling to generate distinctive contextual fingerprints for the words to be disambiguated.

References

- R. Andersen, F. Chung, and K. Lang. Local graph partitioning using pagerank vectors. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science, FOCS '06*, pages 475–486, Washington, DC, USA, 2006a. IEEE Computer Society. ISBN 0-7695-2720-5. doi: 10.1109/FOCS.2006.44.
- R. Andersen, F. Chung, and K. Lang. Local graph partitioning using pagerank vectors. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pages 475–486. IEEE, 2006b.
- A. V. Antonov, S. Dietmann, P. Wong, and H. W. Mewes. Ticl – a web tool for network-based interpretation of compound lists inferred by high-throughput metabolomics. *FEBS Journal*, 276(7):2084–2094, 2009. ISSN 1742-4658. doi: 10.1111/j.1742-4658.2009.06943.x.
- B. Bahmani, A. Chowdhury, and A. Goel. Fast incremental and personalized pagerank. *Proceedings of the VLDB Endowment*, 4(3):173–184, 2010.
- M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, volume 14, pages 585–591, 2001.
- M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15:1373–1396, 2003.
- P. Berkhin. A survey on pagerank computing. *Internet Mathematics*, 2:73–120, 2005.
- L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. ISSN 0885-6125. doi: 10.1023/A:1010933404324.
- S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.
- R. Bryll, R. Gutierrez-Osuna, and F. Quek. Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets. *Pattern recognition*, 36(6):1291–1302, 2003.
- R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proceedings of the National Academy of Sciences of the United States of America*, 102(21):7426–7431, 2005.
- J. Dubuisson. Diffusion Fingerprints – Application demo to text classification, 2014.
- J. Dubuisson, J.-P. Eckmann, C. Scheible, and H. Schütze. The topology of semantic knowledge. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 669–680, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

- P. Dupont, J. Callut, G. Dooms, J.-N. Monette, Y. Deville, et al. Relevant subgraph extraction from random walks in a graph. *Université catholique de Louvain, UCL/INGI, Number RR, 7*, 2006.
- K. Faust and J. van Helden. Predicting metabolic pathways by sub-network extraction. In Jacques van Helden, Ariane Toussaint, and Denis Thieffry, editors, *Bacterial Molecular Networks*, volume 804 of *Methods in Molecular Biology*, pages 107–130. Springer New York, 2012. ISBN 978-1-61779-360-8. doi: 10.1007/978-1-61779-361-5_7.
- K. Faust, P. Dupont, J. Callut, and J. van Helden. Pathway discovery in metabolic networks by subgraph extraction. *Bioinformatics*, 26(9):1211–1218, 2010. doi: 10.1093/bioinformatics/btq105.
- F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *Knowledge and data engineering, IEEE transactions on*, 19(3):355–369, 2007.
- Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of the Second European Conference on Computational Learning Theory*, EuroCOLT '95, pages 23–37, London, UK, UK, 1995. Springer-Verlag. ISBN 3-540-59119-2.
- M. Gori, A. Pucci, V. Roma, et al. Itemrank: A random-walk based scoring algorithm for recommender engines. In *IJCAI*, volume 7, pages 2766–2771, 2007.
- H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, 1933.
- R. I. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete input spaces. In *ICML*, volume 2, pages 315–322, 2002.
- M. Koppel, J. Schler, and S. Argamon. Authorship attribution in the wild. *Language Resources and Evaluation*, 45(1):83–94, 2011.
- A. Mukherjee and B. Liu. Improving gender classification of blog authors. In *Proceedings of the 2010 conference on Empirical Methods in natural Language Processing*, pages 207–217. Association for Computational Linguistics, 2010.
- B. Nadler, S. Lafon, R. Coifman, and I. Kevrekidis. Diffusion maps, spectral clustering and eigenfunctions of fokker-planck operators. In *Advances in Neural Information Processing Systems 18*, pages 955–962. MIT Press, 2005.
- D. Nelson, C. McEvoy, and T. Schreiber. The University of South Florida free association, rhyme, and word fragment norms. *Behavior Research Methods, Instruments, & Computers*, 36(3):402–407, 2004.
- L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. 1999.
- K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2: 559–572, 1901.
- B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM, 2014.
- P. Pons and M. Latapy. Computing communities in large networks using random walks. *J. Graph Algorithms Appl.*, 10(2):191–218, 2006.

- D. Rajagopalan and P. Agarwal. Inferring pathways from gene lists using a literature-derived network of biological relationships. *Bioinformatics*, 21(6):788–793, 2005. doi: 10.1093/bioinformatics/bti069.
- S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.
- J. Schler, M. Koppel, S. Argamon, and J. W. Pennebaker. Effects of age and gender on blogging. In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, volume 6, pages 199–205, 2006.
- B. Schölkopf and A. J. Smola. *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT press, 2002.
- M. S. Scott, T. Perkins, S. Bunnell, F. Pepin, D. Y. Thomas, and M. Hallett. Identifying regulatory subnetworks for a set of genes. *Molecular & Cellular Proteomics*, 4(5):683–692, 2005.
- Y. Seroussi, F. Bohnert, and I. Zukerman. Authorship attribution with author-aware topic models. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ACL '12, pages 264–269, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.
- J. Shi and J. Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.
- J. Tenenbaum, V. Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.
- A. Zien, R. Küffner, R. Zimmer, and T. Lengauer. Analysis of gene expression data with pathway scores. In *Ismb*, volume 8, pages 407–417, 2000.