Exact Inference on Hierarchical Clustering in Particle Physics and Cancer Genomics

Craig S. Greenberg[*,1,2], **Sebastian Macaluso**[*,3], **Nicholas Monath**[1], **Patrick Flaherty**[1], **Kyle Cranmer**[3], **Andrew McGregor**[1], **Andrew McCallum**[1]
[1] University of Massachusetts Amherst
[2]National Institute of Standards and Technology
[3]New York University
csgreenberg@cs.umass.edu
seb.macaluso@nyu.edu

Abstract

Hierarchical clustering is a fundamental task often used to discover meaningful structures in data. We present dynamic-programming algorithms for *exact* inference, i.e we can compute the partition function and maximum likelihood hierarchical clustering. Our algorithms scale in time and space proportional to the powerset of N elements which makes it significantly faster than considering every possible hierarchy ($(2N-3)!!$). We show applications in particle physics and cancer genomics, where our algorithms outperform greedy and beam search baselines.

1 Introduction

Hierarchical clustering is often used to discover meaningful structures, such as phylogenetic trees of organisms Kraskov et al. [2005], taxonomies of concepts Cimiano and Staab [2005], subtypes of cancer Sørlie et al. [2001], and jets in particle physics Cacciari et al. [2008]. Among the reasons that hierarchical clustering has been found to be broadly useful is that it forms a natural data representation of data generated by a Markov tree, i.e., a tree-shaped model where the state variables are dependent only on their parent or children.

We define a hierarchical clustering as a recursive splitting of a dataset of elements, $X = \{x_i\}_{i=1}^N$ into subsets until reaching singletons, e.g. leaves of a binary tree. This can equivalently be viewed as starting with the set of singletons and repeatedly taking the union of sets until reaching the entire dataset. We show a schematic representation in Figure 1, where we identify each x_i with a leaf of the tree and the latent state as $\mathtt{H}$.

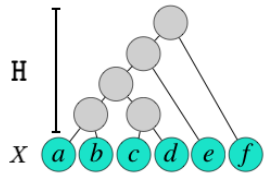

Figure 1: Schematic representation of a hierarchical clustering. $\mathtt{H}$ denotes the latent state and X the dataset of leaves.

We consider an energy-based probabilistic model for hierarchical clustering. We provide a general (and flexible) definition of this model and implementations in particle physics and cancer genomics. Our model is based on measuring the compatibility of each pair of sibling nodes, described by a potential function $\psi : 2^X \times 2^X \to \mathbb{R}^+$. We also denote the potential function for a hierarchical clustering $\mathtt{H}$ and dataset X as $\phi(X|\mathtt{H})$. Then, the probability of $\mathtt{H}$ for the dataset X, $P(\mathtt{H}|X)$, is equal to the unnormalized potential of $\mathtt{H}$ normalized by the partition function, $Z(X)$:

*The first two authors contributed equally.

Third Workshop on Machine Learning and the Physical Sciences (NeurIPS 2020), Vancouver, Canada.

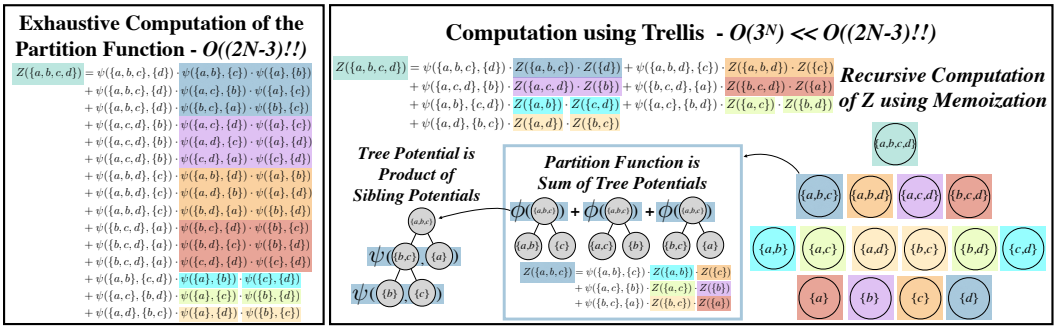

Figure 2: **Computing the partition function for the dataset** $\{a, b, c, d\}$. Left: exhaustive computation, consisting of the summation of $(2 \cdot 4 - 3)!! = 15$ energy equations. Right: computation using the trellis. The sum for the partition function is over $2^{4-1} - 1 = 7$ equations, each making use of a memoized Z value. Colors indicate corresponding computations over siblings in the trellis.

$$P(\mathtt{H}|X) = \frac{\phi(X|\mathtt{H})}{Z(X)} \quad \text{with} \quad \phi(X|\mathtt{H}) = \prod_{X_L, X_R \in \text{siblings}(\mathtt{H})} \psi(X_L, X_R) \tag{1}$$

where the partition function $Z(X)$ is given by:

$$Z(X) = \sum_{\mathtt{H} \in \mathcal{H}(X)} \phi(X|\mathtt{H}). \tag{2}$$

and $\mathcal{H}(X)$ gives all binary hierarchical clusterings of the elements X. We refer to this as an energy-based model since often it is the case that $\psi(\cdot, \cdot)$ is defined by the unnormalized Gibbs distribution, as $\psi(X_L, X_R) = \exp(-\beta E(X_L, X_R))$, where β is the inverse temperature and $E(\cdot, \cdot)$ is the energy.

Next, we define MAP hierarchy as the maximum likelihood hierarchical clustering given a dataset X. Exactly performing inference on the MAP hierarchy and finding the partition function by enumerating all hierarchical clusterings over N elements is exceptionally difficult because the number of hierarchies grows extremely rapidly, namely $(2N - 3)!!$ (see Callan [2009], Dale and Moon [1993] for more details and proof). To overcome the computational burden, we introduce a cluster trellis data structure for hierarchical clustering (see Greenberg et al. [2018] for the equivalent algorithm over flat clustering). Our algorithms compute these quantities in the $\mathcal{O}(3^N)$ time, without having to iterate over each possible hierarchy. While still exponential, this is feasible in regimes where enumerating all possible trees would be infeasible, and is to our knowledge the fastest exact MAP/partition function result, making practical exact inference for datasets on the order of 20 points (~3×10^9 operations vs ~10^{22} trees) or fewer. Our proposed approach is inspired by classic uses of dynamic programming in inference, such as the Sum-Product Algorithm and Viterbi. To the best of our knowledge these algorithms and related ones (e.g. belief propagation, message passing, etc) cannot be directly applied to the aforementioned probabilistic model for hierarchical clustering because that would require to express the distribution over hierarchies as a graphical model.

Contributions of this paper. We achieve *exact*, not approximate, solutions to compute the **partition function** $Z(X)$ and **MAP inference**, i.e. find the maximum likelihood tree structure.

2 Hierarchical Cluster Trellis Algorithm

Computing the Partition Function. Given a dataset of elements, $X = \{x_i\}_{i=1}^N$, the partition function, $Z(X)$, for the set of hierarchical clusterings over X, $\mathcal{H}(X)$, is given by Equation 2. The partition function for every node in the trellis is computed in order (in a bottom-up approach), memoizing the partial value at each node. A visualization comparing the trellis algorithm to the brute force method for a dataset of four elements is shown in Figure 2. To implement the trellis, we need to re-write Equation 2 in the corresponding recursive way as follows,

Proposition 1. For any $x \in X$, the hierarchical partition function can be written recursively, as $Z(X) = \sum_{\mathtt{H} \in \mathcal{H}(X)} \phi(\mathtt{H}) = \sum_{X_i \in X_x} \psi(X_i, X \setminus X_i) \cdot Z(X_i) \cdot Z(X \setminus X_i)$ where X_x is the set of all clusters containing the element x (omitting X), i.e,. $X_x = \{X_j : X_j \in 2^X \setminus X \wedge x \in X_j\}$.

Computing the Maximum Likelihood Hierarchical Clustering. The MAP hierarchy for dataset X, $\mathtt{H}^\star(X)$, is $\mathtt{H}^\star(X) = \text{argmax}_{\mathtt{H} \in \mathcal{H}(X)} P(\mathtt{H}|X) = \text{argmax}_{\mathtt{H} \in \mathcal{H}(X)} \phi(\mathtt{H})$. As in the partition function,

we can use a recursive memoized technique. Each node will store a value for the MAP hierarchy, denoted $\phi(\mathtt{H}^\star(X))$ and a backpointer $\Xi(\mathtt{H}^\star(X))$. Specifically,

Proposition 2. For any $x \in X$, let $X_x = \{X_j : X_j \in 2^X \setminus X \wedge x \in X_j\}$, then $\phi(\mathtt{H}^\star(X)) = \max_{X_i \in X_x} \psi(X_i, X \setminus X_i) \cdot \phi(\mathtt{H}^\star(X_i)) \cdot \phi(\mathtt{H}^\star(X \setminus X_i))$.

3 Experiments

3.1 Jet Physics

Background Detectors at the Large Hadron Collider (LHC) at CERN measure the energy (and momentum) of particles produced from proton-proton collisions. The final-state particles that hit the detector are stable and originated by a *showering process* where an initial (unstable) particle goes through successive binary splittings until reaching the final-state ones, represented by the leaves in a binary tree. Typically, a collimated set of final-state particles are clustered together as a *jet*. These leaves are observed while the latent showering process, described by quantum chromodynamics (QCD), is not. As a result, there are several latent trees that correspond to a set of leaves. This representation, first suggested in Louppe et al. [2019], connects jets physics with natural language processing (NLP) and biology.

Currently, generative models in full physics simulations for the showering process that produces a set of leaves do not admit a tractable density (they are implicit models). A main problem in data analyses in collider physics deals with estimating this latent showering process. Thus, an open area of research aims to unify generation and inference, which typically requires extracting additional information from the simulator; e.g estimate the clustering history of a set of leaves. At the moment, clustering algorithms implemented in data analyses are greedy and based on heuristics.

At present, it is very hard to access the joint likelihood in state-of-the-art parton shower generators in full physics simulations. Also, typical implementations of parton showers involve sampling procedures that destroy the analytic control of the joint likelihood. Thus, to aid in machine learning research for jet physics, a python package for a toy generative model of a parton shower, called Ginkgo, was introduced in Cranmer et al. [2019b]. Ginkgo has a tractable joint likelihood, and is as simple and easy to describe as possible but at the same time captures essential ingredients of parton shower generators in full physics simulations. Within the analogy between jets and NLP, Ginkgo can be thought of as ground-truth parse trees with a known language model.

Probabilistic Model The potential of a hierarchy is identified with the product of the likelihoods of all the $1 \to 2$ splittings of a parent cluster into two child clusters in the binary tree. Each cluster, X, corresponds to a particle with an energy-momentum vector $x = (E \in \mathbb{R}^+, \vec{p} \in \mathbb{R}^3)$ and squared mass $t(x) = E^2 - |\vec{p}|^2$. A parent's energy-momentum vector is obtained from adding its children, i.e., $x_P = x_L + x_R$. We study a toy model for jet physics, where for each pair of parent and left (right) child cluster with masses $\sqrt{t_P}$ and $\sqrt{t_L}$ ($\sqrt{t_R}$) respectively, the likelihood function is,

$$\psi(X_L, X_R) = f(t(x_L)|t_P, \lambda) \cdot f(t(x_R)|t_P, \lambda) \quad \text{with} \quad f(t|t_P, \lambda) = \frac{1}{1 - e^{-\lambda}} \frac{\lambda}{t_P} e^{-\lambda \frac{t}{t_P}} \tag{3}$$

where the first term in $f(t|t_P, \lambda)$ is a normalization factor associated to the constraint that $t < t_P$.

Data and Methods The ground truth hierarchical clusterings of our dataset are generated with the toy generative model for jets Ginkgo, see Cranmer et al. [2019a] for more details. This is a simulation model for cascades of particle physics decays in jet physics. This model implements a recursive algorithm to generate a binary tree, where each node is represented by a four dimensional energy-momentum vector and the leaves are the jet constituents. We compare the trellis results with greedy and beam search baselines. Greedy simply chooses the pairing of nodes that locally maximizes the likelihood at each step, whereas beam search maximizes the likelihood of multiple steps before choosing the latent path. The current implementation only takes into account one more step ahead, with a beam size given by $N/2(N-1)$, with N the number of jet constituents to cluster. Also, when two or more clusterings had an identical likelihood value, only one of them was kept in the beam, to avoid counting multiple times the different orderings of the same clustering (see Boyles and Welling [2012] for details about the different orderings of the internal nodes of the tree). This approach significantly improved the performance of beam search.

Results We show results for the implementation of the trellis algorithm on a jet physics dataset of 5000 Ginkgo Cranmer et al. [2019b] jets with a number of leaves between 5 and 10, and we refer to it

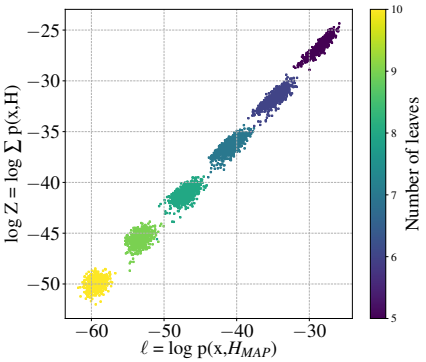

Figure 3: **Jet Physics.** Scatter plot of the partition function Z vs. the trellis MAP ℓ for the Ginkgo510 dataset. There appears to be a correlation between Z and the MAP.

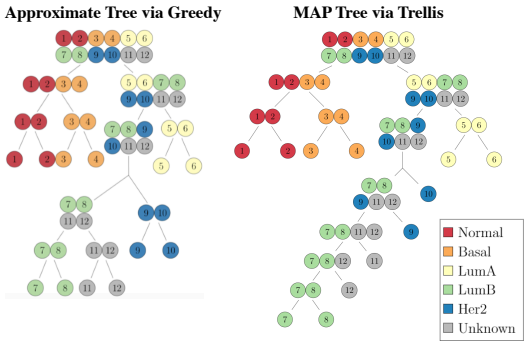

Figure 4: **Cancer Genomics**. Comparison of trees from greedy (left) and exact MAP clustering using the trellis (right) on the subsampled `pam50` data set. The colors indicate subtypes of breast cancer (grey if unknown).

as Ginkgo510. We start by comparing in Table 1 the mean difference among the MAP values for the hierarchies obtained with the trellis, beam search and greedy algorithms. We see that the likelihood of the trees increase from greedy to beam search to the trellis one, as expected. Next, in Figure 3 we show a plot of the partition function versus MAP for each set of leaves. It is interesting to note that there seems to be a correlation between Z and the Trellis MAP hierarchy. We want to emphasize that the trellis enables the calculation of the partition function.

3.2 Cancer Genomics

Background In cancer genomics, we want to model subtypes of cancer, which can help determine prognosis and treatment plans. Hierarchical clustering is a common clustering approach for gene expression data [Sørlie et al., 2001]. However, standard hierarchical clustering uses a greedy agglomerative or divisive heuristic to build a tree. It is not uncommon to have a need for clustering a small number of samples in cancer genomics studies. An analysis of data available from https://clinicaltrials.gov shows that the median sample size for 7,412 completed phase I clinical trials involving cancer is only 30.

	Beam Search	Greedy
Trellis	0.4 ± 0.5	1.5 ± 1.1
Beam Search		1.1 ± 1.1

Table 1: Mean and standard deviation for the difference in log likelihood for the MAP tree found by algorithms indicated by the row and column headings on the Ginkgo510 dataset.

Probabilistic Model In this case we are given a dataset of vectors indicating the level of gene expressions which are endowed with pairwise affinities that are both positive and negative. We define the energy of a pair of sibling nodes in the tree to be the sum of the positive edges from elements in one child to elements in the other one, minus the negative edges between two elements in the same child.

$$\psi(X_i, X_j) = \exp(-\beta E(X_i, X_j)) \tag{4}$$

$$E(X_i, X_j) = \sum_{x_i, x_j \in X_i \times X_j} w_{ij} \mathbb{I}[w_{ij} > 0] - \sum_{\substack{x_i, x_j \in X_i \times X_i, \\ x_i \neq x_j}} w_{ij} \mathbb{I}[w_{ij} < 0] - \sum_{\substack{x_i, x_j \in X_j \times X_j, \\ x_i \neq x_j}} w_{ij} \mathbb{I}[w_{ij} < 0] \tag{5}$$

where w_{ij} is the affinity between x_i and x_j. This energy is the correlation clustering objective Bansal et al. [2004].

Data and Methods We compare a greedy agglomerative clustering to our exact MAP tree using the Prediction Analysis of Microarray 50 (`pam50`) gene expression data set. The `pam50` dataset ($n = 232$, $d = 50$) is available from the UNC MicroArray Database [University of North Carolina, 2020]. It has intrinsic subtype annotations for 139 of the 232 samples. Missing data values (2.65%) were filled in with zeros. We drew a stratified sample of the total data set with two samples from each known intrinsic subtype and two samples from the unknown group.

Results Figure 4 displays the greedy hierarchical clustering tree and the MAP tree with transformed weights for the twelve samples selected from the pam50 dataset. The main difference between these trees is in the split of the subtree including LumB, HER2, and unknown samples. The greedy method splits HER2 from LumB and unknown, while the MAP hierarchy shows a different topology for this

subtree. For the MAP solution, we note that the subtree rooted at $\{7, 8, 9, 10, 11, 12\}$ is consistent. All of the correlation coefficients among this cluster are positive, so the optimal action is to split off the item with the smallest (positive) correlation coefficient.

4 Conclusion

This paper describes a data structure and dynamic-programming algorithm to exactly compute the partition function and MAP hierarchy over all hierarchical clusterings given a dataset. Our method improves upon the computation cost of brute-force methods from $(2N-3)!!$ to sub-quadratic in the substantially smaller powerset of N. We demonstrate that our methods outperform current baselines on jet physics and cancer genomics datasets.

5 Broader Impact

Hierarchical clustering is a fundamental task that is used in a wide range of domains including phylogenetics, physics, and information sciences. Therefore advances in hierarchical clustering have the potential for broad impact. Our work is particularly relevant in situations where one would like to consider many such clusterings weighted by a domain-motivated energy function. Providing a computationally efficient means to consider all such clusterings enables the the treatment of uncertainty and other probabilistic concepts, which can aid in the responsible use of such clusterings for down-stream tasks.

Unlike approximate inference methods, our exact method depends only on the energy based model and not the inference method. This provides the practitioner the ability to analyze and better understand the energy-based model independent of approximate inference considerations. It also carries with it the responsibility of the practitioner to design energy-based models that account for potential impacts of the particular application.

In particular, the implementation of our algorithm in the context of jet physics could improve analyses of data from the Large Hadron Collider at CERN. The algorithm can remove computational bottle necks in various approaches to unify the generative models and inference tasks encountered there. It also has the potential to speed up state-of-the-art simulators used in particle physics. However, there are remaining challenges to implement our algorithm on the more complex models used in those physics simulators.

In the genomics case, hierarchical clustering is a ubiquitous tool in the analysis of gene expression data and used to better understand diseases such as cancer and neurodegenerative disorders. However, algorithms for finding a hierarchical clustering are greedy and may not find the optimal tree; thus data items may be misclustered.

In medical genetic association studies, such as the one present in Section 4, the data items in hierarchical clustering are samples from real people who have a life-threatening disease. In modern precision medicine, targeted therapeutics are allocated based on a connection between a sample's genetic profile and a targeted therapeutic. Therefore, correctly and exactly clustering samples means that an individual is allocated to the correct group and can mean the difference between a person receiving a life-saving treatment and not.

It is important to acknowledge the role of an algorithm such as hierarchical clustering in the allocation of treatments to individuals on the basis of a genetic profile. We should think about how clustering can help advanced medical treatments to be allocated fairly and how the results of the algorithm can drive the development of targeted therapeutics. Even if they are accruing benefits in terms of improved lifespan and quality of life to individuals, we should ask ourselves if the allocations of resources is increasing inequality in society.

Acknowledgements

Kyle Cranmer and Sebastian Macaluso are supported by the National Science Foundation under the awards ACI-1450310 and OAC-1836650 and by the Moore-Sloan data science environment at NYU. Patrick Flaherty is supported in part by NSF HDR TRIPODS award 1934846. Andrew McCallum and Nicholas Monath are supported in part by the Center for Data Science and the Center for Intelligent Information Retrieval, and in part by the National Science Foundation under Grant No. NSF-1763618.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

References

N. Bansal, A. Blum, and S. Chawla. Correlation clustering. *Machine learning*, 56(1-3):89–113, 2004.

L. Boyles and M. Welling. The time-marginalized coalescent prior for hierarchical clustering. In *Advances in Neural Information Processing Systems*, pages 2969–2977, 2012.

M. Cacciari, G. P. Salam, and G. Soyez. The anti-k_t jet clustering algorithm. *JHEP*, 04:063, 2008. doi: 10.1088/1126-6708/2008/04/063.

D. Callan. A combinatorial survey of identities for the double factorial, 2009.

P. Cimiano and S. Staab. Learning concept hierarchies from text with a guided agglomerative clustering algorithm. In *Proceedings of the ICML 2005 Workshop on Learning and Extending Lexical Ontologies with Machine Learning Methods*, 2005.

K. Cranmer, S. Macaluso, and D. Pappadopulo. Toy Generative Model for Jets, 2019a. Toy Generative Model for Jets.

K. Cranmer, S. Macaluso, and D. Pappadopulo. Toy Generative Model for Jets Package, 2019b. https://github.com/SebastianMacaluso/ToyJetsShower.

E. Dale and J. Moon. The permuted analogues of three Catalan sets, 1993.

C. Greenberg, N. Monath, A. Kobren, P. Flaherty, A. McGregor, and A. McCallum. Compact representation of uncertainty in clustering. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 8630–8640. Curran Associates, Inc., 2018. http://papers.nips.cc/paper/8081-compact-representation-of-uncertainty-in-clustering.pdf.

A. Kraskov, H. Stögbauer, R. G. Andrzejak, and P. Grassberger. Hierarchical clustering using mutual information. *EPL (Europhysics Letters)*, 70(2):278, 2005.

G. Louppe, K. Cho, C. Becot, and K. Cranmer. QCD-Aware Recursive Neural Networks for Jet Physics. *JHEP*, 01:057, 2019. doi: 10.1007/JHEP01(2019)057.

T. Sørlie, C. M. Perou, R. Tibshirani, T. Aas, S. Geisler, H. Johnsen, T. Hastie, M. B. Eisen, M. Van De Rijn, S. S. Jeffrey, et al. Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. *Proceedings of the National Academy of Sciences*, 98(19): 10869–10874, 2001.

University of North Carolina. UNC microarray database, 2020. https://genome.unc.edu/.