Programming Languages in GitHub: A Visualization in Hyperbolic Plane

Dorota Celińska
Faculty of Economic Sciences
University of Warsaw, Poland
dcelinska@wne.uw.edu.pl

Eryk Kopczyński
Institute of Informatics
University of Warsaw, Poland
erykk@mimuw.edu.pl

Abstract

GitHub is nowadays the largest software repository hosting service that incorporates social media functionalities to writing and assessing source code. We visualize the weighted network of programming languages used by the registered users of GitHub as of December 2016. There is an edge in the graph between language A and B if and only if A and B are both used by the same user in any of her repositories. The weight depends on the number of times such edge appears. Our visualization utilizes hyperbolic geometry, which is intrinsic to networks based on similarity and popularity. RogueViz, a novel tool, based on the Open Source game HyperRogue, is used to map the network and navigate the hyperbolic graph.

Keywords: GitHub, hyperbolic plane, visualization, popularity, similarity, programming languages

Introduction

Programming languages can be seen not only as tools for directing computers, but also as a means of exchanging ideas among developers, which makes them a means of communication. As many other information goods, programming languages are subjects to network externalities: utility derived from adopting one of them depends on the number of its existing users. Choosing the main programming language for an application appears as one of the crucial development decisions – for languages popular in a given area, there are more resources available online to help with the common challenges, and there are more experienced developers potentially willing to contribute to a project. Therefore understanding of the relationships among usage of the programming languages is of vast importance in software development.

The Open Source software license allows the end users to study, modify, and distribute the publicly accessible source code to anyone, and for any purpose. These principles effectively lower costs of participating in projects for volunteers, creating both a friendly environment for beginners, and the signaling opportunities for professionals. Despite being relatively new website, GitHub has recently become the largest repository hosting service related to the development of Open Source software, featuring elements of a social network service. As of Dec 31, 2016, GitHub has more than 15 million of registered users and hosts over 40 million of repositories. The popularity of the service and the relative ease of obtaining data about events that occur among its users make GitHub a rich and promising source of information for researchers.

Dataset and visualization

In this article we present a visualization of a weighted network of programming languages used by the registered users of GitHub as of Dec 31, 2016. Since GitHub data is huge and the service is continuously evolving, the complete download of the data is impossible (Gousios and Spinellis 2012). To minimize the number of missing observations, we utilize a dataset combined from three sources: GHTorrent project (Gousios 2013), GitHub Archive project (Grigorik 2012) and our own database obtained by web-scrapping GitHub. The resulting dataset contains information about the activity of 10,620,313 users in 42,636,285 repositories and the usage of 321 programming languages (based on GitHub's definition of a programming language). GitHub offers both paid plans for private repositories, and free public repositories on the same account, yet information about private repositories is not available. This explains the difference between the number of users in our sample, and the number of actual registered users.

For each user u and programming language l, we count $n_{u,l}$, the number of repositories belonging to u where the language l is used. From this data, we create a weighted undirected graph $\mathcal{G}_l$, where vertices correspond to programming languages. There is an edge in $\mathcal{G}_l$ between language l_A and l_B if and only if l_A and l_B are both used by the same user in any of her repositories. The weight of the edge is computed using the formula $w_{l_A,l_B} = \sum_{u \in U} \frac{n_{u,l_A} n_{u,l_B}}{N_u}$, where $N_u = \sum_{l_A \neq l_B} n_{u,l_A} n_{u,l_B}$. This way, for each user u using at least two languages, total weight 1 is distributed between the edges connecting languages used by u, with more weight given to the languages used more frequently by u.

Using hyperbolic geometry

We decided to embed $\mathcal{G}_l$ in the hyperbolic plane (Munzner 1998). In hyperbolic plane, the amount of space in distance d from the given point is exponential in d. Such geometry

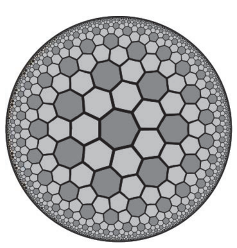

Figure 1: Order-7 truncated triangular tiling.

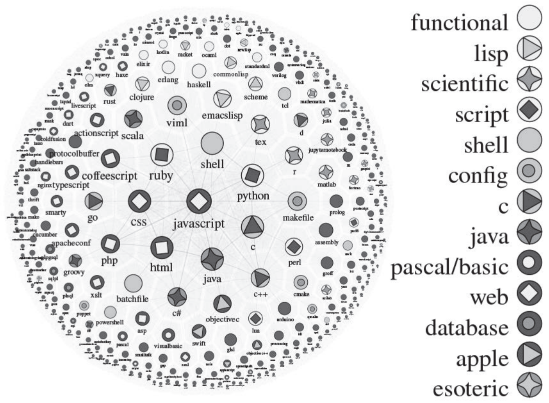

Figure 2: Our visualization (grayscale printable version).[1]

is intrinsic in many real world networks based on similarity and popularity (Papadopoulos et al. 2012). An efficient embedding algorithm has been recently shown (Bläsius et al. 2016). Unfortunately, this algorithm does not deal with weighted networks, and it did not give visually appealing results in our experiments. Furthermore, no powerful tools exist to visualize such embeddings of graphs.

We use our own method to embed the graph. Our implementation is derived from the game HyperRogue, which is a puzzle/roguelike game set in the hyperbolic plane. HyperRogue is Open Source software, and its publicly accessible source code contains a very rich set of powerful and efficient tools for representing, visualizing, navigating, and performing computations in the hyperbolic plane, both continuous and based on the tiling. HyperRogue allows wandering freely through the exponential expanse of hyperbolic plane, thus giving hope of working interactively even with very large hyperbolic graphs.

HyperRogue uses the order-7 truncated triangular tiling of hyperbolic plane, shown in Figure 1. Our algorithm tries to map each vertex of our graph into one of hexagonal or heptagonal faces of the tiling. It attempts to minimize the value $V = \sum_{e \in E} w(e)^{0.4} d(e)$, where E is the set of edges in our network, $w(e)$ is the weight of the edge, and $d(e)$ is the distance between its endpoints in the current mapping. This minimization is performed by an algorithm based on simulated annealing.

Discussion

How important is hyperbolic geometry for our visualization (Fig. 2)? First, since we use the Poincaré model, vertices further away from the center, i.e., rarely used languages, are smaller on the screen. This allows to investigate the popularity of the programming languages. What is more, languages sharing same properties are mapped close together. One can easily notice the communities of ”web-programming” languages, data science and research oriented ones, and functional ones. Eventually, the result is (in our opinion) aesthetically more pleasing than its Euclidean counterpart.

To measure the quality of our mapping in a more objective way, we have computed how good is our mapping of vertices at predicting existence of an edge between two languages, based on the distance between them. Log-likelihood is computed using the formula

$$\sum_i p_i \log(p_i) + q_i \log(q_i) - (p_i + q_i) \log(p_i + q_i)$$

where p_i is the number of pairs of nodes at distance i which are connected by an edge of weight at least 0.1, and q_i is the number of such pairs which are not connected. We obtain $L = -20544$ for our hyperbolic mapping, and $L = -23713$ for a mapping obtained by the same algorithm over the usual Euclidean hexagonal grid, which means that our hyperbolic mapping is significantly better at predicting the connections between languages. Note that our algorithm has been optimizing V rather than L, and these two metrics are quite different – first, L does not take weights into account, and second, to minimize V it is always worth to move points closer even if there are no connections between them, even though this might not be optimal for predictions.

Acknowledgment

This work was supported by the National Science Centre, Poland, grant DEC-2016/21/N/HS4/02100.

References

Bläsius, T.; Friedrich, T.; Krohmer, A.; and Laue, S. 2016. Efficient embedding of scale-free graphs in the hyperbolic plane. In *European Symposium on Algorithms (ESA)*, 16:1–16:18.

Gousios, G., and Spinellis, D. 2012. GHTorrent: Github’s data from a firehose. In Lanza, M.; Penta, M. D.; and Xie, T., eds., *9th IEEE Working Conference on Mining Software Repositories (MSR)*, 12–21. IEEE.

Gousios, G. 2013. The ghtorrent dataset and tool suite. In *Proceedings of the 10th Working Conference on Mining Software Repositories*, MSR ’13, 233–236. Piscataway, NJ, USA: IEEE Press.

Grigorik, I. 2012. Github Archive. Accessed Dec 5, 2016. https://www.githubarchive.org/.

Munzner, T. 1998. Exploring large graphs in 3d hyperbolic space. *IEEE Computer Graphics and Applications* 18(4):18–23.

Papadopoulos, F.; Kitsak, M.; Serrano, M. A.; Boguñá, M.; and Krioukov, D. 2012. Popularity versus Similarity in Growing Networks. *Nature* 489:537–540.

[1] Order-3 heptagonal tiling, Euclidean hexagonal grid and an interactive version of this visualization can be accessed here: http://coin.wne.uw.edu.pl/dcelinska/en/pages/rogueviz-langs.html.