Interaction between Concepts: a Pseudolikelihood Approach

Author Giacomo Giudice

Supervisors Alfredo Pasquarello
Paolo De Los Rios

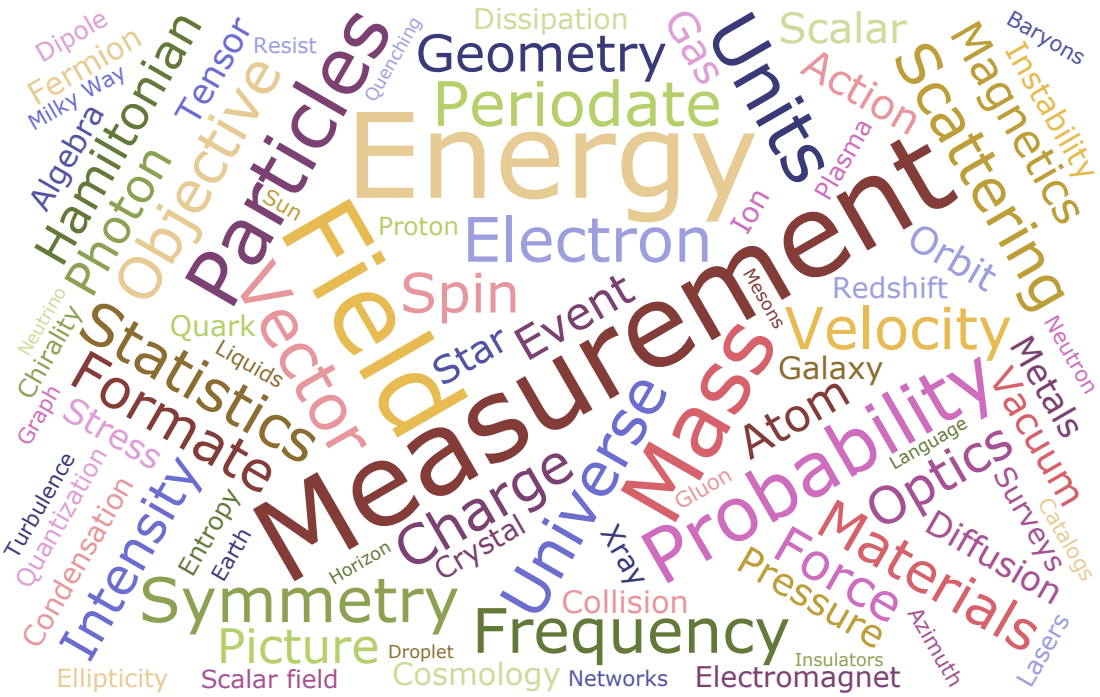

Figure 1: The most common concepts in the corpus, with size representing frequency.

Abstract

In this paper we derive an *effective ontology* by estimating the *interaction strength* between scientific concepts from their use in the scientific literature. This is done by applying the principle of maximum entropy to map the system onto an *Ising model* — a well known model of ferromagnetism in statistical physics — that is then solved with a pseudolikelihood maximization approach. The network thus formed is partitioned with *community detection* to find clusters of concepts related to each other.

Acknowledgments

I would like to express my gratitude to Professor Paolo De Los Rios for proposing and supervising this project — his guidance has been very valuable. I would also like to thank Duccio Malinverni for his support, advice, and continuous availability throughout the project, Finally, I would like to express my appreciation for the networking and clustering techniques suggested by Andrea Martini.

Contents

1 Introduction

Scientific papers (available e.g. from arXiv[1]) can be classified according to the concepts they contain, and studying their relationships can lead to a useful and automatic hierarchical grouping of papers. An issue that has received less attention is the way concepts are related with each other. Intuitively, concepts that often appear together in papers are related to each other, but of course the precise intensity of this connection is not known. Moreover, the ensemble of all these connections defines a network, whose structure is presently not known. The relation between concepts has been already explored in other domains. However, connections are typically defined a priori according to their semantic definitions, and are just binary — namely, there is or there isn't a connection. This relation network is what we call an *ontology* [6].

We propose to analyze the corpus of 54198 scientific articles published in arXiv during 2013 in physics (see table 1). From these papers *concepts* have been extracted and have passed a procedure of collaborative validation from physicists using ScienceWise[2]. The number of papers with at least one article is 52979. We do not consider directly the correlation between concepts — it can be trivially calculated – but rather to try find insights about the causal ties. This is achieved by adapting a method of interaction–strength inference that has been recently introduced by statistical physicists and that is rapidly gaining ground in different domains such as biology, computer science, social sciences, neuroscience, etc.

In order to do so we derive an *Ising model* for the concepts in the articles, and solve an optimization problem to deduce the parameters that best determine the interaction between concepts. To continue the analogy with statistical physics, we want to estimate the force of attraction or repulsion between concepts. This has been done already by Floretta [5] using mean-field inversion. We propose a method that uses *pseudolikelihood maximization* instead.

We then used the interactions to form a network, where each node represents a concept. Techniques to subdivide network into highly connected groups of nodes have been developed in graph theory and complex networks [9]. By exploiting these techniques we attempt to construct a division of the network based on *community detection*.

[1] http://arxiv.org/
[2] http://sciencewise.info/

Abbreviation	Papers	Percentage [%]
astro-ph	12475	23.4
cond-mat	12749	24.0
gr-qc	2274	4.3
hep-ex	943	1.8
hep-lat	726	1.4
hep-ph	4630	8.7
hep-th	3367	6.3
math-ph	1771	3.3
nlin	880	1.7
nucl-ex	529	1.0
nucl-th	1294	2.4
physics	7476	14.1
quant-ph	4044	7.6
Total	54198	100.0

Table 1: Composition of our corpus: number and percentage of articles in each arXiv category.

2 The Ising Model

2.1 Definitions

In order to formulate the problem we have adapted a method to detect protein folding from sequence alignments — in particular we follow very closely the derivation by Ekeberg [4].

The presence of a concept in an article can be viewed as a binary variable — it is present or not. In analogy to the study of ferromagnetism in statistical physics, we represent this as a spin state, that can assume the value 0 or 1.[3] Using n for the number of concepts and N for the number of articles, we can represent an article by a set of n spins: $\mathbf{s} = (s_1, s_2, \ldots, s_n)$ where $s_i = \{0, 1\} \;\; \forall\, i = 1, \ldots, n$.[4] Therefore, the entire corpus of articles can be viewed as a N-tuple of these sets, represented in

[3]This choice is completely arbitrary, as one can label the spins with whichever pair of values one chooses, and the equations remain the same. For example, one can prove that any result is invariant under the transformation $\mathbf{s} \mapsto \alpha\mathbf{s} + \beta, \;\; \alpha, \beta \in \mathbb{R}$, with appropriate modifications of the field and coupling parameters.

[4]Unless stated otherwise, the indices $i, j, k \in 1, \ldots, n$.

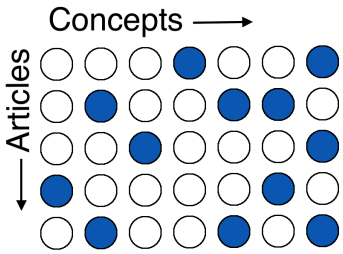

Figure 2: *Spin matrix: full dots mark concepts that are present in an article.*

matrix form as

$$s \equiv \{\mathbf{s}\}_{a=1}^{N} \equiv \begin{pmatrix} s_1^{(1)} & s_2^{(1)} & \dots & s_n^{(1)} \\ s_1^{(2)} & s_2^{(2)} & \dots & s_n^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ s_1^{(N)} & s_2^{(N)} & \dots & s_n^{(N)} \end{pmatrix} \tag{1}$$

A visual example is presented in fig. 2. We define the individual and pairwise *frequencies* along the columns to be

$$f_i \equiv \langle s_i \rangle = \frac{1}{N} \sum_{a=1}^{N} s_i^{(a)} \tag{2a}$$

$$f_{ij} \equiv \langle s_i s_j \rangle = \frac{1}{N} \sum_{a=1}^{N} s_i^{(a)} s_j^{(a)} \tag{2b}$$

The f_i represent the probability the i-th concept is present. Instead the pairwise frequencies f_{ij} represent the the i-th and j-th appear together in an article. The total number of concepts we work with is 13174; a visualization of the most common concepts in the entire corpus is presented in fig. 1.

2.2 Principle of Maximum Entropy

We are interested in deriving a probabilistic model $\mathbb{P}(\mathbf{s})$ of the concept interactions that is compatible with the empirical observations, but at the same time can describe these interactions in the most general way. This is achieved by applying the *principle of maximum entropy*, which states that the probability distribution that represents the empirical observations must maximize the entropy [8].

Although it may seem odd to apply a principle of statistical mechanics to a situation with no physical basis, it can be interpreted from an information theory point of view, expressing the epistemological modesty of our view of the system. In fact this principle is compelling because choosing a distribution that maximizes entropy suggests the state of least information. A state with lower entropy would imply that there is unknown information; therefore the states with maximum entropy express the situation of "most ignorance".

Explicitly, the *entropy* of the system is defined as

$$S \equiv -\sum_{\{s_k=0,1\}} \mathbb{P}(\mathbf{s}) \ln \mathbb{P}(\mathbf{s}) \tag{3}$$

where $\sum_{\{s_k=0,1\}} \equiv \sum_{\mathbf{s}}$ indicates the sum over all possible configurations of $\mathbf{s}$.

The distribution $\mathbb{P}(\mathbf{s})$ must satisfy the constraints of being with the experimental observations f_i and f_{ij}. This condition is expressed as

$$\sum_{\mathbf{s}} s_i \mathbb{P}(\mathbf{s}) = f_i \tag{4a}$$

$$\sum_{\mathbf{s}} s_i s_j \mathbb{P}(\mathbf{s}) = f_{ij} \tag{4b}$$

and $\mathbb{P}(\mathbf{s})$ is normalized

$$\sum_{\mathbf{s}} \mathbb{P}(\mathbf{s}) = 1 \tag{5}$$

The maximization of the entropy in eq. (3) under the constraints in eq. (4) and eq. (5) is achieved with the Lagrange multipliers, ie. solving

$$\frac{\delta}{\delta \mathbb{P}(\mathbf{s})} \left[-\sum_{\mathbf{s}} \mathbb{P}(\mathbf{s}) \ln \mathbb{P}(\mathbf{s}) + \lambda_0 \left(\sum_{\mathbf{s}} \mathbb{P}(\mathbf{s}) - 1 \right) + \sum_i \lambda_i \left(\sum_{\mathbf{s}} s_i \mathbb{P}(\mathbf{s}) - f_i \right) + \sum_{i,j} \mu_{ij} \left(\sum_{\mathbf{s}} s_i s_j \mathbb{P}(\mathbf{s}) - f_{ij} \right) \right] = 0$$

After a straightforward derivation, and after reabsorbing the constants we find

$$\ln \mathbb{P}(\mathbf{s}) = \lambda_0 + \sum_i \lambda_i s_i + \sum_{i,j} \mu_{ij} s_i s_j$$

In a notation that is more customary with Ising models, this becomes

$$\mathbb{P}(\mathbf{s}) = \frac{1}{\mathcal{Z}} \exp \mathcal{H}(\mathbf{s}) \tag{6}$$

where

$$\mathcal{H}(\mathbf{s}) = \sum_i h_i s_i + \sum_i \sum_{j>i} J_{ij} s_i s_j \tag{7}$$

and the *partition function* $\mathcal{Z}$ derives directly from eq. (5)

$$\mathcal{Z} = \sum_{\mathbf{s}} \exp \mathcal{H}(\mathbf{s}) \tag{8}$$

The *field* $\mathbf{h} = \{h_i\}$ and *coupling* $\mathbf{J} = \{J_{ij}\}$ are the parameters that determine the distribution function. As mentioned before, they must satisfy eq. (4). Just as a matter of notation, since $J_{ij} = J_{ji}$ (the pairwise interactions are not directional) we can avoid double-counting of the pairwise interactions by summing over $\sum_i \sum_{j>i}$.

A note on the Intractability of $\mathcal{Z}$ Considering eq. (8) we notice that even if the $\{\mathbf{h}, \mathbf{J}\}$ are known, the computational time required to evaluate $\mathcal{Z}$ becomes enormous very rapidly —its complexity is around $\mathcal{O}(n^2 2^n)$. For our case where $n \approx 10^4$ an exhaustive evaluation remains completely outside consideration from a practical point of view. We therefore will have to resort to methods that avoid this.

3 Method Development

3.1 Maximum Likelihood Estimation

Given a set $\{\mathbf{s}\}_{a=1}^{N}$ of observations we want to determine the parameters $\{\mathbf{h}, \mathbf{J}\}$. The standard statistical approach is to maximize the *likelihood*,

$$\mathcal{L}(\mathbf{h}, \mathbf{J}) = \prod_{a=1}^{N} \mathbb{P}(\mathbf{s}^{(a)}) \tag{9}$$

Indeed the likelihood of the set of parameters $\{\mathbf{h}, \mathbf{J}\}$ is equal to the probability of measuring the observed outcomes given that set of parameters. A maximum likelihood estimator coincides with the most probable Bayesian estimator, assuming the set of observations is independent, i.e. there is no correlation between the different $\mathbf{s}^{(a)}$. For practical reasons it is much more convenient to minimize the average *negative log-likelihood* [4]

$$\ell(\mathbf{h}, \mathbf{J}) \equiv -\frac{1}{N} \sum_{a=1}^{N} \ln \mathbb{P}(\mathbf{s}^{(a)}) \tag{10}$$

Therefore the estimators can be obtained by minimizing eq. (10), i.e.

$$\{\mathbf{h}^{\text{MLE}}, \mathbf{J}^{\text{MLE}}\} = \arg\min \ell(\mathbf{h}, \mathbf{J}) \tag{11}$$

Attempts to apply this in our case — plugging eq. (8) into eq. (10) and then deriving ℓ to find the minima — are illustrated in literature [4]. Unfortunately this yields a result that is explicitly dependent of the partition function. As illustrated previously, in order to have an estimator it is necessary to get rid of explicit dependencies in $\mathcal{Z}$.

3.2 Pseudolikelihood Approximation

A method to avoid full-space normalization is to use the *pseudolikelihood* as an approximation of the likelihood that has spatial dependencies [3][4]. Using the definition of *conditional probability*

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(A \bigcap B)}{\mathbb{P}(B)}$$

we can derive the conditional probability of having s_k given all the other spins. By denoting $\mathbf{s}_{\setminus k} = (s_1, \ldots, s_{k-1}, s_{k+1}, \ldots, s_n)$ we have

$$\mathbb{P}(s_k|\mathbf{s}_{\setminus k}) = \frac{\mathbb{P}(s_k, \mathbf{s}_{\setminus k})}{\mathbb{P}(\mathbf{s}_{\setminus k})} = \frac{\mathbb{P}(\mathbf{s})}{\mathbb{P}(s_k = 0, \mathbf{s}_{\setminus k}) + \mathbb{P}(s_k = 1, \mathbf{s}_{\setminus k})} \tag{12}$$

We can insert eq. (6) and cancel all terms that do not contain s_k

$$\mathbb{P}(s_k|\mathbf{s}_{\setminus k}) = \frac{\exp\left(h_k s_k + \sum_{j\neq k} J_{jk} s_j s_k\right)}{1 + \exp\left(h_k + \sum_{j\neq k} J_{jk} s_j\right)} \tag{13}$$

We notice at this point that the the dependency in $\mathcal{Z}$ has been eliminated. We use the denominator in eq. (13) as an approximation for the partition function for the site k. By summing over k and our set of articles we can define *negative pseudo-log-likelihood* in a form that is similar to eq. (10)

$$\hat{\ell}(\mathbf{h}, \mathbf{J}) = -\frac{1}{N} \sum_{a=1}^{N} \sum_{k} \ln \mathbb{P}(s_k^{(a)}|\mathbf{s}_{\setminus k}^{(a)}) \tag{14}$$

The estimators of $\mathbf{h}, \mathbf{J}$ are therefore obtained by minimizing $\hat{\ell}$

$$\{\mathbf{h}^{\mathrm{PLM}}, \mathbf{J}^{\mathrm{PLM}}\} = \arg\min \hat{\ell}(\mathbf{h}, \mathbf{J}) \tag{15}$$

It is important to note that in general the estimators $\{\mathbf{h}^{\mathrm{PLM}}, \mathbf{J}^{\mathrm{PLM}}\}$ which maximize the pseudolikelihood do *not* maximize the likelihood, i.e. satisfy eq. (10). It is however considered a valid approximation that often leads to meaningful results [3].

3.3 Regularization

What we did not mention about eq. (15) is that there may be multiple solutions, we prefer, however, the ones with most terms at zero, since they can describe the system with the minimal amount of interactions. Additionally it helps the convergence of the numerical optimization. As a matter of fact, $\hat{\ell}$ is convex — therefore a solution is always expected, but is extremely very flat around zero. As has previously been shown by Ekeberg [4], the standard way of doing this is introducing a *penalty function* R, so that eq. (15) becomes

$$\{\mathbf{h}^{\mathrm{PLM}}, \mathbf{J}^{\mathrm{PLM}}\} = \arg\min \left(\hat{\ell}(\mathbf{h}, \mathbf{J}) + R(\mathbf{h}, \mathbf{J})\right) \tag{16}$$

This procedure is called *regularization* [3], and in particular we choose to regularize using the L^2 norm, which is a convenient choice because it maintains the function to minimize differentiable. Using arbitrary positive constants λ_h and λ_J, our penalty function becomes

$$R(\mathbf{h}, \mathbf{J}) = \lambda_h \sum_i h_i^2 + \lambda_J \sum_i \sum_{j<i} J_{ij}^2 \tag{17}$$

We see that R pushes collectively the parameters towards zero. In fact to minimize R the Euclidean norms of $\mathbf{h}$ and $\mathbf{J}$ should separately be the smallest possible. The λ constants are chosen by striking a compromise between accurately fitting the data and maintaining small parameters.

3.4 Gradient of the Objective Function

It is now formulate the problem as a function optimization. Using eq. (13) and eq. (14), we can find an explicit formulation for $\hat{\ell}$

$$\hat{\ell}(\mathbf{h},\mathbf{J}) = -\frac{1}{N}\sum_{a=1}^{N}\sum_{k}\left[h_k s_k^{(a)} + \sum_{j\neq k} J_{jk} s_j^{(a)} s_k^{(a)} - \ln\left(1+\exp\left(h_k + \sum_{j\neq k} J_{jk} s_j^{(a)}\right)\right)\right]$$

We now define $F(\mathbf{h},\mathbf{J}) \equiv \hat{\ell}(\mathbf{h},\mathbf{J}) + R(\mathbf{h},\mathbf{J})$ as the *objective function*. Introducing the frequencies of eq. (2) we get

$$F = \sum_{k}\left[-f_k h_k - \sum_{j\neq k} -f_{jk} J_{jk} + \frac{1}{N}\sum_{a=1}^{N}\ln\left(1+\exp\left(h_k + \sum_{j\neq k} J_{jk} s_j^{(a)}\right)\right)\right] + R \quad (18)$$

Since F is well-behaved the minimum of eq. (16) can be found by deriving F along our parameters, i.e. when $\nabla_{\{\mathbf{h},\mathbf{J}\}} F(\mathbf{h},\mathbf{J}) = \mathbf{0}$. Using eq. (17) and eq. (18) we obtain

$$\partial_{h_i} F = -f_i + 2\lambda_h h_i + \frac{1}{N}\sum_{a=1}^{N}\frac{\exp\left(h_i + \sum_{k\neq i} J_{ik} s_k^{(a)}\right)}{1+\exp\left(h_i + \sum_{k\neq i} J_{ik} s_k^{(a)}\right)} = 0 \quad (19a)$$

$$\partial_{J_{ij}} F = -f_{ij} + 2\lambda_J J_{ij} + \frac{1}{N}\sum_{a=1}^{N} s_j^{(a)}\frac{\exp\left(h_i + \sum_{k\neq i} J_{ik} s_k^{(a)}\right)}{1+\exp\left(h_i + \sum_{k\neq i} J_{ik} s_k^{(a)}\right)} = 0 \quad (19b)$$

This set of equations is coupled, non-linear and not solvable analytically. We have however reduced the search for the field and coupling parameters to an optimization problem, that can be solved with numerical methods.

3.5 Notation with Triangular Matrices

To simplify the notation, let us introduce some new variables. We combine the field and couplings in a *lower triangular matrix*, so that the field is in on the diagonal elements. Similarly we generalize f_{ij} to contain the individual frequencies on the diagonal, and λ_{ij}. This is equivalent to

$$x_{ij} \equiv \begin{cases} J_{ij} & \text{if } i > j \\ h_i & \text{if } i = j \\ 0 & \text{if } i < j \end{cases}, \; f_{ij} \equiv \begin{cases} f_{ij} & \text{if } i > j \\ f_i & \text{if } i = j \\ 0 & \text{if } i < j \end{cases}, \; \lambda_{ij} \equiv \begin{cases} \lambda_h & \text{if } i = j \\ \lambda_J & \text{if } i \neq j \end{cases} \quad (20)$$

The objective function and its derivatives can be rewritten in a more compact form by defining

$$\xi_i^{(a)} \equiv \exp\left(x_{ii} + 2\sum_{k<i} x_{ik} s_k^{(a)}\right) \tag{21}$$

Finally we have

$$F = \sum_i \left(-\sum_{j\leqslant i} f_{ij} x_{ij} + \sum_{j\leqslant i} \lambda_{ij} x_{ij}^2 + \frac{1}{N}\sum_{a=1}^{N} \ln\left(1+\xi_i^{(a)}\right)\right) \tag{22a}$$

$$(\nabla F)_{ij} = -f_{ij} + 2\lambda_{ij} x_{ij} + \frac{1}{N}\sum_{a=1}^{N} \left\{s_j^{(a)}\right\}_{i\neq j} \frac{\xi_i^{(a)}}{1+\xi_i^{(a)}} \tag{22b}$$

where we write $\{s\}_{i\neq j} \equiv \delta_{ij} + (1-\delta_{ij})s$ using the Kronecker delta δ_{ij}.

4 Algorithm

4.1 Overview

A *steepest descent* algorithm finds the minimum of a function given its gradient, which is exactly what we have in this situation, see eq. (22). This technique translates x of an n-dimensional vector along the negative gradient, i.e. where the function descends the fastest. The library we chose[5] uses a improved method, called L-BFGS, that stands for *Limited-memory Broyden–Fletcher–Goldfarb–Shanno*. This algorithm keeps a history of a certain number of the previous updates to approximate the inverse Hessian. It can therefore build a quadratic model and get a better idea of what the function looks like, hopefully converging faster towards the minimum.

4.2 Description

We now present a more detailed description of the numerical optimization (see algorithm 1). For the initialization step the frequency matrix is initially calculated using eq. 2. The objective function F and its gradient ∇F are calculated in lines 4-21. This is achieved by iterating over the indexes i, a and calculating the exponential ξ that is found in eq. (22). Notice that for clarity we store the exponential and its argument in separate variables. This is however unnecessary and we can very well store this in the same variable, i.e. line 12 becomes $\xi \leftarrow \exp(\xi)$.

We can now keep the sum of the logarithmic terms for F in a temporary variable (see line 13) Σ and update the gradient terms $(\nabla F)_{ij}$ in lines 14-16. The sum Σ is

[5] ALGLIB, see www.alglib.net

Data: Spin matrix s

Result: Parameter matrix x

begin
  compute frequency matrix f
  $x \leftarrow \mathbf{0}$
  while $\neg$*haltingCondition* **do**
    $F \leftarrow 0$, $\nabla F \leftarrow \mathbf{0}$
    for $i = 1, \ldots, n$ **do**
      $\Sigma \leftarrow 0$ `// Stores the log sum`
      for $a = 1, \ldots, N$ **do**
        $\theta \leftarrow x_{ii}$ `// Stores the exp argument`
        for $j : s_{ij}^{(a)} = 1$ **do**
          $\theta \leftarrow \theta + x_{ij}$
        $\xi \leftarrow \exp(\theta)$ `// Different name for clarity`
        $\Sigma \leftarrow \Sigma + \log(1 + \xi)$
        $(\nabla F)_{ii} \leftarrow (\nabla F)_{ii} + \xi/(1 + \xi)$
        for $j : s_{ij}^{(a)} = 1$ **do**
          $(\nabla F)_{ij} \leftarrow (\nabla F)_{ij} + \xi/(1 + \xi)$
      $F \leftarrow F + \Sigma/N - f_{ii}x_{ii} + \lambda_h x_{ii}^2$
      $(\nabla F)_{ii} \leftarrow (\nabla F)_{ii}/N - f_{ii} + 2\lambda_h x_{ii}$
      for $j < i$ **do**
        $F \leftarrow F - 2f_{ij}x_{ij} + \lambda_J x_{ij}^2$
        $(\nabla F)_{ij} \leftarrow (\nabla F)_{ij}/N - f_{ij} + 2\lambda_J x_{ij}$
    `// Values of previous iterations are implicitly used for these steps`
    $x \leftarrow \mathrm{LBFGS}step(F, \nabla F, x)$
    update *haltingCondition*

Algorithm 1: Calculation of the objective function and its gradient for the optimization routine.

then added to F, along with the other terms polynomial in x. Similarly, the partial sum in $(\nabla F)_{ij}$ is divided by N and the rest of the terms are added (lines 17-21).

We then use F and ∇F to update x using the L-BFGS method (line 22). The operations on lines 22-23 actually use F, ∇F and x of previous iterations. We mentioned before that the optimization algorithm uses the previous values to tune the next translation step. But in order to compute the stopping conditions we must also store the values of the previous iteration. For example as a halting condition we choose

$$|F_{m+1} - F_m| \leqslant \epsilon_F \max(|F_{m+1}|, |F_m|, 1) \tag{23}$$

where F_m denotes the value of F at the m^{th} step of the routine. The constant ϵ_F is chosen by the user and determines how close we want to be to the actual minimum.

For *each* function and gradient evaluation, the time complexity is $\mathcal{O}(n^2 N)$. Since the dimensionality of our problem is quite high, the number of operations per iterations is quite important — probably at least 10^{12}. This makes the evaluation of the function and gradient the most time–consuming part compared to the optimization step of the algorithm.

To read how the algorithm was implemented, see appendix A.

5 Verification

5.1 Random Matrices

It is not simple to verify the validity of the algorithm, as the results are not something directly measurable. However, one ne way of verifying the algorithm is to use a *random matrix* for the spins, formed by independent random variables with a uniform distribution. By definition the system should have no correlations, because the spins are independent random variables. The corresponding Ising model will have a Gaussian distribution depending on the renormalization constant λ.[6] Introducing the penalty function eq. (17) in the negative log-likelihood eq. (10)

$$\ell'(x) = -\frac{1}{N} \sum_{a=1}^{N} \left(\ln \mathbb{P}(\mathbf{s}^{(a)}) - \lambda \|x\|^2 \right)$$

where $\|x\|^2 \equiv \mathrm{Tr}(x^\intercal x)$ denotes the Frobenius matrix norm. We return to the expression of the likelihood, which has the form

$$\mathcal{L}'(x) = \underbrace{\mathcal{L}(x)}_{\text{constant}} \, \mathrm{e}^{-N\lambda\|x\|^2} \tag{24}$$

[6] From now on we set $\lambda_h = \lambda_J \equiv \lambda$.

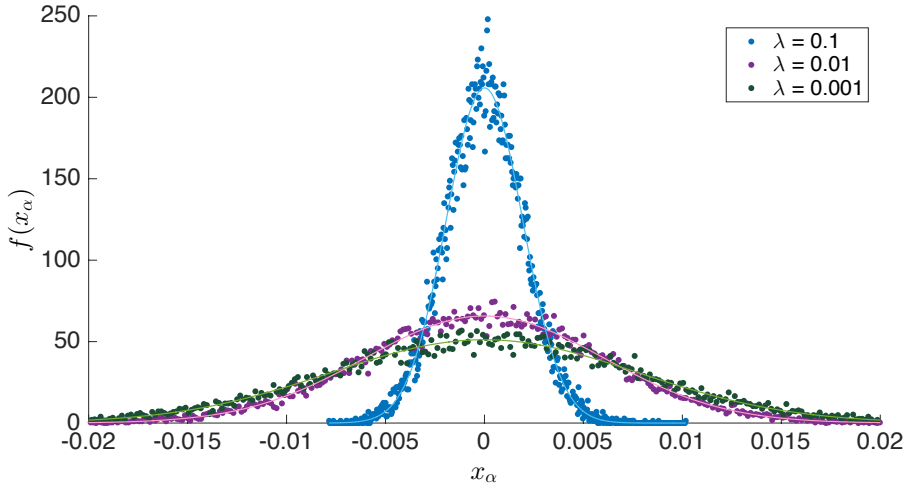

Figure 3: Comparison of the probability density function f for different values of λ for random matrix of dimensions $n = 200$, $N = 25000$. The continuous lines correspond to a Gaussian fit.

since the distribution of $\mathbf{s}^{(a)}$ is uniform and independent of the index a. From this form we deduce that the parameter distribution centered on zero, with standard deviation $\sigma = 1/\sqrt{2N\lambda}$. Therefore in Bayesian statistics the regularization term corresponds to a *prior*, that is the expected distribution that the parameters should have before considering the empirical data [3].

This prediction is indeed confirmed by what we qualitatively measure by binning the different parameters x_α[7] by their value and looking at their distribution (see fig. 3). The parameters are spread around zero with a variance that is inversely proportional to λ. From now on we shall choose to take $\lambda = 0.01$. This value is what is usually found in literature for inferences in protein sequences, but does not guarantee good results in this case. A full study of the prior parameter has not been attempted because of time constraints and the time complexity of our implementation.

5.2 Comparison with J Domain Results

The algorithm proposed was tested on results derived from *J domain* protein sequencings provided by Duccio Malinverni. These results were obtained using an asymmetric version of a pseudolikelihood inference on a Potts model, so the initial sequences had to binarized accordingly. Nonetheless, our results have a similar distribution and are strongly correlated (see fig. 4). The absolute value is taken as the results we compare with are all positive because they are under a different gauge. Finally, one can qualitatively compare the results by looking at the patterns in the

[7]The index α is used to indicate a generic index pair (i, j).

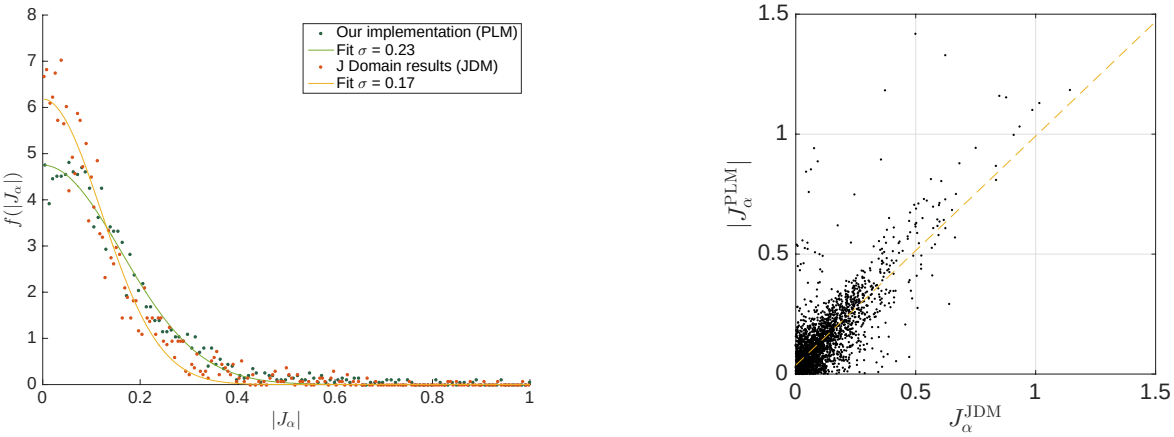

Figure 4: *Left*: comparison of the couplings distribution for the two methods. *Right*: correlations between the couplings of the two methods. The yellow line represents a linear fit.

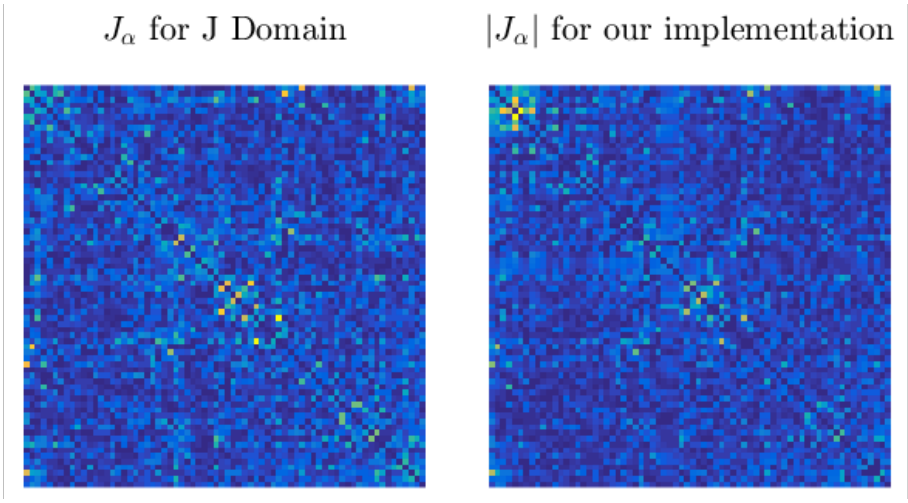

Figure 5: Visualization of the couplings matrix.

couplings, as shown in fig. 5. The signatures are convincingly similar.

6 Community Detection

6.1 Graph Construction

We have the couplings describing the interaction between all concepts. The goal now is to bunch them together to find sets of concepts that are tighly connected internally. There is a great abundance of clustering algorithms available [7], but it is difficult to define an appropriate metric (or pseudometric) that defines a distance between concepts. We resort to mapping the coupling matrix to an *undirected graph* or network, where each concept is identified with a vertex. Attemping to

incorporate the values of the couplings in edge weights is not a trivial problem. Couplings can assume all values in $\mathbb{R}$, while weights have to be stricty positive, with larger weights corresponding to weaker interactions, the opposite of couplings. We therefore decided to crudely binarize the couplings according to a *threshold value* $J_\perp$, such that all couplings under this value were set to zero (corresponding to breaking the graph edge corresponding to the coupling) and adding an edge for each coupling larger than $J_\perp$.

Once our graph has been defined we can attempt to find a *community structure*, by partitioning the network into *communities* that contain highly interconnected nodes, while nodes of different communities are only sparsely connected [9]. We focus on approaches based on the optimization of a quantity called the *modularity*, which roughly measures the density of edges inside communities to edges outside communities.

For a network partition into m different communities, we define the $m \times m$ symmetric matrix e such that e_{vw} is the fraction of all edges in the network that link vertices in community v to vertices in community w. The sum of the diagonal components of e indicates the fraction of edges that fall into the community partition. This must however weighed against a network with the same partition with random edges [9]. In this notation we define the modularity as

$$Q = \mathrm{Tr}(e) - |\mathbf{a}|^2 \tag{25}$$

where we write the sum of e over an index as $a_v = \sum_w e_{vw}$. Therefore the modularity measures the fraction of edges falling into the given communities compared with the expected value if the edges were distributed at random. If the number of community edges is equivalent to random connections, we obtain $Q = 0$. A value of Q approaching 1 indicates a partition highlighting a strong community structure. Exact maximization of the modularity is believed to be an NP-complete problem, but fortunately several heuristic algorithms have been proposed. We choose to use the *Louvain Method* [1], known for empirically running in linearithmic time.

6.2 Full Dataset

The distribution of the couplings obtained with the full dataset are presented in fig 6. The results are compared with the distribution for a random matrix that the same individual concept frequencies. This allows us to use the uncorrelated couplings as a *null model*. We then introduce the quantity t_γ that is the value of J corresponding to γ-quantile of the null model. Thus, by choosing t_γ correctly, we effectively select the part of couplings we consider significant.

Using the whole dataset does not yield a significant community structure. This may be caused by many factors, including a too stringent prior, or a large difference

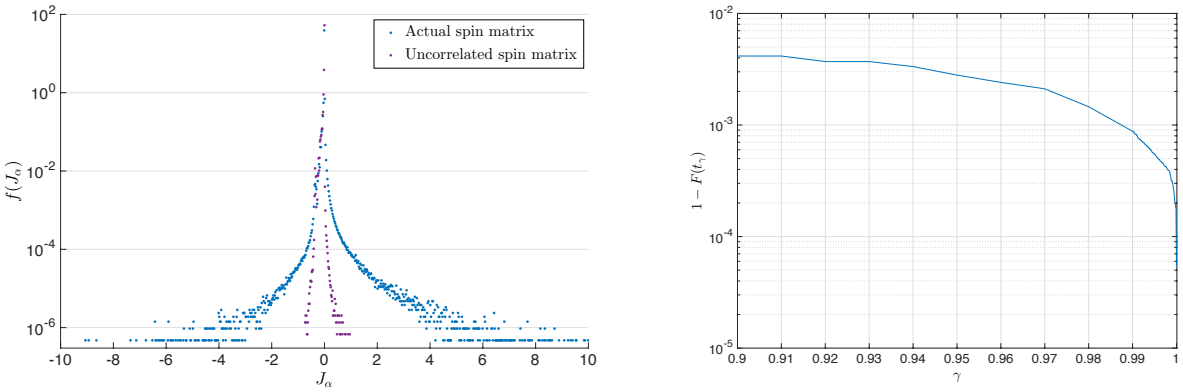

Figure 6: Distribution of couplings compared to the null model (*left*) and fraction of points over the quartile in function of t_γ (*right*).

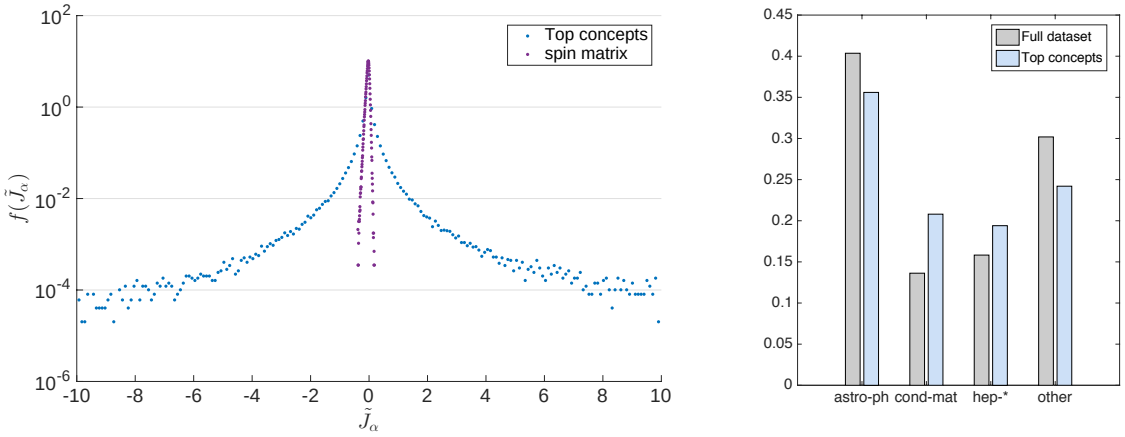

Figure 7: Distribution of couplings compared to the null model (*left*) and distribution of the concepts by category for the full dataset and the subset (*right*).

in frequencies. In fact the statistic may not be sufficient to reconstruct accurately the interaction of concepts that appear very rarely in the dataset.

6.3 Subset of Most Frequent Concepts

We restrict our focus to the first thousand most frequent concepts using a lower regularization factor of $\lambda = 0.001$. In order to compensate for strong frequency disparities, we introduce the *average product correction* is

$$\tilde{J}_{ij} = J_{ij} - \frac{|J|_{i\bullet}|J|_{\bullet j}}{|J|_{\bullet\bullet}} \tag{26}$$

where $|J|$ denotes the absolute value of each element and $\bullet$ indicates an average over that index. We then proceed in creating a network with the threshold value $J_\perp = 0.14 \approx t_1$.

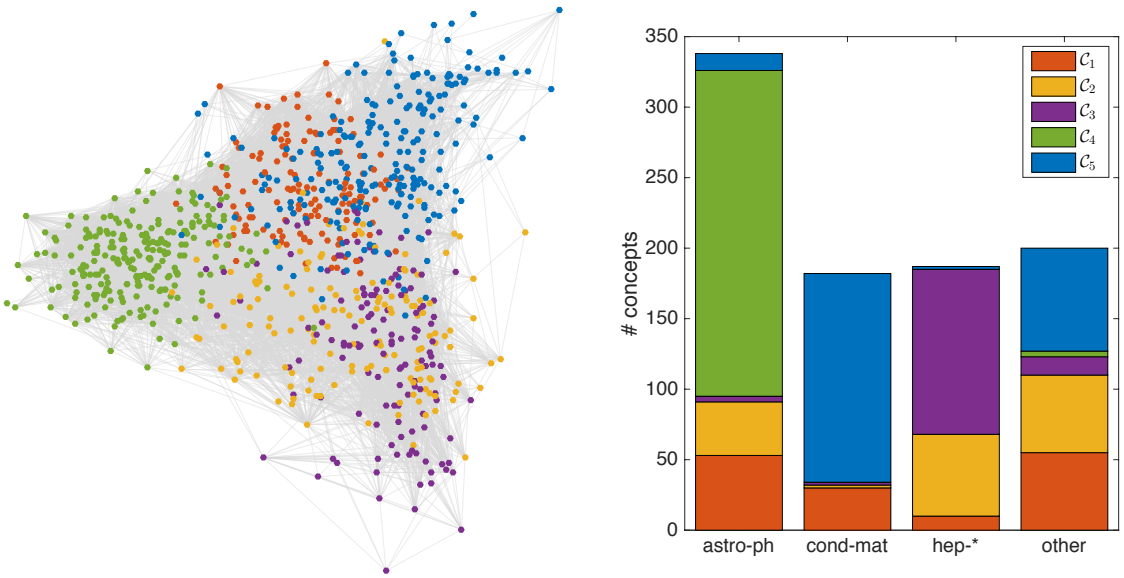

Figure 8: *Left*: visualization of the network, different communities are highlighted with different colors. *Right*: distribution within the different communities of the concepts associated with different categories.

One can take a look at some of the results in table 3 of appendix B. First for each concept we count the frequency in which it appears in articles associated with each arXiv category. Therefore we say that a concept is associated with the category in which it appears the most. In order to maintain an approximately even distribution (see table 1 and fig. 7), we combine smaller arXiv categories in the following groups: **hep-*** contains 'hep-ex', 'hep-lat', 'hep-ph', and 'hep-th'; **other** contains 'gr-qc', 'math-ph', 'nlin', 'nucl-ex', 'nucl-ph', 'physics' and 'quant-ph' . The categories **astro-ph** and **cond-mat** are kept separately since they contain a large portion of the concepts.

We notice that community $\mathcal{C}_1$ contains the most frequent concepts in the whole dataset (see table 2). These concepts are very generic, indeed they seem to be distributed in each of our categories (see fig. 8). The best results seem to be obtained for $\mathcal{C}_3$, $\mathcal{C}_4$ and $\mathcal{C}_5$. Both from a qualitative analysis and looking at their distribution along the categories they clearly contain keywords related to particle physics, astronomy and condensed matter respectively. A qualitative analysis of $\mathcal{C}_2$ suggest it is semantically related to cosmology. The distribution in fig 8 shows that its concepts belong mostly to **hep-***, **astro-ph** and **other**. This is quite possible since the 'gr-qc' category is contained in **other**, and we expect this field is tightly related to high energy physics and astronomy.

$\mathcal{C}$	size	$\overline{f}$	σ_f	main categories
1	148	0.14	0.11	other/astro-ph
2	153	0.024	0.019	hep-*/other
3	136	0.032	0.027	hep-*
4	235	0.026	0.022	astro-ph
5	235	0.026	0.016	cond-mat

Table 2: Summary of the community properties: size, average frequency $\overline{f}$, standard deviation σ_f and the main communities detected.

7 Conclusion

In this project we have proposed a method to build an effective ontology of concepts out of a corpus of scientific papers and extract subsets of similar concepts. By applying the principle of maximum entropy we construct an implicit Ising model, for which we want to determine the parameters. This is achieved using pseudolikelihood maximization to reduce the problem to a function optimization. An algorithm is then proposed to solve the problem numerically.

As a proof of concepts, we focused on the first thousand most frequent concepts in our dataset, and proceeded to extract a community structure out of the interaction network so as to identify groups of semantically similar concepts. We then compared our results with the categorizations within arXiv; this has allowed to validate our method by recognizing that the concepts of each community to belong to a specific category.

The current implementation is not fully mature. The biggest challenges lie in the interpretation of the couplings and on the construction of the taxonomy. For practical purposes we built an undirected graph by using a cutoff value. We imagine that more refined techniques which retain more coupling intensity information can be developed. Finally, a more careful study of the regularization term should be done to establish which regularization norm and which parameter values are optimal. Moreover, these details can be tailored for specific applications.

Automated ontology learning is a topic that has been enjoying growing interest in recent years, both in academia and in commercial applications. The extraction of a hierarchy or of an underlying structure will allow a new range of algorithms and services based on semantics rather than syntax [2].

The source code for this project is available at
https://github.com/giacomogiudice/concepts

References

[1] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[2] Dario Bonino, Fulvio Corno, Laura Farinetti, and Alessio Bosca. Ontology driven semantic search. *WSEAS Transaction on Information Science and Application*, 1(6):1597–1605, 2004.

[3] Yadolah Dodge. *The Oxford dictionary of statistical terms*. Oxford University Press, 2006.

[4] Magnus Ekeberg. Detecting contacts in protein folds by solving the inverse Potts problem – a pseudolikelihood approach. Master's thesis, Royal Institute of Technology, 2012.

[5] Lucio Floretta. *Topics in Community Detection*. PhD thesis, École Polytechnique Fédérale de Lausanne, 2014.

[6] Nicola Guarino. *Formal ontology in information systems: Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy*, volume 46. IOS press, 1998.

[7] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[8] Edwin T Jaynes. Information theory and statistical mechanics. *Physical Review*, 106(4):620, 1957.

[9] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113, 2004.

A Technical Considerations

Implementation As one can see from the source code, our implementation of algorithm 1 was developed in **C++**, using an object–oriented approach to subdivide the tasks. A simplified view of the main classes is presented in fig. 9. We can follow the arrows and view the flow of information within the class methods. The data is initially treated by an instance of the `IO` class, is used to start up the `Optimizer` and initialize correctly the `Function`. There is then a cycle between `Optimizer::run()` and `Function::evaluate()`, that perform the optimization steps and the function evaluations respectively. Finally, the data is saved on the disk.

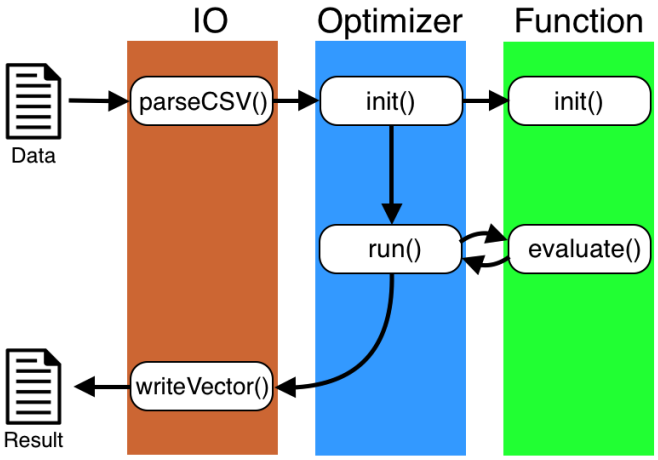

Figure 9: Flow of calls to the class methods.

Data Structures Regarding data structures, the triangular matrices in eq. (22) are stored as an array, rather than a matrix so as to not waste memory storing the upper diagonal, which is zero. Therefore one must change indices when accessing memory: this is achieved with the change of indices

$$(i, j) \mapsto \alpha = \frac{i(i+1)}{2} + j, \; i \geqslant j$$

so that $x_\alpha = x_{ij}$.

It is worth mentioning the fact that the spin matrix s, in our representation of the spins is mostly composed of zeros, i.e. it is a *sparse matrix*. We can therefore keep the non-zero indices of each row in an array of arrays, thus saving a considerable amount of memory. It also has the advantage of knowing directly which terms contribute in the sum in eq. (22b), saving additional computational time.

B List of Concepts

Community 1	Community 2	Community 3	Community 4	Community 5
Energy	Equation of motion	Cross section	Star	Crystal
Measurement	Curvature	Fermion	Absorptivity	Electric field
Field	Commutator	Bosonization	Luminosity	Lasers
Potential	Black hole	Lagrangian	Galaxy	Dipole
Mass	Quantization	Experimental data	Binary star	Plasma
Particles	Differential equations	Quark	X-ray	Hybridization
Temperature	Scalar field	Proton	Redshift	Chemical potential
Probability	Dark matter	Renormalization	Spectroscopy	Resist
Units	General relativity	Standard Model	Ellipticity	Hilbert space
Vector	Equations of state	Phase space	Accretion	Elasticity
Electron	Gauge invariance	Hadronization	Ionization	Liquids
Frequency	Horizon	Center of mass	Gaussian distribution	Green's function
Periodate	Spinor	Baryons	Full width at half maximum	Eigenvector
Symmetry	Coordinate system	P-symmetry	Star formation	Energy level
Universe	Cosmological constant	Leptons	Turbulence	Density matrix
Velocity	Quantum field theory	Quantum number	Sun	Density of states
Objective	Tension	Neutron	Milky Way	Insulators
Scattering	Gauge theory	Gluon	Signal to noise ratio	Free energy
Statistics	Einstein field equations	Strong interactions	Photometry	Ferromagnetism
Materials	The early Universe	Neutrino	Observatories	Carbonate
Charge	Duality	Nucleon	Companion	Semiconductor
Spin	String theory	Mesons	Extinction	Superconductor
Momentum	Permutation	Bound state	Light curve	Critical point
Formate	Cosmic microwave background	Supersymmetry	Likelihood	Wavefunction
Optics	Covariant derivative	Charged particle	Supernova	Quasiparticle
Resolution	Causality	Helicity	Stellar mass	Electromagnetic field
Fluctuation	Subgroup	Vacuum expectation value	Field of view	Plane wave
Force	Dark energy	Branching ratio	Compilers	Dielectric
Geometry	Gravitational constant	Higgs boson	Recombination	Vibration
Intensity	Planck mission	Weak interaction	Point source	Qubit
Photon	Gravitational fields	Pion	Cosmic ray	Partition function
Picture	Gauge transformation	Electric charge	Radial velocity	Dispersion relation
Hamiltonian	Cosmological model	Unitarity	A giants	Phonon
Event	Gravitational wave	Parton	Interstellar medium	Potential energy
Atom	Evolution equation	Form factor	Sloan Digital Sky Survey	Electronic density
Magnetics	Scalar curvature	Particle physics	Optical depth	Viscosity
Order of magnitude	Planck scale	Renormalization group	Neutron star	Unit cell
Polarization	Effective action	Spectrometers	Quadrature	Effective mass
Magnetic field	Quantum gravity	Muon	Primary star in a binary system	Pauli matrices
Degree of freedom	Minkowski space	Gauge symmetry	Mounting	Antiferromagnet
Action	Non-Gaussianity	Isotope	Massive stars	Magnetic moment
Resonance	Isomorphism	Gauge bosons	Planet	Lithium
Scalar	Geodesic	Effective potential	Near-infrared	Eigenfunction
Wavelength	Newton	Invariant mass	Mass ratio	Spin-orbit interaction
Gas	Analytic continuation	Beyond the Standard Model	Spectral energy distribution	Numerical methods
Transformations	Vacuum state	Strangeness	Effective temperature	Harmonic oscillator
Gravitation	Scale invariance	Isospin	X-ray spectrum	Brillouin zone
Algorithms	Dynamical systems	Effective field theory	Statistical significance	Graphene
Orbit	Supergravity	Perturbative expansion	Stellar populations	Diffraction
Homogenization	Cold dark matter	Gauge coupling coupling	Spectrographs	Critical temperature

Table 3: First 50 most frequent concepts for each community.