Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence

# Formalizing Hierarchical Clustering as Integer Linear Programming

**Sean Gilpin**
Department of Computer Science
University of California, Davis
sagilpin@ucdavis.edu

**Siegfried Nijssen**
Department of Computer Science
Katholieke Universiteit Leuven
siegried.nijssen@cs.kuleuven.be

**Ian Davidson**
Department of Computer Science
University of California, Davis
davidson@cs.ucdavis.edu

### Abstract

Hierarchical clustering is typically implemented as a greedy heuristic algorithm with no explicit objective function. In this work we formalize hierarchical clustering as an integer linear programming (ILP) problem with a natural objective function and the dendrogram properties enforced as linear constraints. Though exact solvers exists for ILP we show that a simple randomized algorithm and a linear programming (LP) relaxation can be used to provide approximate solutions faster. Formalizing hierarchical clustering also has the benefit that relaxing the constraints can produce novel problem variations such as overlapping clusterings. Our experiments show that our formulation is capable of outperforming standard agglomerative clustering algorithms in a variety of settings, including traditional hierarchical clustering as well as learning overlapping clusterings.

## Introduction

A recent survey (Kettenring 2008) comparing non-hierarchical and hierarchical clustering algorithms showed that in published scientific articles, hierarchical algorithms are used far more than non-hierarchical clustering. Applications of hierarchical clustering typically can be divided into those that build large trees so that, for instance, a user can navigate a large collection of documents, and those that build trees to represent a scientific process, such as phylogenetic trees (evolutionary trees). We can further differentiate these works by noting that the data collected for the first type of application is easy to collect and hence voluminous, whilst the later application typically takes as much as a year to collect and are typically small.

The focus of this work is the latter category of applications involving a small number of instances taking a long time to collect and which must be thoroughly analyzed. In that way spending hours for an algorithm to run is not uncalled for since the data has taken far longer to collect and a precise answer is worth the wait. Colloquially, no one will complain if a dendrogram places documents in a good but non-optimal ordering but will if species are shown to evolve in the wrong order.

However, hierarchical clustering algorithms remain relatively under-studied with most algorithms being relatively straight-forward greedy algorithms implemented in procedural language. For example, in the above mentioned survey two thirds of the implementations mentioned were simple agglomerative algorithms that start with each instance in a cluster by itself and then the closest two clusters are merged at each and every level. Even more advanced methods published in the database literature such as CLARANS and DBSCAN (Tan, Steinbach, and Kumar 2005) still use this base approach but have more complex distance measures or methods to build the tree iteratively. Whereas non-hierarchical clustering algorithms have moved to elegant linear algebra formulations, hierarchical clustering has not.

In this work we provide to the best of our knowledge the first formalization of agglomerative clustering as a mathematical programming formulation, namely, an ILP problem (for a general introduction to the topic see Nemhauser and Wolsey and for an example of an application of ILP to the clustering data mining problem see Mueller and Kramer). Formulating the problem as an ILP has the benefit that high quality solvers (free and commercial) such as CPLEX can be utilized to find solutions. Formulating the problem as an ILP has a number of other benefits that we now briefly discuss and provide details of later.

*Explicit Global Objective Function Optimizing.* As mentioned most existing work greedily determines the best join at each and every level of the hierarchy. At no time is it possible to reset or revisit an earlier join. Though this is adequate when a "near enough" dendrogram is required, such as building a tree to organize song lists, finding the global optima is most important when the data represents a physical phenomenon. This is discussed in the Section *Hierarchical Clustering as an ILP,* and we show it produces better quantitative results for language evolution in Table 1 and for hierarchical organization of fMRI scans in Table 2 .

*Novel Problem Variations with Relaxing Constraints.* A side benefit of formalizing hierarchy learning is that the properties of a legal dendrogram are explicitly modeled as constraints to the optimization. We will show how novel problem variations can arise if some constraints are relaxed. In particular we show that relaxing the transitivity property allows for overlapping hierarchical clustering, that is, an instance can appear multiple times in the hierarchy and is akin to overlapping clustering. To our knowledge the problem of building dendrograms when an object appears multiple times is novel. This topic is explored in the Section *Relaxed Problem Settings* and in Figure 5 we show empirically that our method is capable of discovering overlapping hierarchies.

*Approximation Schemes* A large literature exists on general methods to create approximate methods for ILPs. We show that by exploiting simple randomized algorithms we can create a factor two approximation schemes. We also explore using LP relaxations and randomized rounding. Figure 6 shows the empirical results of how well our LP relaxation and randomized rounding scheme compares to the optimal solution.

## Hierarchical Clustering as an ILP

In this section, we discuss how to formulate hierarchical clustering as an ILP problem. Solving hierarchical clustering using ILP allows us to find globally optimal solutions to hierarchical clustering problems, in contrast to traditional hierarchical clustering algorithms which are greedy and do not find global optimums. The two challenges to an ILP formulation are: i) ensuring that the resultant dendrogram is legal; and ii) encoding a useful objective function.

### Enforcing a Legal Dendrogram

In this section, we describe how a function (referred to as `merge`), representable using $O(n^2)$ integer variables (where $n$ is the number of instances), can represent any dendrogram. The variables represent what level a pair of instances are joined at and the constraints described in this section enforce that these variables obey this intended meaning. The `merge` function is described in Definition 1.

**Definition 1.** ***`merge` function***

$$\begin{aligned} \texttt{merge} : (\mathsf{Instances} \times \mathsf{Instances}) &\rightarrow \mathsf{Levels} \\ (a, b) &\mapsto \textit{first level } a, b \textit{ are in same cluster} \end{aligned}$$

In this work we will learn this function. For a particular instance pair $a, b$ the intended meaning of $\texttt{merge}(a, b) = \ell$ is that instances $a, b$ are in the same cluster at level $\ell$ of the corresponding dendrogram, but not at level $\ell - 1$. Therefore the domain of the `merge` function is all pairs of instances and the range is the integers between zero (bottom) and the maximum hierarchy level $L$ (top).

The fact that any dendrogram from standard hierarchical clustering algorithms can be represented using this scheme is clear, but it is not the case that any such `merge` function represents a legal dendrogram. Specifically this encoding does not force each level of the dendrogram to be a valid clustering. Therefore, when learning dendrograms using this encoding we must enforce the following partition properties: reflexivity, symmetry, and transitivity (Definitions 2,3,4). Later we shall see how to enforce these requirements as the linear inequalities in an ILP.

**Definition 2.** ***Reflexivity*** *An instance is always in the same cluster as itself.*

$$\forall a\, [merge(a, a) = 0]$$

**Definition 3.** ***Symmetry*** *If instance $a$ is in the same cluster as instance $b$ then instance $b$ is also in the same cluster as instance $a$.*

$$\forall a, b\, [merge(a, b) = merge(b, a)]$$

**Definition 4.** ***Transitivity*** *If at a certain level instance $a$ is in the same cluster as instance $b$ and $b$ is in the same cluster as $c$, then $a$ is also in the same cluster as $c$ at the same level.*

$$\forall a, b, c\, [max\, (merge(a, b), merge(b, c)) \geq merge(a, c)]$$

Additionaly, constraining the `merge` function to represent a sequence of clusterings is not enough (for hierarchical clustering) because not every sequence of clusterings forms a dendrogram. In particular once points are merged they can never be unmerged, or put another way, clusterings at level $l$ can only be formed by merging clusters at level $l - 1$ (Definition 5).

**Definition 5.** ***Hierarchical Property*** *For all clusters $C_i$ at level $\ell$, there exists no superset of $C_i$ at level $\ell - 1$.*

### Hierarchical Clustering Objective

The objective for hierarchical clustering is traditionally specified as a greedy process where clusters that are closest together are merged at each level. These objectives are called linkage functions (e.g. single (Sibson 1973), complete (Sørensen 1948), UPGMA (Sokal and Michener 1958),WPGMA (Sneath and Sokal 1973)) and are used by agglomerative algorithms to determine which clusters are closest and should therefore be merged next. The intuition behind this process can be interpreted to mean that points that are close together should be merged together before points that are far apart.

To formalize that intuition we created an objective function that favors hierarchies where this ordering of merges is enforced. For example if $D(a, c)$ (distance between $a$ and $c$ ) is larger than $D(a, b)$ then the points $a$ and $c$ should be merged after the points $a$ and $b$ are merged ($\texttt{merge}(a, b) < \texttt{merge}(a, c)$). This objective is shown formally in Equation 1 which uses the definitions of variables $O$ and $w$ from Equations 2 and 3 respectively. As mentioned before, we learn the merge function and $\mathcal{M}$ is a specific instantiation of this function. Intuitively $O_{abc}$ has value one if the instances $a$ and $b$ are merged before $a$ and $c$ are merged and $w_{abc}$ is positive if $a$ and $b$ are closer together than $a$ and $c$.

$$f(\mathcal{M}) = \sum_{a,b,c \in Instances} w_{abc} O_{abc} \tag{1}$$

where:

$$O_{abc} = \begin{cases} 1 & : \mathcal{M}(a, b) < \mathcal{M}(a, c) \\ 0 & : \text{otherwise} \end{cases} \tag{2}$$

$$w_{abc} = D(a, c) - D(a, b) \tag{3}$$

*Interpretations of the Objective Function.* Our aim will be to maximize this objective function under constraints to enforce a legal dendrogram. We can therefore view the objective function as summing over all possible triangles and penalizing the objective with a negative value if the points that form the shorter side are not joined first. Our objective function can also be viewed as learning an ultra-metric. Recall that with an ultra-metric the triangle inequality is replaced with the stronger requirement that $D(a,b) \leq \mathtt{max}(D(a,c), D(b,c))$. Effectively our method learns the ultra-metric that is most similar to the original distance function. There is previous work showing the relationship between ultrametrics and hierarchical clustering that this work can be seen as building upon (Carlsson and Mémoli 2010).

### ILP Transformation

Now that we have presented a global objective function for hierarchical clustering we will address how to find globally optimum legal hierarchies by translating the problem into an integer linear program (ILP). Figure 1 shows at a high level the optimization problem we are solving, but it is not an ILP.

$$\underset{\mathcal{M},\mathbf{O}}{\text{arg max}} \sum_{\{a,b,c \in Instances\}} w_{abc} * O_{abc}$$

subject to :

$$(1)\ \mathcal{M} \text{ is a merge function}$$

$$(2)\ O_{abc} = \begin{cases} 1 & : \mathcal{M}(a,b) < \mathcal{M}(a,c) \\ 0 & : \text{otherwise} \end{cases}$$

Figure 1: High level optimization problem for hierarchical clustering. Constraint 1 specifies that we are optimizing over valid dendrograms. Constraint 2 ensures that indicator variables used in objective have proper meaning.

Figure 2 shows the ILP equivalent of the problem in Figure 1. The reason they are equivalent is explained briefly here, but essentially amounts to converting the definition of dendrogram into linear constraints using standard integer programming modelling tricks. To ensure that $\mathcal{M}$ is a valid hierarchy, we need to add constraints to enforce reflexivity, symmetry, transitivity, and the hierarchical properties (Definitions 2 - 5). Recall that $\mathcal{M}$ can be represented as square matrix of $n \times n$ variables indicating at what level a pair of points are joined. Then reflexivity and symmetry are both easily enforced by removing redundant variables from the matrix, in particular removing all diagonal variables for reflexivity, and all lower triangle variables for symmetry. Transitivity can be turned into a set of linear constraints by noticing that the inequality $max(\mathcal{M}(a,b), \mathcal{M}(b,c)) \geq \mathcal{M}(a,c)$ is logically equivalent to:

$$(\mathcal{M}(a,b) \geq \mathcal{M}(a,c)) \vee (\mathcal{M}(b,c) \geq \mathcal{M}(a,c)) \quad (4)$$

In the final ILP (Figure 2) there is a variable $Z$ introduced for each clause/inequality from Equation 4 (i.e. $Z_{ab\geq ac}$ and $Z_{bc\geq ac}$) and there are three constraints added to enforce that the disjunction of the inequalities is satisfied. Constraints are also added to ensure that the $\mathbf{O}$ variables reflect the ordering of merges in $\mathcal{M}$.

## Relaxed Problem Settings

A benefit of formalizing hierarchy building is that relaxing requirements can give rise to a variety of novel problem set-

$$\underset{\mathcal{M},\mathbf{O},\mathbf{Z}}{\text{arg max}} \sum_{a,b,c \in Instances} w_{abc} * O_{abc}$$

subject to :

$$\begin{aligned}
&\mathbf{O}, \mathbf{Z}, \mathcal{M} \text{ are integers.} \\
&0 \leq \mathbf{O} \leq 1,\ 0 \leq \mathbf{Z} \leq 1 \\
&1 \leq \mathcal{M} \leq L \\
&-L \leq \mathcal{M}(a,c) - \mathcal{M}(a,b) - (L+1)O_{abc} \leq 0 \\
&-L \leq \mathcal{M}(a,b) - \mathcal{M}(a,c) - (L+1)Z_{ab\geq ac} + 1 \leq 0 \\
&-L \leq \mathcal{M}(b,c) - \mathcal{M}(a,c) - (L+1)Z_{bc\geq ac} + 1 \leq 0 \\
&Z_{ab\geq ac} + Z_{bc\geq ac} \geq 1
\end{aligned}$$

Figure 2: ILP formulation of hierarchical clustering with global objective. The third constraint specifies the number of levels the dendrogram can have by setting the parameter $L$. The fourth constraint forces the $O$ objective variables to have the meaning specified in Equation 2. Constraints 5-7 specify that $\mathcal{M}$ must be a valid dendrogram.

tings. Here we consider the effect of relaxing the transitivity constraint of our ILP formulation of hierarchical clustering. We consider this relaxation for two reasons: 1) it allows a form of hierarchical clustering where instances can appear in multiple clusters at the same level (i.e. overlapping clustering); and 2) it allows every possible variable assignment to have a valid interpretation so that we can create approximation algorithms by relaxing the problem into an LP and using randomized rounding schemes. In the following sections, we will discuss how the relaxation of transitivity leads to hierarchies with overlapping clusters and we will discuss how this relaxation affects the ILP formulation. We will also discuss approximation algorithms that can be used in this setting.

### Overlapping Hierarchies

When the transitivity property of the `merge` function is relaxed, the clusterings corresponding to each level of the dendrogram will no longer necessarily be set partitions. Clustering over non-transitive graphs is common in the community detection literature (Aggarwal 2011) where overlapping clusters are learned by finding all maximal cliques in the graph (or weaker generalizations of maximal clique). We use this same approach in our work, finding maximal cliques for each level in the hierarchy to create a sequence of overlapping clusterings. Such a hierarchy is not consistent with the notion of dendrogram from traditional hierarchical clustering algorithms but is still consistent with the general definition of hierarchy.

### Overlapping Clustering Objective

Although we relax the transitivity requirement of the `merge` function in this setting, intuitively graphs in which transitivity is better satisfied will lead to simpler overlapping clusterings. We therefore added transitivity in the objective function (rather than having it as a constraint) as shown in Equation 5 which introduces a new variable $T_{abc}$ (Equation 6) whose value reflects whether the instance triple $a, b, c$ obeys transitivity. We also introduce a new weight $w'_{abc}$ which specifies how important transitivity will be to the objective.

$$f(\mathcal{M}) = \sum_{a,b,c \in Instances} [w_{abc} O_{abc} + w'_{abc} T_{abc}] \quad (5)$$

$$T_{abc} = \begin{cases} 1 & : max\,(\mathcal{M}(a,b), \mathcal{M}(b,c)) \geq \mathcal{M}(a,c) \\ 0 & : \text{otherwise} \end{cases} \quad (6)$$

The full ILP, with the new objective presented in Equation 5 and relaxed transitivity, is presented in Figure 3.

$$\begin{aligned}
&\underset{\mathcal{M},\mathbf{O},\mathbf{Z}}{\text{arg max}} \sum_{a,b,c \in Instances} [w_{abc} * O_{abc} + w'_{abc} * T_{abc}] \\
&\text{subject to :} \\
&\quad \mathbf{T}, \mathbf{O}, \mathbf{Z}, \mathcal{M} \text{ are integers.} \\
&\quad 0 \leq \mathbf{T} \leq 1,\ 0 \leq \mathbf{O} \leq 1,\ 0 \leq \mathbf{Z} \leq 1 \\
&\quad 1 \leq \mathcal{M} \leq L \\
&\quad -L \leq \mathcal{M}(a,c) - \mathcal{M}(a,b) - (L+1) O_{abc} \leq 0 \\
&\quad -L \leq \mathcal{M}(a,b) - \mathcal{M}(a,c) - (L+1) Z_{ab \geq ac} + 1 \leq 0 \\
&\quad -L \leq \mathcal{M}(b,c) - \mathcal{M}(a,c) - (L+1) Z_{bc \geq ac} + 1 \leq 0 \\
&\quad Z_{ab \geq ac} + Z_{bc \geq ac} \geq T_{abc}
\end{aligned}$$

Figure 3: ILP formulation with relaxation on transitivity property of `merge` function.

## Polynomial Time Approximation Algorithms

In this section we consider some theoretical results for approximations of the ILP formulation presented in Figure 3. This formulation allows transitivity to be violated, and has an interpretation that allows any variable assignment to be translated into a valid hierarchy (with overlapping clusters).

### Factor Two Approximation

**Theorem 1.** *Let $\mathcal{M}_0$ be created by independently sampling each value from the uniform distribution $\{1 \dots L\}$, and $\mathcal{M}^*$ be the optimal solution to ILP in Figure 3. Then $E[f(\mathcal{M}_0)] \geq \frac{L-1}{2L} f(\mathcal{M}^*)$.*

*Proof.*

$$\begin{aligned}
E[f(\mathcal{M}_0)] &= E\left[\sum_{abc} (w_{abc} O_{abc} + w'_{abc} T_{abc})\right] \\
&= \sum_{abc} w_{abc} E\left[O_{abc}\right] + \sum_{abc} w'_{abc} E\left[T_{abc}\right] \\
&= \sum_{abc} w_{abc} \frac{L-1}{2L} + \sum_{abc} w'_{abc} \frac{4L^3 + 3L^2 - L}{6L^3} \\
&= \frac{L-1}{2L} \sum_{abc} w_{abc} + \frac{4L^3 + 3L^2 - L}{6L^3} \sum_{abc} w'_{abc} \\
&\geq \frac{L-1}{2L} f(\mathcal{M}^*) \approx \frac{1}{2} f(\mathcal{M}^*)
\end{aligned}$$

□

Note that in the proof we use the following results for the values of $E\left[O_{abc}\right]$ and $E\left[T_{abc}\right]$:

$$E\left[O_{abc}\right] = p(\mathcal{M}_0(a,b) > \mathcal{M}_0(a,c)) = \frac{L-1}{2L}$$

$$\begin{aligned}
E\left[T_{abc}\right] &= p(\mathcal{M}_0(a,b) \geq \mathcal{M}_0(a,c) \vee \mathcal{M}_0(b,c) \geq \mathcal{M}_0(a,c)) \\
&= 1 - \frac{2L^3 - 3L^2 + L}{6L^3} = \frac{4L^3 + 3L^2 - L}{6L^3}
\end{aligned}$$

The bound $E[f(\mathcal{M}_0)] \geq \frac{L-1}{2L} f(\mathcal{M}^*)$ is a constant given $L$ (parameter specifying number of levels in hierarchy), and will generally be close to one half since the number of levels $L$ is typically much greater than 2.

### LP Relaxation

The problem in Figure 3 can be solved as a linear program by relaxing the integer constraints, but the resulting solution, $\mathcal{M}_f^*$ will not necessarily have all integer values. Given such a solution, we can independently round each value up with probability $\mathcal{M}_f^*(a,b) - \lfloor \mathcal{M}_f^*(a,b) \rfloor$ and down otherwise. The expectation of the objective value for the LP relaxation can be calculated and in the experimental section we calculate both optimal integer solutions and the expectation for this simple rounding scheme. Figure 6 shows that their difference is usally very small. If $\mathcal{M}_0$ is the solution created by rounding $\mathcal{M}_f^*$, then the expectation can be calculated as:

$$\begin{aligned}
E[f(\mathcal{M}_0)] &= E\left[\sum_{abc} (w_{abc} O_{abc} + w'_{abc} T_{abc})\right] \\
&= \sum_{abc} w_{abc} E\left[O_{abc}\right] + \sum_{abc} w'_{abc} E\left[T_{abc}\right]
\end{aligned}$$

The expectation for each variable ($T$ and $O$) breaks down into a piecewise function. The expectation for $O_{abc}$ is listed below. This variable relies on the ordering of $\mathcal{M}_f^*(a,b)$ and $\mathcal{M}_f^*(a,c)$. When those variables are far apart (e.g. distance of 2 or greater) then rounding them will not change the value of $O_{abc}$. When they are close the value of $O_{abc}$ will depend on how they are rounded, which breaks down into two cases depending on whether $\mathcal{M}_f^*(a,b)$ and $\mathcal{M}_f^*(a,c)$ are in the same integer boundary.

$$E\left[O_{abc}\right] = \begin{cases} 1 & : \lfloor \mathcal{M}_f^*(a,b) \rfloor > \lceil \mathcal{M}_f^*(a,c) \rceil \\ 1 - p_1 & : \lfloor \mathcal{M}_f^*(a,b) \rfloor = \lceil \mathcal{M}_f^*(a,c) \rceil \\ p_2 & : \lfloor \mathcal{M}_f^*(a,b) \rfloor = \lfloor \mathcal{M}_f^*(a,c) \rfloor \\ 0 & : \lceil \mathcal{M}_f^*(a,b) \rceil < \lfloor \mathcal{M}_f^*(a,c) \rfloor \end{cases}$$

$$p_1 = \left(\lceil \mathcal{M}_f^*(a,b) \rceil - \mathcal{M}_f^*(a,b)\right)\left(\mathcal{M}_f^*(a,c) - \lfloor \mathcal{M}_f^*(a,c) \rfloor\right)$$

$$p_2 = \left(\mathcal{M}_f^*(a,b) - \lfloor \mathcal{M}_f^*(a,b) \rfloor\right)\left(\lceil \mathcal{M}_f^*(a,c) \rceil - \mathcal{M}_f^*(a,c)\right)$$

The expectation for $T_{abc}$ can be calculated using similar reasoning but it is not listed here because it breaks down into a piecewise function with more cases and is very large.

## Experiments

In our experimental section we answer the following questions:

- *Does our global optimization formulation of hierarchical clustering provide better results?* Our results in Tables 1 and 2 show that our method outperforms standard hierarchical clustering for language evolution and fMRI data sets where finding the best hierarchy matters. Figure 4 shows how our method does a better job of dealing with noise in our artificial data sets.
- *Given that the relaxation of constraints can be used to find hierarchies with overlapping clusterings, in practice can it find naturally overlapping clusters in datasets?* Figure 5 shows the results of our method on an artificial overlapping dataset and Table 3 for a real data set.
- *Our randomized approximation algorithms provides theoretical bounds for the difference between the expected and optimal solutions. In practice how do the random solutions compare to optimal solutions?* Figure 6 is a plot of the actual differences between random and optimal solutions.

**Artificial Data Set** We created two artificial hierarchical clustering data sets with a known ground truth hierarchy, so that we could use it to precisely answer some important questions about our new hierarchical clustering formulation. The first hierarchy we created had 80 instances, 4 levels, was a balanced tree, and did not have overlapping clusterings. We used this data set to evaluate our basic ILP formulation which enforced transitivity (Figure 2) and compared the results with standard hierarchical clustering algorithms (single, complete, UPGMA, WPGMA). The distances between two points $a$ and $b$ was measured as a function of the level of their first common ancestor. We increased the challenge of the problem by increasingly adding uniform error to the distance matrix. Figure 4 shows the results and demonstrates our method's ability to outperform all standard agglomerative clustering algorithms for standard hierarchical clustering (no overlapping clusters), we believe this is because we solve a global optimization that is less affected by noise than algorithms that use greedy search heuristics. The F1 score was calculated by creating the set of all clusters generated within the learned and true dendrograms, and matching the learned clusters with the best true clusters. The experiment was repeated ten times for each error factor level and the results are shown in Figure 4.

The second artificial data set had 80 instances, 4 levels and was balanced, but each of the clusters on the highest non-root level shared 50% of their instances with another cluster (overlap). We evaluated our overlapping clustering formulation against standard hierarchical clustering and presented the results in Figure 5. We used the same overlapping hierarchy to test how well our expected linear programming results compared to the optimal results found using an ILP solver. Those results are presented in Figure 6 and show that in practice using an LP solution along with a very simple rounding scheme leads to results very close to the optimal ILP objective.

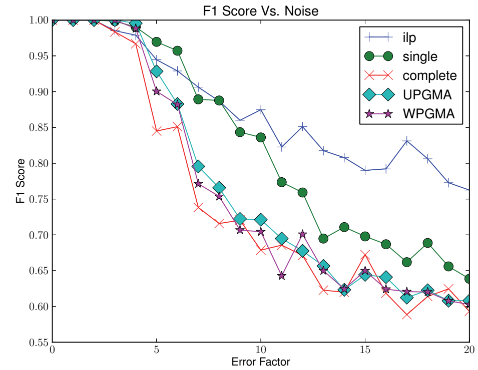


Figure 4: Effect of noise on prediction performance. F1 score versus noise in distance function. Comparison of basic ILP formulation on artificial data against a variety of competitors.

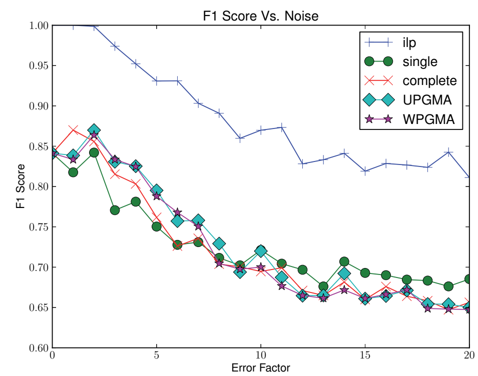


Figure 5: Ability to find overlapping clusterings. F1 score versus noise in distance function. Comparison of overlapping ILP formulation on artificial data against a variety of competitors.

Table 1: Performance of our basic ILP formulation on the languages evolution data set evaluated using H-Correlation (Bade and Benz 2009) compared to the known ground truth dendrogram. The higher the value the better.

| Algorithm | H-Correlation |
|---|---|
| ILP | 0.1889 |
| Single | 0.1597 |
| Complete | 0.1854 |
| UPGMA | 0.1674 |
| WPGMA | 0.1803 |

**Languages Data Set** The language data set contains phonological, morphological, lexical character usage of twenty- four historic languages (Nakhleh, Ringe, and Warnow 2002) (`http://www.cs.rice.edu/~nakhleh/CPHL/`). We chose this data set because it allowed us to test our method's ability to find a ground truth hierarchy from high dimensional data. These languages are known to have evolved from each other with scientists agreeing upon the ground truth. In such problem settings, finding the exact order of evolution of the entities (in this case

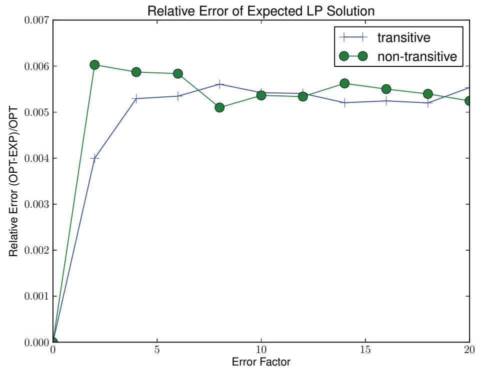


Figure 6: Measuring optimality of randomized solutions. Performance versus noise in distance function. Comparison of expected LP results versus optimal ILP results using two transitivity parameterizations (transitive $\rightarrow$ very large $w'$, non-transitive $\rightarrow w' = 0$).

languages) is important and even small improvements are considered significant. Global optimization provide a closer dendrogram to the ground truth as shown in Table 1.

**fMRI Data Set** An important problem in cognitive research is determining how brain behavior contributes to mental disorders such as dementia. We were given access to a set of MRI brain scans of patients who had also been given a series of cognitive tests that allowed these patients to be organized into a natural hierarchy based on their mental health status. Once again finding the best hierarchy is important since it can be used to determine those most at risk. From these scans we randomly sampled 50 patients and created a distance matrix from their MRI scans. We then evaluated our hierarchical clustering method along side with standard hierarchical clustering methods, comparing each to the ground truth hierarchy using H-Correlation. We repeated this experiment 10 times and reported the results in table 2.

Table 2: Performance of our basic ILP formulation on fMRI data to recreate an hierarchy that matches the ordering of a persons fMRI scans based on their cognitive scores. Our method has higher average H-correlation with greater than $95\%$ confidence.

| Algorithm | H-Correlation | Standard Deviation |
|---|---|---|
| ILP | 0.3305 | 0.0264 |
| Single | 0.3014 | 0.0173 |
| Complete | 0.3149 | 0.0306 |
| UPGMA | 0.3157 | 0.0313 |
| WPGMA | 0.3167 | 0.0332 |

**Movie Lens** The Move Lens data set (Herlocker et al. 1999) is a set of user ratings of movies. This data set is of interest because each movie also has a set of associated genres, and these genres typically have a very high overlap and thus are not easily formed into a traditional hierarchical clustering. We created clusterings from all the genres and cross-genres (e.g. romantic comedy, action comedy) and tested our method's ability to find these overlapping clusterings as compared to standard agglomerative clustering methods. The results are presented in Table 3 and show that our method had better average results (sampled 80 movies 10 times) as well as a smaller variance than all of the agglomerative clustering algorithms.

Table 3: Performance of overlapping formulation on real world data, the Movie Lens data set with an ideal hierarchy described by genre. Our method has higher average F1 score with greater than $95\%$ confidence.

| Algorithm | F1 Score | Standard Deviation |
|---|---|---|
| LP | 0.595 | 0.0192 |
| Single | 0.547 | 0.0242 |
| Complete | 0.569 | 0.0261 |
| UPGMA | 0.548 | 0.0245 |
| WPGMA | 0.563 | 0.0246 |

## Conclusion

Hierarchical clustering is an important method of analysis. Most existing work focuses on heuristics to scale these methods to huge data sets which is a valid research direction if data needs to be quickly organized into any meaningful structure. However, some (often scientific) data sets can take years to collect and although they are much smaller, they require more precise analysis at the expense of time. In particular, a good solution is not good enough and a better solution can yield significant insights. In this work we explored two such data sets: language evolution and fMRI data. In the former the evolution of one language from another is discovered and in the later the organization of patients according to their fMRI scans indicates the patients most at risk. Here the previously mentioned heuristic methods perform not as well, which is significant since even small improvements are worthwhile.

We present to the best of our knowledge the first formulation of hierarchical clustering as an integer linear programming problem with an explicit objective function that is globally optimized. Our formulation has several benefits: 1) It can find accurate hierarchies because it finds the global optimum. We found this to be particularly important when the distance matrix contains noise (see Figure 5); 2) By formalizing the dendrogram creation we can relax certain requirements to produce novel problem settings. We explored one such setting, overlapping clustering, where an instance can appear multiple times in the hierarchy. To our knowledge the problem of overlapping hierarchical clustering has not been addressed before.

## Acknowledgments

The authors gratefully acknowledge support of this research via ONR grant N00014-11-1-0108, NSF Grant NSF IIS-0801528 and a Google Research Gift. The fMRI data provided in this grant was created by NIH grants P30 AG010129 and K01 AG 030514

## References


Aggarwal, C. C., ed. 2011. *Social Network Data Analytics*. Springer.

Bade, K., and Benz, D. 2009. Evaluation strategies for learning algorithms of hierarchies. In *Proceedings of the 32$^{nd}$ Annual Conference of the German Classification Society (GfKl'08)*.

Carlsson, G., and Mémoli, F. 2010. Characterization, stability and convergence of hierarchical clustering methods. *J. Mach. Learn. Res.* 99:1425–1470.

Herlocker, J. L.; Konstan, J. A.; Borchers, A.; and Riedl, J. 1999. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99. ACM.

Kettenring, J. R. 2008. A Perspective on Cluster Analysis. *Statistical Analysis and Data Mining* 1(1):52–53.

Mueller, M., and Kramer, S. 2010. Integer linear programming models for constrained clustering. In Pfahringer, B.; Holmes, G.; and Hoffmann, A., eds., *Discovery Science*, volume 6332. Springer Berlin Heidelberg. 159–173.

Nakhleh, L.; Ringe, D.; and Warnow, T. 2002. Perfect phylogenetic networks : A new methodology for reconstructing the evolutionary history of natural languages. *Networks* 81(2):382–420.

Nemhauser, G. L., and Wolsey, L. A. 1988. *Integer and combinatorial optimization*. Wiley.

Sibson, R. 1973. SLINK: an optimally efficient algorithm for the single-link cluster method. *The Computer Journal* 16(1):30–34.

Sneath, P. H. A., and Sokal, R. R. 1973. *Numerical taxonomy: the principles and practice of numerical classification*. San Francisco, USA: Freeman.

Sokal, R. R., and Michener, C. D. 1958. A statistical method for evaluating systematic relationships. *University of Kansas Science Bulletin* 38:1409–1438.

Sørensen, T. 1948. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons. *Biol. Skr.* 5:1–34.

Tan, P.-N.; Steinbach, M.; and Kumar, V. 2005. *Introduction to Data Mining, (First Edition)*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.