Efficient and Effective Explanation of Change in Hierarchical Summaries

Deepak Agarwal
Yahoo! Research

Dhiman Barman
UC Riverside

Dimitrios Gunopulos
UC Riverside

Flip Korn
AT&T Labs–Research

Divesh Srivastava
AT&T Labs–Research

Neal E. Young
UC Riverside

ABSTRACT

Dimension attributes in data warehouses are typically hierarchical (e.g., geographic locations in sales data, URLs in Web traffic logs). OLAP tools are used to summarize the measure attributes (e.g., total sales) along a dimension hierarchy, and to characterize changes (e.g., trends and anomalies) in a hierarchical summary over time. When the number of changes identified is large (e.g., total sales in many stores differed from their expected values), a parsimonious explanation of the most significant changes is desirable. In this paper, we propose a natural model of parsimonious explanation, as a composition of node weights along the root-to-leaf paths in a dimension hierarchy, which permits changes to be aggregated with maximal generalization along the dimension hierarchy. We formalize this model of explaining changes in hierarchical summaries and investigate the problem of identifying optimally parsimonious explanations on arbitrary rooted one dimensional tree hierarchies. We show that such explanations can be computed efficiently in time essentially proportional to the number of leaves and the depth of the hierarchy. Further, our method can produce parsimonious explanations from the output of any statistical model that provides predictions and confidence intervals, making it widely applicable. Our experiments use real data sets to demonstrate the utility and robustness of our proposed model for explaining significant changes, as well as its superior parsimony compared to alternatives.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications–Datamining; F.2.m [**Analysis of Algorithms and Complexity**]: Miscellaneous

General Terms

Algorithms,Experimentation,Theory, Performance

Keywords

OLAP, hierarchical summary, change, parsimonious explanations, statistical model

KDD'07, August 12–15, 2007, San Jose, California, USA.

1. INTRODUCTION

Dimension attributes in data warehouses are typically hierarchical, and a variety of OLAP applications (such as point-of-sales analysis and decision support) call for summarizing the measure attributes in fact tables along the hierarchies of these attributes. For example, the total sales at different WalMart stores can be summarized hierarchically by geographic location (e.g., state/city/zip_code/store), by time (e.g., year/month/day/hour), or by product category (e.g., clothing/outerwear/jackets/brand). Existing OLAP tools help to summarize and navigate the data at different levels of aggregation (e.g., jackets sold in each state during December 2006) via drill-down and roll-up operators. OLAP tools are also used to characterize changes in these hierarchical summaries over time (e.g., the sales in December 2006 compared to their expectations over different locations), to detect anomalies and characterize trends. When the number of changes identified is large (e.g., the total sales at many locations differed significantly from their expectations), one seeks explanations.

Explanations may be verbose (e.g., a separate ad hoc explanation for the observed change at each location) or parsimonious (e.g., a single explanation for multiple observed changes, such as attributing a drop in sales at a large number of locations in Louisiana in 2005 to Hurricane Katrina). Parsimonious explanations are obviously more desirable and more effective than (ad hoc) verbose explanations. In this paper, we are interested in *parsimonious explanations* of changes in measure attributes (e.g., total sales) aggregated along an associated dimension attribute hierarchy (e.g., location).

Existing work has addressed the issue of explaining change between OLAP aggregates in terms of subaggregates [21] (we describe in more detail how this work differs from ours in Section 2), but these changes were expressed as outliers of point-to-point subaggregate comparisons. We seek a more holistic explanation. We propose a natural model that makes effective use of the dimension hierarchy and describes changes at the leaf nodes of the hierarchy (e.g., individual stores in the location hierarchy) as a composition of "node weights" along each node's root-to-leaf path in the dimension hierarchy – each node weight constitutes an explanatory term. For example, *overall sales in stores in California increased by a factor of three; sales in San Jose stores further grew by a factor of two (a six-fold total increase), whereas sales in Los Angeles stores increased less than the statewide increase by half (so a total growth factor of 1.5).* Formally, we assume that the dimension hierarchy remains fixed over time, and each data item (e.g., a record in a fact table) has a timestamp and is associated with a leaf node (e.g., an individual store) of the hierarchy. A hierarchical summary or snapshot (over some time interval) then associates with each node in the dimension hierarchy (e.g., store, zip_code, city, state) the ag-

gregated value of the measure attribute (e.g., total sales) of all data items (with a timestamp in that time interval) in its subtree.

If we consider two snapshots, it is clear that the changes between the trees can be expressed over the different levels of the dimension hierarchy in numerous possible ways. For example, if the sales at all California stores increased four-fold, we can model this change (among other possibilities) as a weight of four for each individual store, or a weight of four at the California state level, or as a weight of two at the California state level and a weight of two for each store. The important question is, what are the nodes in the hierarchy that explain the (most significant) changes parsimoniously.

A straightforward and intuitive attempt at identification of parsimonious explanations is a top-down approach. Starting from the roots of the two snapshots, compare aggregate values of the measure attributes at corresponding nodes. If the difference between the aggregates is completely "explained" by the composition of node weights along the path from the root to the parent of that node, no additional node weight (or explanatory term) is needed at that node. Otherwise, the node weight is set appropriately to the differential value with respect to the composition of weights along nodes for ancestor path from the root to that node. While straightforward and intuitive, such an explanation can be easily shown to not be optimally parsimonious. For example, if 1 out of 5 stores in Los Angeles that all used to have the same sales exhibited a 4-fold increase in sales, while the other 4 exhibited no change in sales, a top-down explanation would attribute a 1.6-fold (8/5) increase at the Los Angeles city level, and would then have to have additional explanations at each store to explain the differences with the city-level explanation - thus needing 6 explanatory terms. An optimally parsimonious explanation, on the other hand, needs only 1 explanatory term - a 4-fold increase at the anomalous store. This explanation is parsimonious in the sense that changes are aggregated with maximal generalization along the dimension hierarchy.

We envision that in many practical cases the user may want to compare a snapshot of the hierarchy with the values predicted by a model. Such an operation would be particularly useful for example when validating a forecasting model, or to identify conditions that are not properly modeled or to provide parsimonious explanation of anomalies that are expected to be related through the hierarchical structure. In this scenario, the use of statistical modeling would provide an expected value for each leaf of the hierarchy, with associated confidence intervals. Our proposed method can provide parsimonious explanation after incorporating uncertainty in the forecasts, quantified through confidence intervals.

1.1 Our Contributions

We summarize our contributions as follows:

- We formalize the notion of parsimonious explanation of change when comparing two hierarchical summaries, or when we compare a snapshot with the results of a forecasting model. To account for confidence intervals provided by a forecasting model, and to deal with noise, our model allows for a maximum tolerance between the observed change and the root-to-leaf explanation.

- We prove that optimally parsimonious explanations of our problem can be computed efficiently in polynomial time, proportional to the product of the number of leaves and the depth of the dimension hierarchy.

- To complement our conceptual and algorithmic contributions, we conduct a statistically sound experimental evaluation to understand the effectiveness and efficiency of our approach on real hierarchical datasets. We use a predictive model based on an exponentially weighted moving average (EWMA), which is widely used in time series applications. Our experiments demonstrate the effectiveness and robustness of our proposed approach for explaining significant changes, and show that it is more efficient than the worst-case bounds in practice.

The rest of this paper is structured as follows. In Section 2, we discusss related work. In Section 3, we formalize the variants of our problem of parsimonious explanation. Algorithms, along with their proofs of correctness and complexity analysis, are presented in Section 4. Experimental results along with description of the statistical models used are given in Section 5.

2. RELATED WORK

Hierarchies on data attributes have played a significant role in data warehouses, for which database operators such as the datacube have been developed to summarize and navigate the data at the different levels of aggregation [6]. In the data mining literature, several tools have been proposed for summarizing hierarchical data at a single time instance, including GMDL regions [16], Icecubes [10], and Hierarchical Heavy Hitters (HHH) [9, 7].

With respect to detecting changes in data, recent approaches include velocity density estimation [1] for visualizing change, windowed statistical testing [14] for detecting distributional changes, and histogram differencing [8] for identifying items which exhibit the largest changes in frequencies. However, these papers deal with flat (non-hierarchical) data. There have been a few papers explicitly dealing with hierarchical data. Zhang et al. studied change detection of (aggregated) time series corresponding to HHH IP prefixes in the IP address hierarchy [26]. Chawathe et al. studied the problem of change detection on semi-structured data, but for topological changes [23].

The problem of path explorations of hierarchies was studied in [22]. Here the user defines a set of linear constraints and the values in the datacube cells are predicted using the Maximum Entropy Principle. Given a supplied model, the technique finds the cells that are significantly different values from the expected values. Our problem is essentially the opposite: to find the best model that explains the changes. There is also some marginally related work on identifying bursts in hierarchical time series data, that is, the time intervals tightly capturing high arrival frequencies [27, 15].

Most related to our work is the DIFF operator for explaining differences in the datacube [21]. In their problem, a user selects two aggregates at the same level in the datacube which fixes some of the dimensions. The ratio between the selected aggregates is then explained in terms of the free dimensions, and subaggregates having deviating ratios explained recursively. This puts constraints on the intermediate node ratios, whereas our solution has the freedom to explain leaf aggregate changes in terms of intermediate node ratios, and is thus more parsimonious. In the example involving the Los Angeles stores mentioned earlier, the need to additionally "explain" the ratio of 8/5 at the Los Angeles city level internal node results in a verbose explanation using the DIFF approach. To demonstrate this verbose behavior of DIFF on real data, we shall experimentally compare against the DIFF operator in Section 5.

The problem of using compact hierarchical histograms for approximating leaf-level data was studied in [20] which employed a predefined hierarchy, similar to our approach but solved the dual problem: given a bound on the size of the synopsis (i.e., the number of explanations), find the synopsis that minimizes the error. Further, the paper considered three different partitioning functions and solved via dynamic programming to reduce distributive error met-

rics given a space bound Their LPM variant is the same problem studied in [21] but solved heuristically due to the expensive cost of distribute error metrics; the other partitioning functions find inferior solutions to LPM. Our work is based on the initial problem formulation presented in [4].

2.1 Connection to Wavelets

Recently [18] investigated a problem similar to our work. The proposed solution used the Haar wavelet representation to construct dataset synopses of minimum space. The use of the wavelet representation restricts this approach to less efficient (i.e., less parsimonious) explanations than our hierarchical parsimonious explanations. Another problem relevant to ours (for the case of binary hierarchies) is Haar wavelet compression with maximum-error metrics, introduced in [17]. The best current solution requires $O(n^2)$ time and $O(n)$ space to solve the dual problem [11] and, just as in [21], constraints are imposed at all nodes rather than just at the leaves, leading to less parsimonious solutions. [12] introduced the notion of unrestricted Haar wavelets and [13] defined the Haar+ tree as an improvement, but these exploit discretization of values and therefore are not comparable with our approach which allows for any (potentially infinite sized) domain. However [13] is equivalent to the model given by [20] when the hierarchy is restricted to binary trees. [13] presents provably good approximate algorithms to solve this problem in $O(R^2 n \log n \log^2 B)$ or $O(R^2 n \log^2 B)$ time (n is the size of the input, B the maximum number of coefficients in the synopsis and R the number of the examined values per coefficient), for general error metrics. Interestingly, our problem (with binary hierarchies) offers an alternative to Haar wavelet compression, yielding better answers with smaller complexity: $O(n \log n)$ for the primal problem and $O(n \log n \log \epsilon^*)$ for the dual. An algorithm that solves the dual problem (as in [20] or [13]) can be modified to solve the primal problem using a binary search procedure on B. Thus, these algorithms would need to run an additional $\log B$ factor slower if modified to solve our problem.

3. PROBLEM DEFINITION

In this section, we first define a natural change explanation model, which expresses the change between the leaf nodes of two hierarchical summaries as a composition of changes top-down from the root to the leaves of the tree.[1] We then discuss the model in the context of Occam's Razor to find a parsimonious explanation of change. Let S be a set of items from a domain D where the elements come from a well-defined hierarchy. Each item $i \in S$ has an associated measure value $v \in V$. The ordered pairs (i, v) could have been obtained by summing over the (projected) columns in a data warehouse fact table containing a multiset of `(itemID, value)` pairs where `itemID` is a dimension attribute and `value` is a measure attribute. Or they could have been aggregated over some time series window (eg, moving window average). Let T be a rooted tree obtained by inducing the dimension hierarchy on S, where the nodes correspond to different prefixes in the dimension hierarchy. We do not assume a total ordering over the dimension hierarchy, only that it is partially ordered with maximum height h. Let ℓ denote a leaf node and $m(\ell)$ denote some value attached to the leaf node ℓ. Given values attached to leaf nodes that represent some measure of change, we define a class of *hierarchical* change explanation models below.

DEFINITION 3.1. **Hierarchical Change Explanation:** *Given a hierarchy T and change values $m(\ell)$ attached to leaves ℓ, a hierarchical change explanation model is a complete, top-down composition of changes ("weights") $w(n)$ between nodes along the root-to-leaf path, for each leaf node.*

More formally, for each leaf node ℓ,

$$m(\ell) = W(\ell) \tag{1}$$

where

$$W(root) = w(root) \tag{2}$$

$$W(n) = w(n) + W(p(n)) \tag{3}$$

for tree nodes n where $p(n)$ is the parent node of n. A solution to this system gives weights $w(n)$ for each n. In fact, if $\mathcal{P}(n)$ denotes the ancestor path from the root down to a tree node n, then by unraveling Equations 1-3, our problem is to find weights $w(n)$ of each node n in the tree subject to the constraints $m(\ell) = \sum_{n \in \mathcal{P}(\ell)} w(n)$. Since this system of equations is under-specified, there are multiple solutions each of which provides a hierarchical change explanation.

In general, the change values $m(\ell)$ are obtained as some discrepancy measure $d(m_1(\ell), m_2(\ell))$ between two sets of values observed for the hierarchy T. For example, consider the Census dataset [5] where we have population counts $m_1(.)$ and $m_2(.)$ for zip codes and a geographical hierarchy that defines aggregations at state, county and city levels at two different snapshots T_1 and T_2 as exemplified in Figures 1(a) and (b). Here, 3D $m(\ell) = d(m_1(\ell), m_2(\ell)) = log(m_2(\ell)/m_1(\ell))$. In general statistical anomaly detection problems, $m_1(\ell)$ is forecasted value based on some statistical model that captures normative behavior and $m_2(\ell)$ is the actual observed value with higher discrepancy being indicative of anomalous behavior.

3.1 Parsimonious Explanation

Definition 3.1 provides a rich class of hiearchical change explanation models; we provide a couple of examples that are trivial to compute but sub-optimal and then provide a notion of an optimal or parsimonious hierarchical change explanation model.

One possible assignment of weights that is used in anomaly detection applications is the one that completely ignores the hierarchical structure and assigns each leaf node ℓ in T a weight of $m(\ell)$, and 0 to the non-leaf nodes. We call this the "non-hierarchical" model, comparison w.r.t this model helps in quantifying the gain achieved by using the hierarchy. Figure 1(c) shows a non-hierarchical assignment for the trees T_1 and T_2 shown in Figures 1(a) and (b), respectively. The leaf-level nodes encircled boldly have non-zero assigned weights. Using trees T_1 and T_2, we construct a third tree as in Figure 1(c) such that the value associated with a leaf is log of the ratio of the correspoding leaf counts. Assuming the existence of a rollup operator that aggregates values of children to the parent, another possible assignment is top-down, which recursively assigns weights from the root down such that $m(\ell) = W(n) = \sum_{u \in \mathcal{P}(n)} w(u)$, for all leaves *and* intermediate nodes n. Figure 2(a) provides an example where values at each snapshot are rolled up using the sum operator. Both of these assignments satisfy the equations of hierarchical change explanation but are not necessarily parsimonious: the former ignores all opportunities to group leaves with equal differences in the same subtree whereas the latter is too greedy in that it groups unequal leaf differences.

A node weight $w(n) = 0$ implies no change to node n relative to $p(n)$ and does not need to be reported in an explanation. Thus, the *explanation size* is the number of non-zero weights in the explanation. Applying Occam's Razor, we prefer an explanation of the

[1] Our method works for both multiplicative and additive compositions by transforming the former to latter using logarithms; we illustrate using the additive scale.

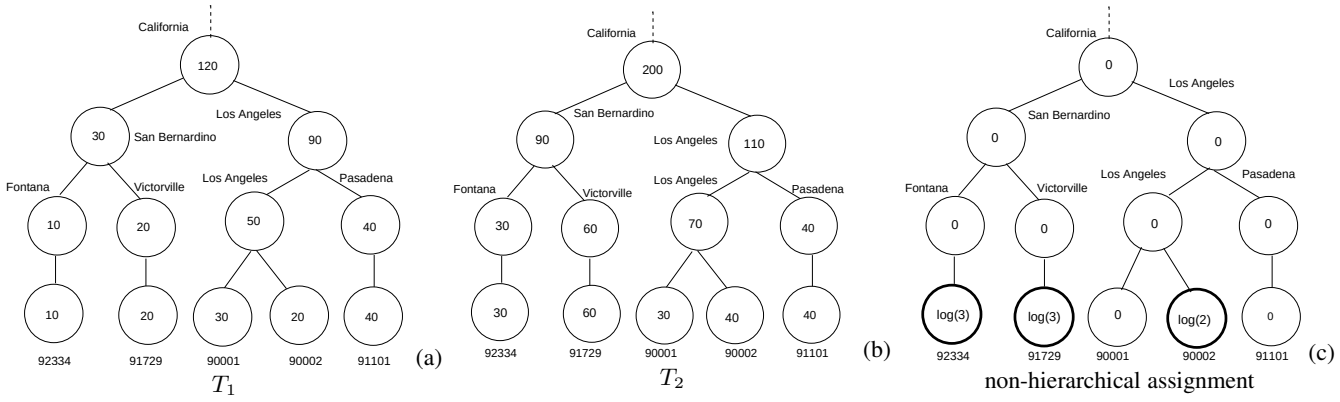


Figure 1: Each distinct ZIP code appears as a leaf in the tree along with its associated population counts (shown inside the node). The population count for internal nodes is the sum of the population counts from the leaves in its subtree. (a) and (b) give trees gives two snapshots T_1 and T_2, of hierarchy T respectively; (c) shows a non-hierarchical weight assignment

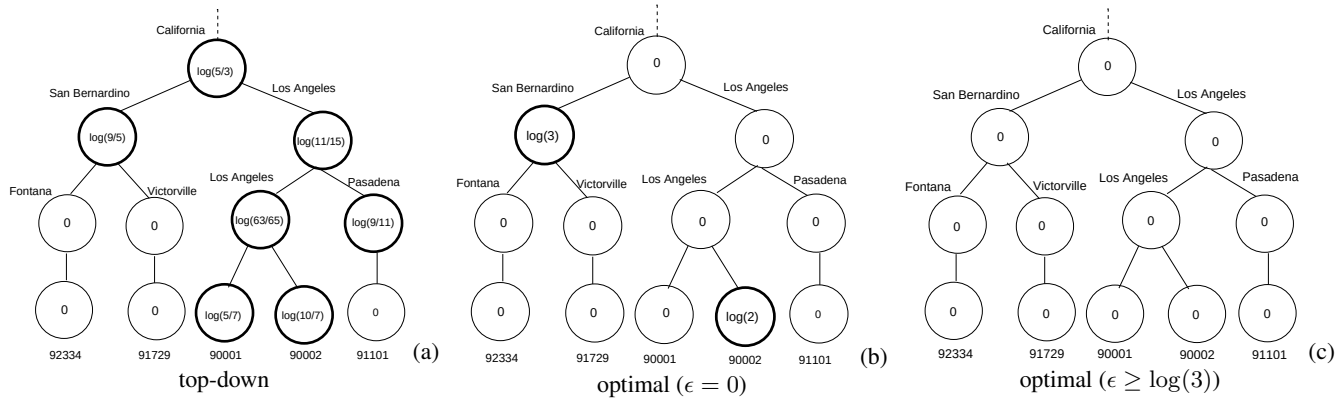


Figure 2: Weight assignment based on top-down and optimal assignments.

smallest size. Therefore, we define a *parsimonious explanation* of hierarchical change as one with the smallest explanation size, that is, the minimum number of weights not equal to zero. Consider Figure 2(b), which is able to explain the changes using only 2 non-zero weights compared to 3 for the non-hierarchical strategy and 7 for the top-down one; in fact, it is optimal. We describe the algorithms which lead to assignments in Figure 2(b)-(c) in Section 4.

However, this explanation model has certain shortcomings. Often one wants to compare a snapshot with expected values; large deviations from these values can be reported as anomalies. Statistical forecasting models (e.g., based on moving averages) typically yield confidence intervals based on a supplied confidence level. An important shortcoming of the current model is that it does not work with such a forecasting model because its formulation does not deal with ranges of possible values. In addition, the model is sensitive to noise. Intuitively, we would like to capture similar changes among related leaves which may not have exactly equal differences but are roughly the same. For example, if two sibling leaves have differences of 1.98 and 2.02, we may wish to describe this at the parent using a difference of 2. Since the deviations from this description at the leaves are small (1%), we may tolerate this error as being a good enough approximation to report only significant changes and to avoid overfitting the data. Our original description above, which only allows exact matches, does not allow this.

In order to ameliorate this, we extend the definition to allow a tolerance parameter ϵ on the values of the leaves. We allow weights on the nodes that result in differences of at most ϵ between two leaves in the snapshots. We assume that in practice this tolerance parameter will be provided by the confidence interval of the prediction model, which can be different from leaf to leaf, so the model allows different tolerances $\epsilon(\ell)$ at each ℓ. Specifically, we assign weights such that $|m(\ell) - W(\ell)| \leq \epsilon(\ell)$ for each ℓ, where $W(\ell) = \sum_{u \in \mathcal{P}(\ell)} w(u)$.

To see the connection with a forecasting model, we assume $m(\ell) = d(m_1(\ell), m_2(\ell)) = m_2(\ell) - m_1(\ell)$. In fact, rewriting this equation $m_2(\ell) - (m_1(\ell) + \epsilon(\ell)) \leq W(\ell) \leq m_2(\ell) - (m_1(\ell) - \epsilon(\ell))$ and denoting $m_1(\ell) + \epsilon(\ell)$ and $m_1(\ell) - \epsilon(\ell)$ by $UB(\ell)$ and $LB(\ell)$, respectively, clearly shows how to use output from a forecasting model in our framework. $LB(\ell)$ and $UB(\ell)$ are lower and upper confidence bounds that are obtained from the estimated forecasting distribution. One possibility which works for symmetric distributions is to choose $m_1(\ell)$ as the predicted mean and $\epsilon(\ell)$ to be proportional to the predicted standard deviation, the constant of proportionality depending on the desired coverage of the confidence interval. For instance, a choice of 1.96 under a Gaussian assumption on the statistical distribution of our node values guarantees 95% coverage. In general, our method is agnostic to the particular choice of forecasting model; the only requirement is the availability of $LB(\ell)$ and $UB(\ell)$. This makes it a highly general purpose method with wide applicability in anomaly detection problems involving hierarchical data where changes are expected to be spatially clustered in subregions of the hierarchy.

We now define our parsimonious explanation model, which allocates a tolerance budget along each path that can be distributed among the individual path nodes in any fashion while maintaining the constraint $|m(\ell) - \sum_{n \in \mathcal{P}(\ell)} w(n)| \leq \epsilon(\ell)$.

DEFINITION 3.2. **Hierarchical Parsimonious explanation:** *Given a set of leaf changes $m(\ell)$ and a tolerance budget $\epsilon(\ell) \geq 0$ on the total sum of weights along the path to ℓ for all leaves ℓ, a* hierarchical parsimonious explanation *of change finds the smallest explanation size, that is, minimum number of node weights $w(n)$ s.t. $w(n) \notin [-k, k]$; $k \geq 0$.*

In definition 3.2, for positive tolerances, only $k = 0$ is of interest to us in practice. However, to facilitate comparison with DIFF algorithm, extended definition which allows thresholding on positive values of k is necessary as we shall see later in section 5.

4. ALGORITHMS

In this section, we describe an algorithm to compute optimal weight assignments (that is, minimizing the explanation size), for the problem defined in Section 3 (Definition 3.2). The algorithm presented here generalizes this problem by allowing any supplied error tolerances for the leaves as well as intermediate nodes of the hierarchy.

The problem with $\epsilon = 0$ is a special case of the following problem: *Given real matrix A and vector b, find x such that $Ax = b$ minimizing the number of non-zero x_i's.* That problem is not only NP-hard, but is not approximable within $2^{\log^{1-\delta} n}$ for any $\delta > 0$ (in polynomial time, assuming NP is not contained in quasi-polynomial time) [2, 3]. For the special case studied here we give a fast, exact algorithm. We first describe the algorithm intuitively, and then present it formally.

The algorithm makes two passes over the tree: the first bottom-up and the second top-down. In the first pass the algorithm computes a tentative set of "best" incoming partial sums for each node, using dynamic programming. We prove that the best partial sums for a node are those that allow the node to incur no cost for its own weight and, simultaneously, to provide best partial sums for the maximum number of children. This set of best partial sums for a node is a union of closed intervals, at most one for each leaf.

In the second pass the algorithm works down from the root to assign weights. Each node chooses its own weight so as to benefit the maximum number of its children. If the incoming partial sum for a node is one of the best for the node, it can do this without incurring a cost at the node. Otherwise, the node incurs a cost of 1 for its own weight, which it chooses to benefit the maximum number of its children. We illustrate this process in Fig. 3 for the case $\epsilon = 1$.

Note that instead of taking $k > 0$, one may add $k|\mathcal{P}(\ell)|$ to each $\epsilon(\ell)$, then take $k = 0$. This expands the set of feasible weightings. We present the algorithm for the general case $k > 0$ for compatibility with [21], which we shall compare against in Section 5.

DEFINITION 4.1. *For any subtree T' and real value x, define $\mathrm{cost}(x, T')$ to be the minimum cost of any feasible labeling of T', given that the partial sum coming into the root of T' from above is x. (Formally, this is the minimum cost of any feasible labeling of the tree T' in isolation, where each leaf change $m(\ell)$ has been decreased by x.)*

Define $\mathrm{bestcost}(T') = \min_x \mathrm{cost}(x, T')$ *and* $\mathrm{bestsums}(T') = \{x : \mathrm{cost}(x, T') = \mathrm{bestcost}(T')\}$.

We start with the observation that a bad incoming partial sum increases the cost of T' by only 1.

Let $[x \notin S]$ denote 0 if $x \in S$ and 1 otherwise.

LEMMA 4.2. *For any subtree T' and real x,* $\mathrm{cost}(x, T') = \mathrm{bestcost}(T') + [x \notin \mathrm{bestsums}(T')]$

PROOF. Since the costs are integers, it's enough to prove that $\mathrm{cost}(x, T') \leq \mathrm{bestcost}(T') + [x \notin \mathrm{bestsums}(T')]$.

Let x' be a partial sum achieving $\mathrm{bestcost}(T')$, and let w be a corresponding min-cost weighting of T' for partial sum x'. Adding $x' - x$ to the weight of the root of T' gives a feasible weighting for T' with partial sum x, and increases the cost of w by at most 1. □

Next we prove a recurrence which will be the basis for the algorithm. Let $A \oplus B$ denote $\{a + b : a \in A, b \in B\}$.

THEOREM 4.3. *Let T' be any subtree with immediate subtrees $T'_1, T'_2, \ldots, T'_c$. Then* $\mathrm{bestsums}(T')$ *equals*

$$[-k, k] \oplus \big\{ z : z \text{ minimizes } |\{i : z \notin \mathrm{bestsums}(T'_i)\}| \big\}.$$

PROOF. Let $\mathrm{kiddiff}(z, T')$ denote $|\{i : z \notin \mathrm{bestsums}(T'_i)\}|$.

Let $\mathrm{bestkiddiff}(T')$ denote $\min_z \mathrm{kiddiff}(z, T')$.

Fix x and T'. By definition, $\mathrm{cost}(x, T')$ equals

$$\min_y \quad [y \notin [-k, k]] + \sum_i \mathrm{cost}(x + y, T'_i)$$

(y is the weight given to the root of T'). By lemma 4.2, this is

$$\big(\sum_i \mathrm{bestcost}(T'_i)\big) + \min_y \quad [y \notin [-k, k]] + \mathrm{kiddiff}(x + y, T').$$

The term on the left is independent of x, while the $\min_y \ldots$ term on the right will equal $\mathrm{bestkiddiff}(T')$ for some x (e.g. when $y = 0$ and x minimizes $\mathrm{kiddiff}(x, T')$).

Thus, $x \in \mathrm{bestsums}(T')$ (that is, x minimizes $\mathrm{cost}(x, T')$) iff

$$\exists y \in [-k, k] : \mathrm{kiddiff}(x + y, T') = \mathrm{bestkiddiff}(T').$$

Taking $z = x + y$, this condition is equivalent to

$$x \in [-k, k] \oplus \{z : \mathrm{kiddiff}(z, T') = \mathrm{bestkiddiff}(T')\}.$$

□

Theorem (4.3) gives a recurrence relation for $\mathrm{bestsums}()$. Using this recurrence, computeDS(T, d) (Algorithm 1) uses dynamic programming to compute, for all subtrees T', $\mathrm{bestsums}(T')$ and $\mathrm{kidopt}(T') = \big\{z : z \text{ minimizes } |\{i : x \notin \mathrm{bestsums}(T'_i)\}|\big\}$.

Algorithm 1 computeDS(subtree T', leaf values m)

```
1: If T' is a leaf (a single node ℓ):
2:     let kidopt(T') ← {m(ℓ)} ⊕ [−ε(ℓ), ε(ℓ)]
3: else:
4:     for each subtree T'_i of T': computeDS(T'_i, m)
5:     kidopt(T') ← {z minimizing |{i : x ∉ bestsums(T'_i)}|}
6: bestsums(T') ← kidopt(T') ⊕ [−k, k]
```

The algorithm first calls computeDS(T, m) to compute $\mathrm{kidopt}(T')$ and $\mathrm{bestsums}(T')$ for all subtrees T'. It then weights T by calling weightTree$(0, T)$ (Algorithm 2).

LEMMA 4.4. *weightTree(x, T') finds a feasible weighting of T' (assuming incoming partial sum x) of optimal cost* $\mathrm{cost}(x, T')$.

PROOF. From Theorem 4.3, computeDS correctly computes kidopt and $\mathrm{bestsums}$. Theorem 4.3 assures that y exists in the second line of weightTree(). A standard proof by induction shows that the weighting is feasible.

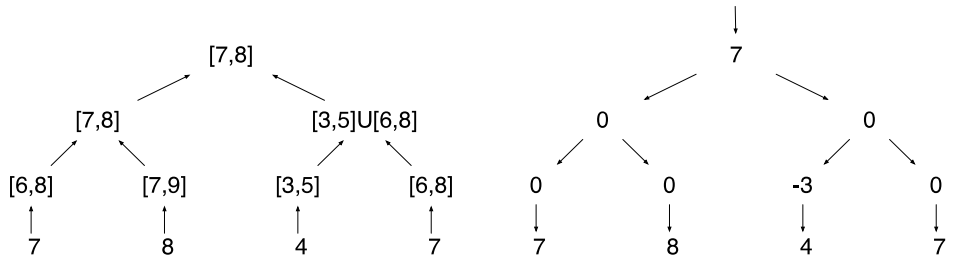


Figure 3: Computing an optimal node weighting $(k = 0, \epsilon = 1)$**.**

Algorithm 2 weightTree(partial sum x, subtree T')

1: if $x \in \text{bestsums}(T')$:
2: pick $y \in [-k, k]$ s.t. $x + y \in \text{kidopt}(T')$
3: else: let $y \leftarrow x' - x$ for any $x' \in \text{kidopt}(T')$
4: give the root of T' weight y
5: for each subtree T'_i of T': weightTree$(x + y, T'_i)$

To finish we consider the cost. By inspection weightTree(x, T') chooses a root weight y so $x+y \in \text{kidopt}(T')$. Thus (assuming by induction that the subtrees are weighted optimally), the total weight for nodes in the subtrees $\{T'_i\}$ is $\text{bestcost}(T')$. In addition, at the root we pay $[y \notin [-k, k]]$. By inspection of weightTree(), this equals $[x \notin \text{bestsums}(T')]$. Thus, the total cost of our weighting is $\text{bestcost}(T') + [x \notin \text{bestsums}(T')]$. By Lemma 4.2, this is best possible. □

LEMMA 4.5. *The running time of the algorithm is $O(hN \log N)$, where h is the height of the tree and N is the number of leaves.*

PROOF. (Sketch) The running time of the algorithm is dominated by the time it takes to compute the optimal shift for each node. We note that the total size of the optimal shifts for a node is bounded by the number of leaves in the subtree rooted at that node. Computing the optimal shifts of a parent node from the labeling of its children nodes requires sorting and merging of the children node labels.

For any tree, assume that at depth d there are $c(d)$ nodes. A node v_i at depth d will have $lv(v_i)$ leaves in its subtree, but $\sum_1^{c(d)} lv(v_i) = N$ where N is the number of leaves. This bounds the size of the labeling at that node. The cost of sorting and merging M nodes is $M \log M$. So the total processing time at level d is given by

$$\sum_1^{c(d)} lv(v_i) \log lv(v_i) \leq N \log N$$

Hence, total time of processing over the entire tree is

$$\sum_{d=0}^{h} N \log N = O(hN \log N)$$

□

5. EVALUATION

In this section, we investigate both the effectiveness and the efficiency of the proposed algorithms and the outputs they generate using real data. We evaluate the effectiveness of our proposed change detection model according to its ability to capture interesting hierarchical changes as well as the robustness and stability of the output under small perturbations of error tolerance.

5.1 Experimental Setup

We define stability to measure the sensitivity of the set of explanation weights as a function of confidence level c. Let S_l^c be the set of nodes at level l where "explanations" occur. Then the stability of the output at level l, given a change in tolerance parameter from $c - \Delta c$ to c, is given by $S_c = \frac{|S_l^{c-\Delta c} \cap S_l^c|}{|S_l^c|}$, where $c - \Delta c$ refers to the previous value of c.

We used the following two real data sets: Census, which gives population counts for a geographical hierarchy given by state/county/city/zip_code [5]; and WorldCup, which is a Web log over a duration of several months of URL accesses to files having a maximum path length of 7 [25]. Note that the hierarchy induced by the URL file paths are not homogeneous, that is, the nodes have different fanouts and the paths have different depths. The Census data has approximately 81,000 leaf nodes and 130,000 total nodes in the tree. The maximum height of the tree is 5 (including the root which stands for the whole country). The World Cup datasets have about 4300 leaf nodes and around 4500 total number of nodes. In the non-homogeneous World Cup datasets, the maximum height of the tree is 8 including the root.

All experiments were run on a Pentium(R) machine with 4 CPU and clock speed 2.66GHz.

5.2 Forecasting Model

We provide a description of the models that were used in our experimental evaluation on real data. We do not claim any novelty here and use the popular exponentially weighted moving average (EWMA) for both our datasets. In our analysis, we assume a Gaussian distribution for the node values. Although this may not be a reasonable assumption for count data on the original scale, it is often a good approximation on a transformed scale (log and squared-root are widely used for count data). For our example datasets, we consider an exponentially weighted moving average (EWMA) to model the transformed leaf counts. We use a single smoothing parameter for all our leaf nodes, the value being selected to minimize the average predictive squared-error loss on a tuning set across all nodes. We assume there is no seasonality in our time series. This is the case for both the data sets analyzed in this paper. Consider a single leaf node and let $\hat{x_t}$ denote predicted value at time t based on data until time $t - 1$.

For EWMA, $\hat{x_t} = m_{t-1}$; and

$$\hat{m_t} = \lambda x_t + (1 - \lambda) m_{t-1}; \tag{4}$$

where $\lambda \in (0, 1)$ is a smoothing constant with higher values giving less weight to historical observations. Equation 4 can be shown to be a steady state model obtained form a simple random walk model given by

$$x_t = m_t + \epsilon_t \tag{5}$$

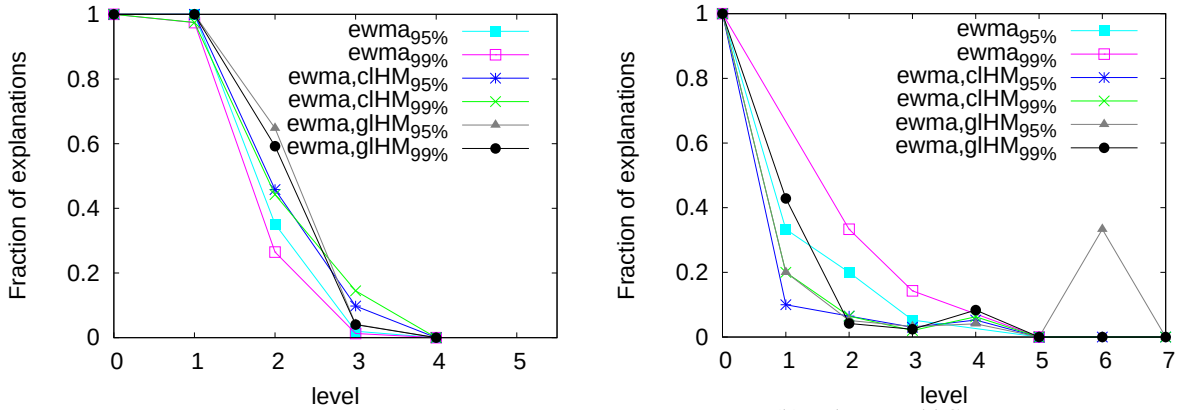
$$m_t = m_{t-1} + \gamma_t \tag{6}$$

EWMA		EWMA with Cluster Harmonic Mean of Variances	
node	weight	node	weight
Illinois/Lake/LibertyVille/27923	1.10083	Illinois/Lake/66027/27923	1.06641
Texas/Parker/FortWorth	1.00523	Texas/Parker/FortWorth	1.00523
Illinois/St. Clair/47423/69550	0.870329	Illinois/Kankakee/Bourbonnais/46513	0.815077
Minnesota/Le Seur/0/Mankato City/	0.868159	Texas/Hays/Austin	0.654658
Texas/Hays/Austin	0.654658	Illinois/Lake/LibertyVille/27923	0.564962

Table 1: Top 5 explanation nodes in the Census data sets in the descending order of relative error using the prediction model for 95% confidence values

EWMA		EWMA with Har. Mean of Cluster Var.	
node	weight	node	weight
/images/jersey_arg_res.gif	41.0524	/english/playing/download/download_memo.html	42.9246
/images/jersey_arg_off.gif	35.8748	/images/f98dnld_memo_sc2.gif	40.2364
/images/jersey_nor_off.gif	32.2865	/images/f98dnld_memo_sc3.gif	39.7132
/images/jersey_nor_res.gif	32.2618	/images/f98dnld_memo_sc1.gif	39.3961
/images/11189.jpg	10.0186	/images/f98dnld_memo_sc5.gif	38.8777

Table 2: Top 5 explanation nodes in the World cup datasets in the descending order of relative error using the prediction model for 95% confidence values



(a) Below, Census Data (b) Below, World Cup Data

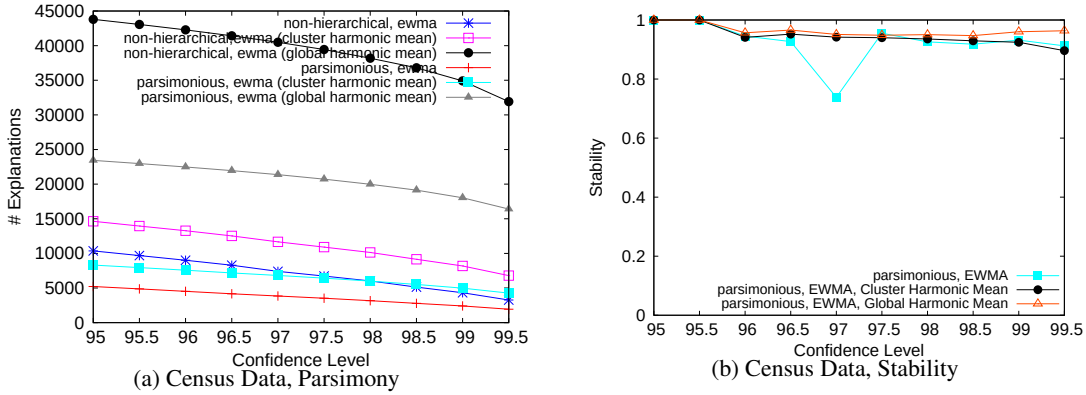
Figure 4: The fraction of explanation nodes at level l which have ancestors and descendants in the explanation, for Census and World Cup data. clHM refers to Harmonic Mean of Cluster Variances; glHM refers to Harmonic Mean of Global Variances; subscripts denote the confidence values.

where x_t is observed value at time t, m_t may be thought of as the truth, ϵ_t and γ_t are uncorrelated random variables with zero means and variances $V(\epsilon)$ and $V(\gamma)$ [19]. At steady state, the optimal prediction obtained through Equation 5 reduces to Equation 4 with an optimal value of λ given by $(\sqrt{(1+4R)}-1)/2R$; $R = V(\epsilon)/V(\gamma)$ and the predictive variance at time t based on data up to time $t-1$ is given as $V = V(\epsilon)/(1-\lambda)$. Thus, estimators which give more weight to historical data achieve more smoothing and have lower predictive variance.

In our scenario, we are dealing with $N > 1$ time series corresponding to the leaf nodes giving rise to pairs $(\lambda_i, V_i(\epsilon))$ to be estimated. For simplicity, we assume $\lambda_i = \lambda$ for all i and estimate the optimal value by minimizing squared-error predictive loss on a tuning set (see [24] for an example of such an estimator). For $V_i(\epsilon)$, we test the following variations: a) separate parameter for each series; b) one parameter for a set of sibling leaf nodes (nodes sharing same parent); c) one parameter for all the time series. We select the best model as the one that minimizes average predictive log-likelihood on the tuning set; this captures both the mean and variance properties of the predictive distribution.

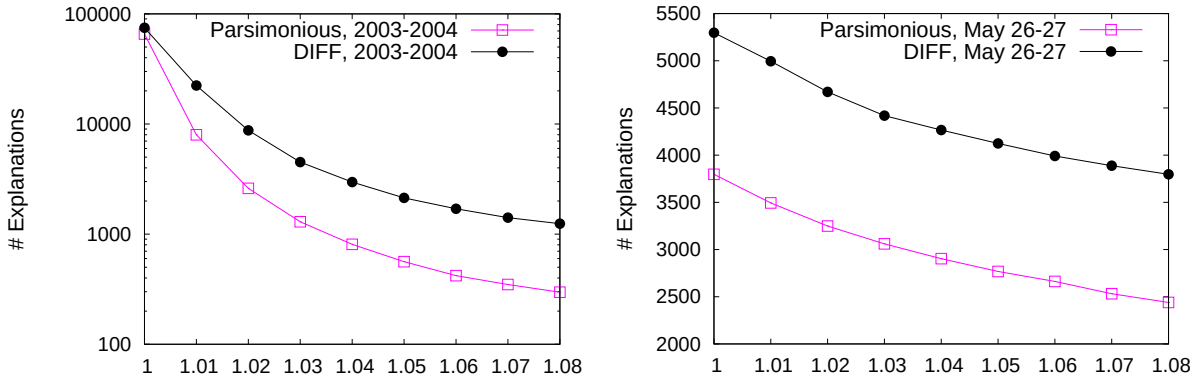
We analyzed two datasets: Census and WorldCup. The Census data have yearly population numbers from $2000-2004$. We used $2000-2003$ as our training period and 2004 as our test period on which we detect anomalies. We have approximately $81K$ leaf nodes on the test period. Since all number are positive (a count of 0 is interpreted as missing data), we modeled the data using a EWMA of 4 time points on the log scale. days on the log scale.

We considered daily counts for the World Cup data and used 32 time points.The 31^{st} time point is used as a tuning set to select the smoothing parameter λ and the variances. The last time point is used as our test set. The optimal value of λ in this case was 0.8 and the model with separate variances for each node also turned out to be the best one for this data. Unlike the Census data, the World Cup hierarchy is not homogeneous, i.e., the nodes have different fanouts and paths have different depths. Also, the structure of the tree is dynamic with new nodes appearing and some old nodes becoming inactive over time. We restrict ourselves only to nodes that occurred at least twice in the last 10 time points. This removes



(a) Census Data, Parsimony (b) Census Data, Stability

Figure 5: (a) Number of explanations; and (b) Stability as function of confidence value for non-hierarchical and parsimonious algorithm on Census data



(a) Census Data (b) World Cup Data

Figure 6: Comparison with related work DIFF operator in terms of number of explanations on (a) Census data (b) World Cup data. k is per-node tolerance and is in linear-scale (not log).

nodes that have a small mean and are not of interest, providing a set of approximately $5.5K$ leaf nodes to be monitored. Since zero counts are common in these time series, a log transform to achieve symmetry is not an option here. Instead, we use a squared-root transformation which, for count data, is known to stabilize variance, achieve approximate symmetry and makes the assumption of a Gaussian distribution reasonable.

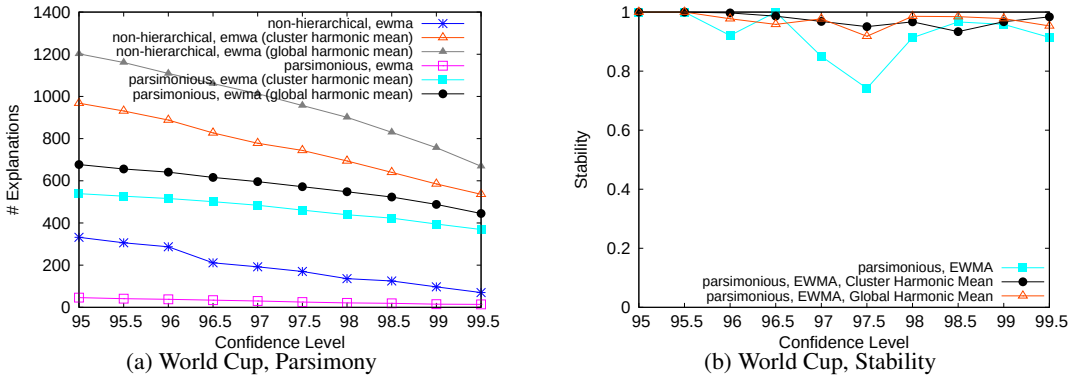
5.3 Goodness of Explanation Model

For illustrative purposes, we present the top 5 nodes in resulting explanations based on absolute magnitude (difference of the weights from 0). Table 1 shows the 5 nodes in the Census datasets which are explanations and whose absolute relative error is among the Top 5. We show the lists for two different prediction models: a superior EWMA model, with a separate variance component for each leaf and an inferior EWMA model with a separate but fixed variance for each leaf in a cluster (nodes under same parent) which is set to the harmonic mean of the individual leaf variances in the cluster. Note that some nodes are common explanation nodes under both the models such as Illinois/Lake/Libertyville/27923, Texas/Parker/ForthWorth and Taxas/Hays/Austin. Similar examples are shown for World Cup datasets in Table 2.

Figure 4 considers hierarchical relationships among the explanation nodes. If many nodes in the explanation set have descendant nodes that are also part of the explanation, then this indicates the importance of hierarchical explanations, as descendant nodes are needed to explain trends that are different from the ancestors in the explanation; these could be stronger trends or counter-trends compared to the ancestor node. Thus, for each node in the explanation set, we counted how many descendants below it are also part of the explanation. Let $V(l)$ be the number of explanation nodes at level l and $V(l)^B$ be the number of explanations nodes at level l which have at least one explanation node as descendant. Then we compute $V(l)^B/V(l)$. In these plots, level 0 indicates the root. We observe that significant number of counties ($> 25\%$) have cities which have different trends in population under all prediction models.

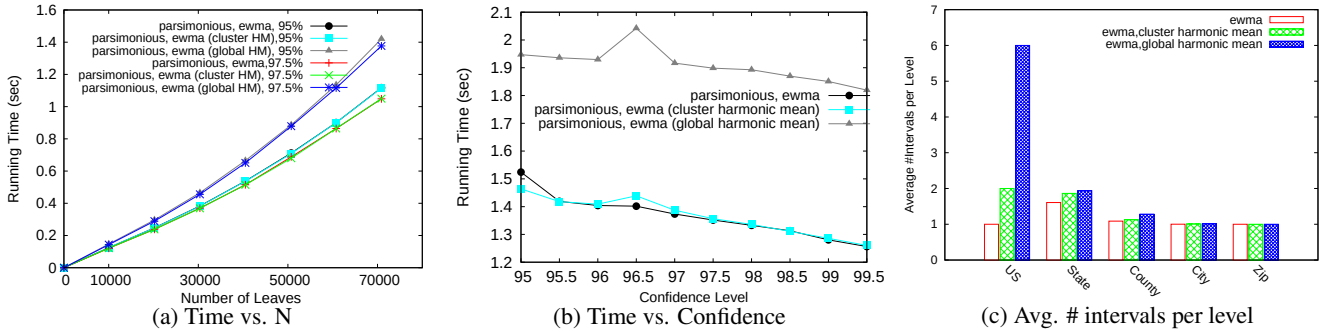
5.4 Parsimony

In Figure 5 we compare the parsimonious explanations against those obtained by the naive non-hierarchical approach. We use three different prediction models in which the mean of prediction is given by the EWMA model but the variances $V(\epsilon)$ are different – EWMA with a separate variance per leaf, same variance for each element in a cluster, set to the harmonic mean of cluster variances (leaves belonging to same parent), single variance for each leaf, set to the harmonic mean of the global variances. Note that the lat-



(a) World Cup, Parsimony (b) World Cup, Stability

Figure 7: (a) Number of explanations; and (b) Stability as function of confidence for non-hierarchical and parsimonious algorithms World Cup data.



(a) Time vs. N (b) Time vs. Confidence (c) Avg. # intervals per level

Figure 8: Running time (in sec) of parsimonious algorithm as a function of (a) number of leaves, N and (b) confidence levels on Census data. (c) shows the average number of intervals per level on Census data.

ter two estimates underestimate variability for a large fraction of nodes; we choose them to study the effect of inferior prediction model on our algorithms.

As the confidence level increases, the precision decreases and therefore, all the curves show decreasing trends monotonically. In Figure 5(a) we observe that the parsimonious model with EWMA offers the best parsimony with the smallest number of explanations, followed by non-hierarchical on EWMA model, thus showing the advantage of parsimonious algorithm. As expected, the performance of EWMA model with global harmonic mean of the variances perform worst in terms of parsimony.

We show the parsimony of our algorithm by comparing with the DIFF operator [21]. Since the technique in [21] puts constraints on the intermediate nodes, we have to modify our algorithm so that we have a tolerance parameter k in each node, and we use this model in this comparison shown in Figure 6. We compare two different snapshots - year 2003 and 2004 from Census data; and May 26 and 27 from World Cup data. It is to be noted that x-axis in Figure 6 is per-node tolerance, $k > 0$ and it is not in log-scale.

The improvement in the number of explanations when using our model is significant, up to two orders of magnitude. The improvement is more evident in the Census data, which exhibit hierarchical trends, compared to the World Cup data.

Figure 5 (b) show the average stability across all levels for both non-hierarchical and parsimonious algorithms. We observe that with increase in confidence level, the stability decreases since the set of nodes which are explanations changes. Since Census data is homogeneous with 4 levels, we observe almost monotonic change in stability with increase in confidence level except for the parsimonious algorithm with the best EWMA model at confidence level 97. Similarly, we observe parsimony and stability on World Cup datasets in Figure 7.

5.5 Efficiency

In Figures 8 (a)-(b), we show the runtime of the parsimonious model on Census data (minimum time over 5 runs) as function of the number of leaves, N and confidence level respectively (using all three models). First, the increase in running time with N follows $O(hN \log(N))$ growth. Second, we observe that all the algorithms show a decreasing trends (prominently in Census data) in running time with increase in confidence level (increasing error) leading to a small number of intervals in the non-leaf nodes. Third, we observe that the parsimonious algorithm with EWMA model has least running time complexity. That means, variances in individual leaves can summarize changes well whereas the algorithm which uses harmonic mean of variances cannot summarize the changes that well and thus leads to many explanations (and intervals) up in the trees. To vary the number of leaves, we sample each leaf with some probability to be included in the tree. We also show it for two different values of confidence levels.

In Figure 8(c), we show the space complexity of the 3 parsimonious algorithms on Census (similar trend on World Cup data but better), averaged over all nodes per level. We observe that the average number of intervals per node is very close to 1 except for par-

simonious algorithm using EWMA model with same variance per leaf node (estimated by the Harmonic Mean of the individual leaf Variances). Furthermore, we observe that the same parsimonious algorithm has higher running time and larger number of average intervals at different levels.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a natural model for explaining changes in hierarchical data and formulated two problem variants for finding a parsimonious explanation in this model. Our model makes effective use of the hierarchy and describes changes at the leaf nodes as a composition of node weights along each path of each root-to-leaf path in the hierarchy. We designed algorithms to minimize the explanation size for both problem variants. Despite the fact that assigning node weights optimally is an under-constrained problem, we have shown that it is not NP-hard and that our algorithms require time proportional to the product of the number of leaves and the depth of the dimension hierarchy.

We evaluated our approach on real data to demonstrate both its efficiency and effectiveness. In practice, the performance and space usage of our algorithms are much less than the worst-case bounds. On population census data, the explanations discovered (counter) trends, mainly at the city-level. We made similar observations when we analyzed HTTP traffic logs from the FIFA World Cup hosting site. Our approach can also be used to reveal "interesting" anomalies in hierarchical data when used in conjunction with a statistically sound predictive model that forecasts values within confidence intervals. These anomalies were explained more parsimoniously using our algorithm compared to the leaf-level anomalies that the predictive model detects.

We are currently extending our approach to multiple dimensions, which presents several non-trivial challenges due to the existence of multiple parents in the hierarchy. Another natural extension we have considered for future work is where there is a global budget on error tolerance for the entire tree. Although we have found a polynomial solution, its complexity appears to be significantly higher than the problems studied in this paper, and its feasibility on massive data sets remains to be shown.

Acknowlegments

We thank the anonymous reviewers for excellent comments on the paper. The work of the third author is supported by NSF awards 0330481 and 0534781.

7. REFERENCES

[1] On change diagnosis in evolving data streams. *IEEE TKDE*, 17(5):587–600, 2005. Charu C. Aggarwal.

[2] E. Amaldi and V. Kann. On the Approximability of Minimizing Nonzero Variables or Unsatisfied Relations in Linear Systems. *TCS*, 209(1-2):237–260, 1998.

[3] Sanjeev Arora, László; Babai, Jacques Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *J. Comput. Syst. Sci.*, 54(2):317–331, 1997.

[4] Dhiman Barman, Flip Korn, Divesh Srivastava, Dimitris Gunopulos, Neal E. Young, and Deepak Agarwal. Parsimonious Explanations of Change in Hierarchical Data. In *Proc. of ICDE 2007*.

[5] Census (population vs. location), 2000-2004. `http://www.census.gov/popest/datasets.htm`.

[6] Surajit Chaudhuri and Umeshwar Dayal. An overview of data warehousing and olap technology. *SIGMOD Record*, 26(1):65–74, 1997.

[7] Graham Cormode, Flip Korn, S. Muthukrishnan, and Divesh Srivastava. Finding Hierarchical Heavy Hitters in Data Streams. In *Proc. of VLDB*, pages 464–475, 2003.

[8] Graham Cormode and S. Muthukrishnan. What's new: Finding significant differences in network data streams. In *Proc. of IEEE INFOCOM*, pages 1534–1545, 2004.

[9] Cristian Estan, Stefan Savage, and George Varghese. Automatically inferring patterns of resource consumption in network traffic. In *ACM SIGCOMM*, pages 137–148, 2003.

[10] Min Fang, Narayanan Shivakumar, Hector Garcia-Molina, Rajeev Motwani, and Jeffrey D. Ullman. Computing iceberg queries efficiently. In *Proc. of VLDB*, pages 299–310, NY, NY, August 24-27 1998.

[11] Sudipto Guha. Space Efficiency in Synopsis Construction Algorithms. In *Proc. of VLDB*, pages 409–420, 2005.

[12] Sudipto Guha and Boulos Harb. Approximation algorithms for wavelet transform coding of data streams. In *Proc. of SODA*, pages 273–279, 2006.

[13] Panagiotis Karras and Nikos Mamoulis. The Haar+ Tree: a Refined Synopsis Data Structure. In *Proc. of the IEEE 23rd ICDE*, April 2007.

[14] Daniel Kifer, Shai Ben-David, and Johannes Gehrke. Detecting changes in data streams. In *Proc. of VLDB*, 2004.

[15] Jon Kleinberg. Bursty and hierarchical structure in streams. In *Proc. of the 8th ACM SIGKDD*, 2002.

[16] Laks V. S. Lakshmanan, Raymond T. Ng, Christine Xing Wang, Xiaodong Zhou, and Theodore Johnson. The generalized mdl approach for summarization. In *VLDB*, pages 766–777, 2002.

[17] Yossi Matias, J.S. Vitter, and M. Wang. Wavelet-Based Histograms for Selectivity Estimation. In *Proc. of ACM SIGMOD'98*, 1998.

[18] S. Muthukrishnan. Subquadratic algorithms for workload-aware Haar wavelet synopses. In *Proc. of FSTTCS*, 2005.

[19] P.J.Harrison. Exponential smoothing and short-term sales forecasting. *Management Science*, 13(11):821–842, 1967.

[20] Frederick Reiss, Minos Garafalakis, and Joseph Hellerstein. Compact Histograms for Hierarchical Identifiers. In *Proc. of VLDB*, 2006.

[21] Sunita Sarawagi. Explaining differences in multidimensional aggregates. In *The VLDB Journal*, pages 42–53, 1999.

[22] Sunita Sarawagi, Rakesh Agrawal, and Nimrod Megiddo. Discovery-driven exploration of olap data cubes. In *Proc. of EDBT*, pages 168–182, March 1998.

[23] Sudarshan S.Chawathe. Differencing data streams. In *Proc. of the Database Engineering and Applications Symposium (IDEAS)*, Montreal,Canada, July 2005.

[24] S.Hill, D.Agarwal, R.Bell, and C.Volinsky. Building an effective representation for dynamic graphs. *Journal of Computational and Graphical Statistics*, 15:584–608, 2006.

[25] WorldCup 1998. `http://ita.ee.lbl.gov/html/contrib/WorldCup.html`.

[26] Yin Zhang, Sumeet Singh, Subhabrata Sen, Nick Duffield, and Carsten Lund. Online identification of hierarchical heavy hitters: Algorithms, evaluation and applications. In *Proc. of ACM IMC'04*, October 2004.

[27] Yunyue Zhu and Dennis Shasha. Efficient elastic burst detection in data streams. In *Proc. of ACM SIGKDD'03*, pages 336–345, New York, NY, 2003.