Diffusion Fingerprints

Jimmy Dubuisson Jean-Pierre Eckmann
Département de Physique Théorique and Section de Mathématiques
Université de Genève
{jimmy.dubuisson,jean-pierre.eckmann}@unige.ch

Abstract

We introduce a new method for classifying and clustering data exhibiting associative properties. By means of graph theoretical tools, we show how to generate *diffusion fingerprints* for each subset of the data collection. We then propose a simple and computationally efficient technique for dimensionality reduction. Throughout this paper, we apply our method to the problem of classifying a corpus of text documents and compare it to other methods.

Keywords graph theory, machine learning, natural language processing, pagerank, dimensionality reduction

Introduction

Our method consists in a combination of several simple steps. Starting from a data collection, we first construct association matrices, and then form a directed graph from it, using a threshold. A diffusion process on this graph explores the vicinities of each node of the graph. This gives vectors of nodes which are related. Since these vectors are in a high-dimensional space (the dimension being, for example, the number of different tokens in texts) we perform a dimension reduction. It is these two last steps which make our method computationally efficient.

The association matrices

We consider a *data collection* $\Sigma = \{\sigma(1), \sigma(2), \dots\}$ of "documents," where each *document* is viewed as a sequence of data items (one should think of "tokens"). The set of different tokens appearing in Σ is called $\mathcal{T}$. We denote $|\mathcal{T}|$ the cardinality of $\mathcal{T}$.

An example: computing words collocation

To illustrate our terminology, 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

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

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

We conduct our experiments with the following families of functions:

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

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. We take $\beta = \sigma = 1$. (These functions might be changed somewhat depending on the study one wants to perform.)

The domain graph

Having generated, for each document k, the association matrix $K(k)$ as described above, we next define a *domain* *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]

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.

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* 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 $\mathcal{T}(k) \backslash \mathcal{T}'(k)$, the better the generated fingerprint represents $\sigma(k)$ within the context of the domain graph.

Let M be the *transition probability matrix* defined by:

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

where D is the diagonal matrix of the degrees of G. We compute the diffusion fingerprint of document $\sigma(k)$ as the *personalized Pagerank* (Andersen et al., 2006) ppr_k defined recursively by:

$$\mathrm{ppr}_k(t+1) = \alpha v_k + (1-\alpha)\,\mathrm{ppr}_k(t) M \; , \qquad (1)$$

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

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

with $f_k(u)$ the frequency of data item u in document $\sigma(k)$. In principle, we define

$$\pi(k) = \lim_{t\to\infty} \mathrm{ppr}_k(t)$$

We call $\pi(k)$ the *stationary diffusion fingerprint* of document $\sigma(k)$.[3]

Dimensionality reduction

After the construction of the diffusion vector, we describe (and use) a simple and yet efficient procedure for dimensionality reduction to lower the computational cost of classifying the generated diffusion fingerprints.

Fix some d and let $\Phi(d) : \mathbb{R}^{|\mathcal{T}|} \to S^d$ be the orthogonal projection of the $|\mathcal{T}|$-dimensional fingerprint vectors onto the d-dimensional node subspace S^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. There corresponds a set $\mathcal{T}_d$ of tokens to this set. 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.

Application to text classification

Gender detection

We first apply our method to a binary 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. Setting the density parameter $\gamma = 10^{-2}$, 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 equals to 6.

We then perform 10-fold cross validation on 10 random shufflings of the 1'000 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 vectors on the same set of blogs.

Moreover, when we apply the OPC 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.

[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 $\sigma(k)$.

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

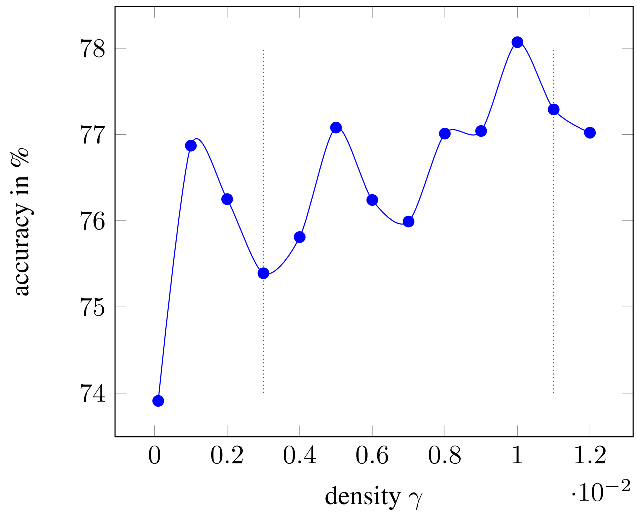

Figure 1: Accuracy of gender detection as a function of γ. The left red vertical line marks the minimum density above which the domain graph becomes a strongly connected component (*i.e.*, the diffusion process can reach all parts of the graph). The right red vertical line marks a density above which the accuracy starts decreasing and the computation of fingerprints becomes very costly.

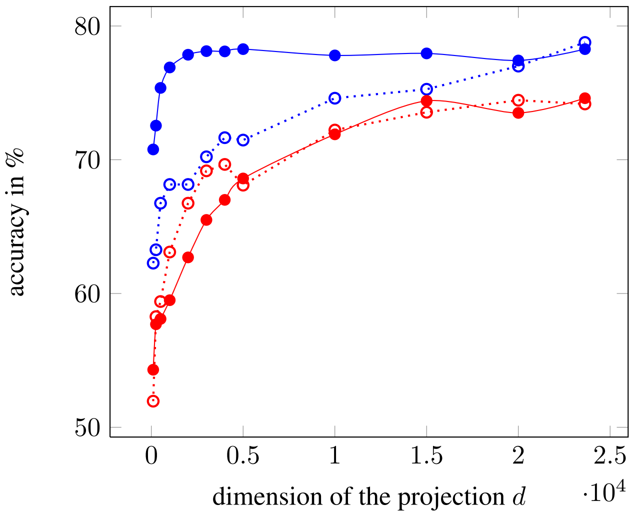

Figure 2: Accuracy of gender detection as a function of the reduced dimension d for $\gamma = 10^{-2}$. The curves are: The diffusive fingerprint method, followed by OPC (thick blue), the diffusive fingerprint method, followed by random projection (dotted blue). The red curves are the analogues, but for the BOW instead of the fingerprint. Note the stability of the fingerprint-OPC result when the dimension d is lowered.

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 halfs for generating the domain graph. By choosing $\gamma = 10^{-2}$, we get a domain graph with 17'036 nodes and 2'902'681 edges. We then generate the diffusion fingerprints for each post, and use the fingerprint vectors of the first parts 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.

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 the problem is to compute instead the center of the training fingerprint vectors, and use it as a reference fingerprint for each author (*i.e.*, class). By using the Manhattan distance for classifying the test vectors, we get an accuracy of 22.25% for the diffusion fingerprints and 9.36% for the corresponding BOW vectors.

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 (Nadler et al., 2005; Belkin and Niyogi, 2003; Roweis and Saul, 2000; Tenenbaum et al., 2000), diffusion fingerprints starts by extracting the latent topological properties of the provided data under the form of a directed graph. The distributions of diffusion processes started from the data subsets are then computed as numerical vectors which can be used subsequently to feed to any classification algorithm.

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

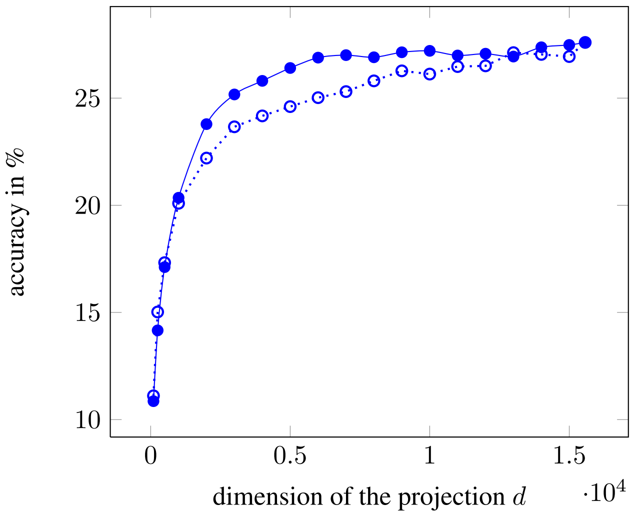

Figure 3: Accuracy of authorship attribution as a function of the reduced dimension d. The curves are: The diffusive fingerprint method, followed by OPC (thick blue), the diffusive fingerprint method, followed by random projection (dotted blue).

distributions, 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 stochastic vectors 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 fingerprints 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.

OPC dimensionality reduction heuristic

Applying dimensionality reduction by projecting orthogonally on the hyperplane spanned by the set of most central nodes limits efficiently 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 previously 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.

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 the data being considered: the association weights may be provided in the data (*e.g.*, USF Free Associations dataset (Nelson et al., 2004)) or it may be necessary to find a way to extract the latent association values existing between data items.

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.*, $G = \cup E(G_k)$). 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 diffusion fingerprints. 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 diffusion fingerprints amounts to computing Pagerank vectors and different efficient iterative methods (Berkhin, 2005) were developed since the Pagerank algorithm was first described (Brin and Page, 1998). Moreover, we observe that it may not be necessary to reach full convergence, as we 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 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 $\gamma = 10^{-2}$). We can thus get a good approximate of the Pagerank vector by computing the average of the fingerprint vectors. Incidentally, we also find a perfect correlation between the component-wise average and component-wise variance of the fingerprint vectors, for a density $\gamma = 10^{-2}$. 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.

Conclusion

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

Our method has been applied 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.

We believe that Diffusion Fingerprints (DF) may prove useful not just for text classification 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. Finally, we think DF could be successfully applied to automatic recommendation systems. For an online shop, the idea would be for example to leverage a domain graph built from the purchase history of its users.

References

R. Andersen, F. Chung, and K. Lang. 2006. 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. IEEE Computer Society.

M. Belkin and P. Niyogi. 2003. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15:1373–1396.

P. Berkhin. 2005. A survey on pagerank computing. *Internet Mathematics*, 2:73–120.

L. Breiman. 2001. Random forests. *Machine Learning*, 45(1):5–32.

S. Brin and L. Page. 1998. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117.

J. Dubuisson, J.-P. Eckmann, C. Scheible, and H. Schütze. 2013. 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. Association for Computational Linguistics.

J. Dubuisson. 2014. Diffusion Fingerprints – Application demo to text classification. `https://github.com/jimbotonic/df`.

Y. Freund and R. Schapire. 1995. 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. Springer-Verlag.

H. Hotelling. 1933. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441.

M. Koppel, J. Schler, and S. Argamon. 2011. Authorship attribution in the wild. *Language Resources and Evaluation*, 45(1):83–94.

B. Nadler, S. Lafon, R. Coifman, and I. Kevrekidis. 2005. Diffusion maps, spectral clustering and eigenfunctions of fokker-planck operators. In *in Advances in Neural Information Processing Systems 18*, pages 955–962. MIT Press.

D. Nelson, C. McEvoy, and T. Schreiber. 2004. The University of South Florida free association, rhyme, and word fragment norms. *Behavior Research Methods, Instruments, & Computers*, 36(3):402–407.

K. Pearson. 1901. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2:559–572.

S. Roweis and L. Saul. 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326.

J. Schler, M. Koppel, S. Argamon, and J. Pennbaker. 2006. Effects of age and gender on blogging. In *Proceedings of 2006 AAAI Spring Symposium on Computational Approaches for Analyzing Weblogs*.

Y. Seroussi, F. Bohnert, and I. Zukerman. 2012. 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. Association for Computational Linguistics.

J. Tenenbaum, V. Silva, and J. Langford. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323.