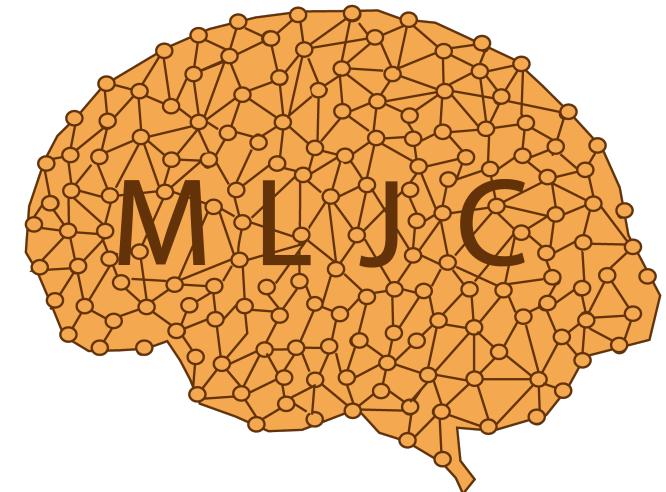


Machine Learning Journal Club: Open Learning for Open Science

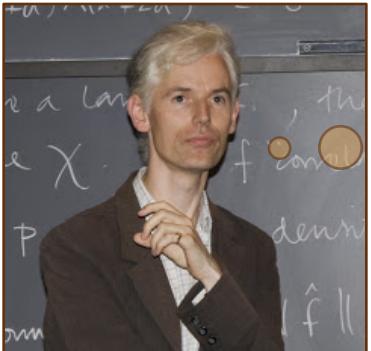
Simone Azeglio
Jacopo Pasqualini



simone.azeglio@edu.unito.it
jacopo.pasqualini@edu.unito.it

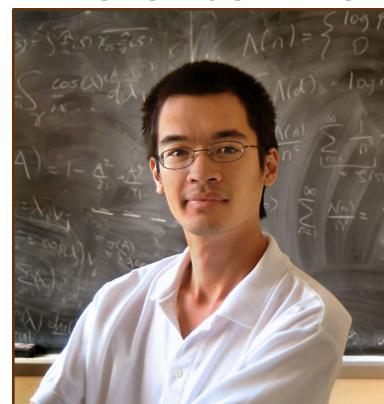
Open Science: is Polymath Project a New Paradigm?

Tim Gowers



Is massively collaborative mathematics possible?

Terence Tao



Gowers's Weblog

Mathematics related discussions

Is massively collaborative mat

Of course, one might say, there are lots of problems that lend themselves to huge collaboration. One might think of the proof of the classification of finite simple groups, or of a rather different kind of example, such as the search for a new largest prime carried out during the Great Internet Mersenne Prime Search, involving thousands of PCs around the world. But my question is: what about the solving of a problem that

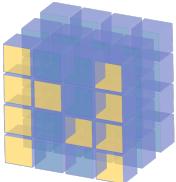
can only be solved by one person? What about the solving of a problem that can only be solved by splitting it up into a vast number of subtasks? Are such problems best tackled by n people for some n that belongs to the set $\{2, 3\}$?

<https://www.scientificamerican.com/article/problem-solved-lol/>

MLJC: a Contribution to Open Science

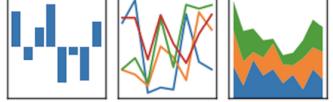
MLJC as a learning experience: Machine Learners

- Python for Scientific Computing



NumPy

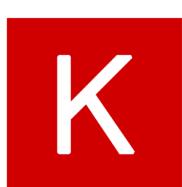
pandas



matplotlib



- Hands On Machine & Deep Learning



Keras



PyTorch



TensorFlow^{2.0}



Machine Learning
Journal Club Unito
MLJCUnito

MLJC: a Contribution to Open Science

MLJC as a research experience: **Social Resistance**

A "quantitative" way to understand and explain reality when **Fake** and **Real** seem to be mixed up

DeepFakes



FakeNews

**Fake News Detection on Social Media using
Geometric Deep Learning**

Federico Monti^{1,2} Fabrizio Frasca^{1,2} Davide Eynard^{1,2} Damon Mannion^{1,2}

Michael M. Bronstein^{1,2,3}

¹Fabula AI
United Kingdom

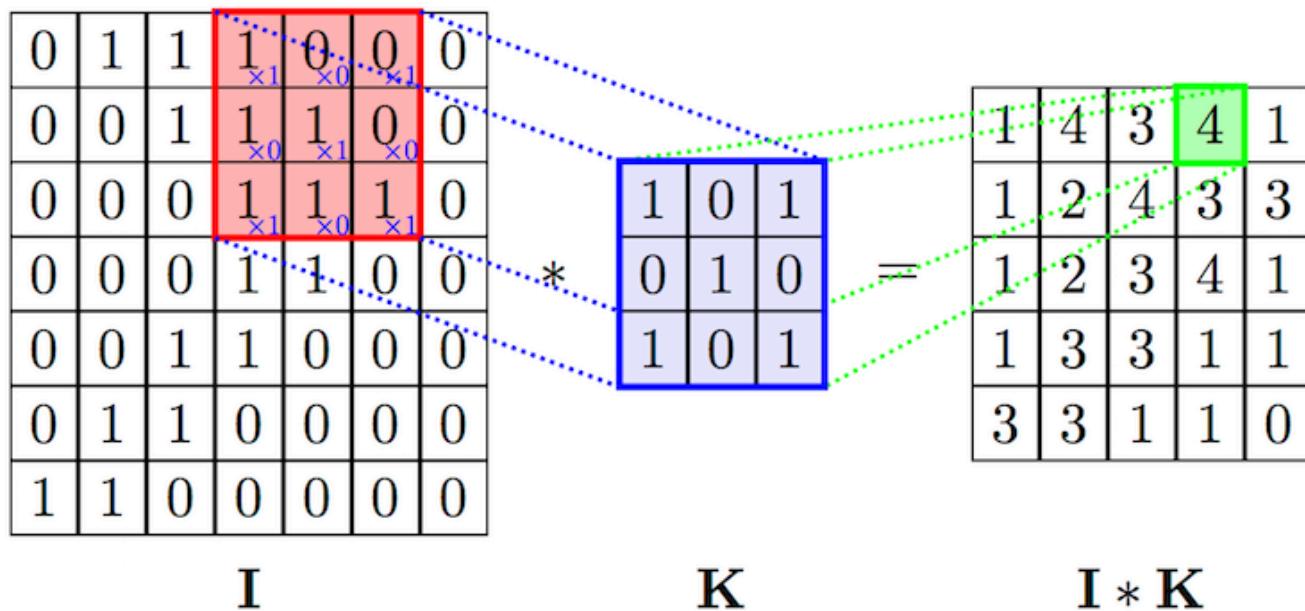
²USI Lugano
Switzerland

³Imperial College
United Kingdom

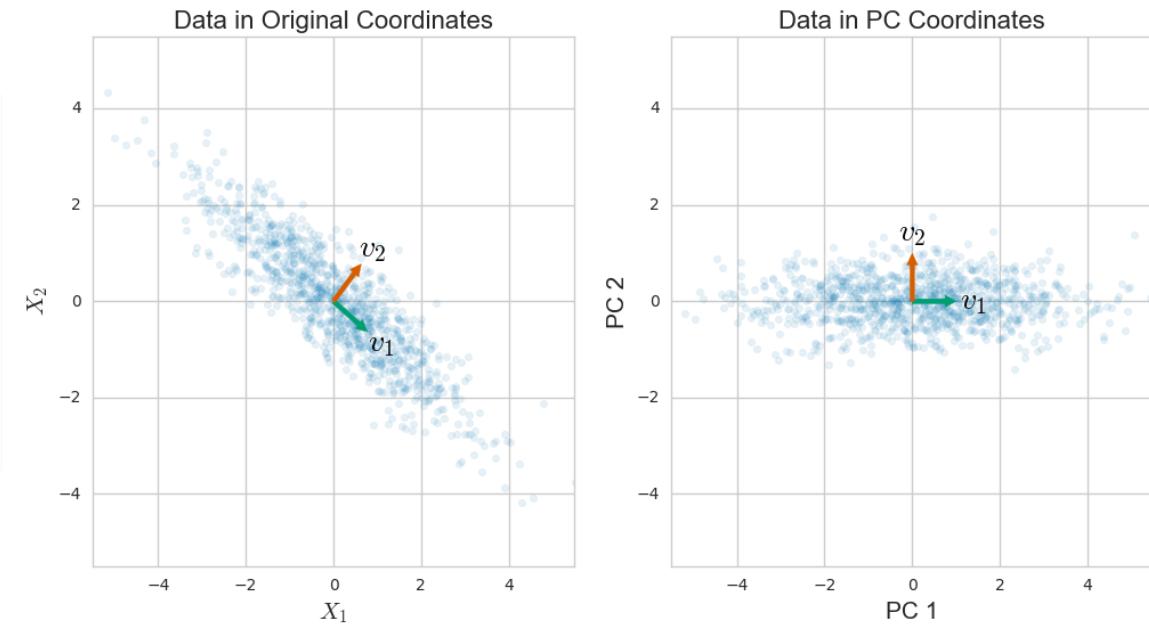
Geometric Priors in ML

In ML there is an ubiquitous **prior** :
data and models are always embedded in a **Euclidean space**

Convolutional Neural Networks (**CNN**)



Principal Component Analysis (**PCA**)



Non Euclidean Data: Graphs

Why?

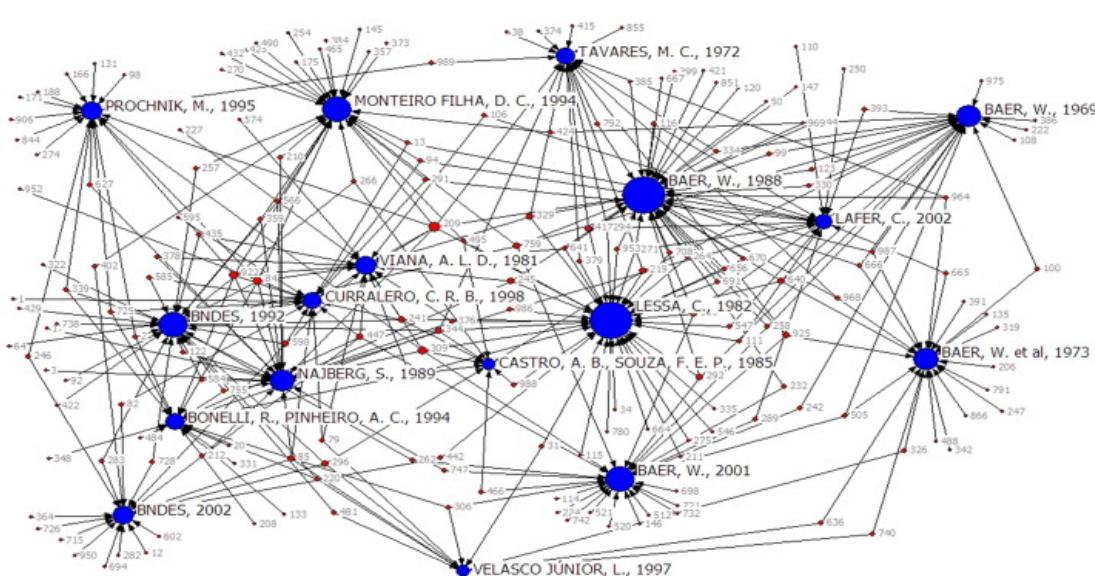
A lot of real data can be represented or are embedded in structures like **Manifolds or Graphs**

$$\mathcal{G} = (\mathcal{V}, \mathcal{E})$$

$$\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$$

$$|\mathcal{V}| = N \quad |\mathcal{E}| = L \leq N^2$$

Can we define **Convolution** on such objects?



Nodes state

$$a = (a_1, \dots, a_N) \in \mathbb{R}^N$$

Links state

$$W = W^T$$

$$W \in \mathbb{R}^{N \times N}$$

Some Definitions

We can define **Hilbert Spaces** by introducing inner products of Graph functions

$$\langle f, g \rangle_{L^2(\mathcal{V})} = \sum_{i \in \mathcal{V}} a_i f_i g_i$$

$$\langle F, G \rangle_{L^2(\mathcal{E})} = \sum_{(i,j) \in \mathcal{E}} w_{i,j} F_{i,j} G_{i,j}$$

In the same way, **operators** on such spaces

$$\nabla : L^2(\mathcal{V}) \longrightarrow L^2(\mathcal{E}) \quad (\nabla f)_{i,j} = f_i - f_j$$

$$div : L^2(\mathcal{E}) \longrightarrow L^2(\mathcal{V}) \quad (div F)_i = \frac{1}{a_i} \sum_{(i,j) \in \mathcal{E}} w_{i,j} F_{i,j}$$

Some Definitions II

We can write the **Laplace operator** and it's equation in matrix form

$$\operatorname{div} \circ \nabla = (\Delta f)_i = \frac{1}{a_i} \sum_{(i,j) \in \mathcal{E}} w_{i,j} (f_i - f_j)$$

$$\Delta f = A^{-1}(D - W)f$$

$$A = \operatorname{diag}(a_1, \dots, a_N) \quad D = \operatorname{diag}\left(\sum_{j:j \neq i} w_{i,j}\right)$$

Solving Laplace Problem on Graphs

It is possible to solve the **Laplace eigenproblem** as an optimization problem :

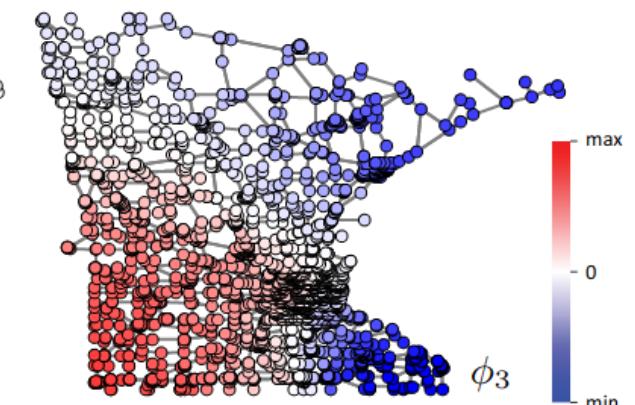
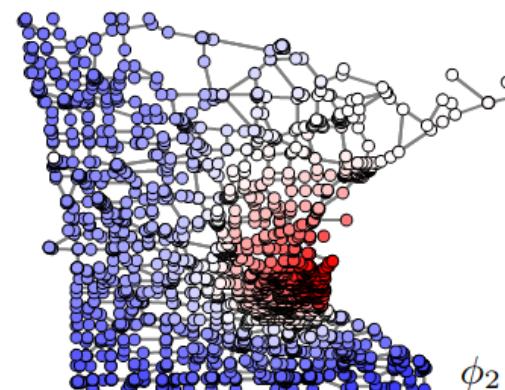
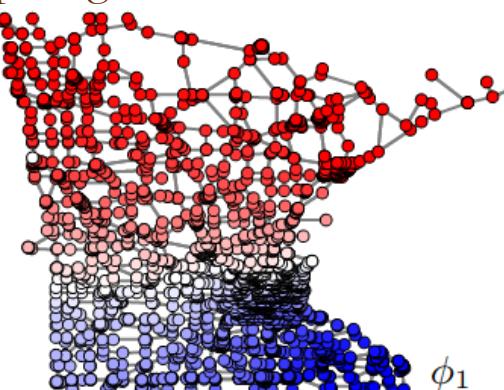
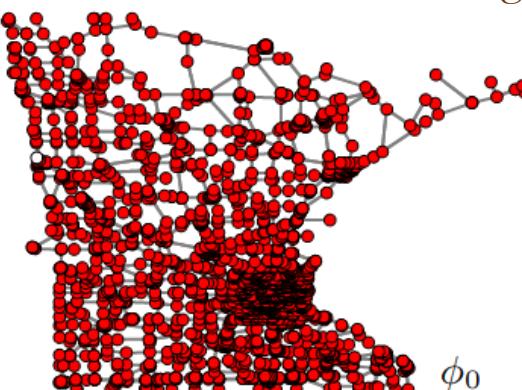
$$\Delta\phi_i = \lambda\phi_i$$

$$\min_{\phi_i: \|\phi_i\|=1} \{\mathcal{E}_{Dir}(\phi_i)\}$$

$$\mathcal{E}_{Dir} = \langle \phi, \Delta\phi \rangle_{L^2(\mathcal{V})}$$

$$\phi_i \perp span(\phi_0, \dots, \phi_{i-1})$$

First four Minnesota road graph eigenfunctions



M.Bronstein et al, IEEE SIC PROG MAG (2016)

Convolution on Graphs

And use these results to design **CNNs** that work on graphs

$$g_\theta * x = \Phi g_\theta \Phi^T x \approx \sum_{k=0}^K \theta'_k T_k(\Delta)$$

Spectral Convolutional Layer (1st order approx.)

$$H^{(l+1)}(g_\theta * H^{(l)}) = \sigma\left(\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} H^{(l)} \Theta\right)$$

e.g. Zachary's
Karate Club

