Noname manuscript No.
(will be inserted by the editor)

Using Coarse Information for Real Valued Prediction

Amit Dhurandhar

Received: 13 July 2011 / Accepted: 2 August 2012

Abstract In domains such as consumer products and manufacturing amongst others, we have problems that warrant the prediction of a continuous target. Besides the usual set of explanatory attributes, we may also have exact (or approximate) estimates of *aggregated targets*, which are the sums of disjoint sets of individual targets that we are trying to predict. The question now becomes can we use these aggregated targets, which are a coarser piece of information, to improve the quality of predictions of the individual targets? In this paper, we provide a simple yet provable way of accomplishing this. In particular, given predictions from any regression model of the target on the test data, we elucidate a provable method for improving these predictions in terms of mean squared error, given exact (or accurate enough) information of the aggregated targets. These estimates of the aggregated targets may be readily available or obtained – through multilevel regression – at different levels of granularity. Based on the proof of our method we suggest a criterion for choosing the appropriate level. Moreover, in addition to estimates of the aggregated targets, if we have exact (or approximate) estimates of the mean and variance of the target distribution, then based on our general strategy we provide an optimal way of incorporating this information so as to further improve the quality of predictions of the individual targets. We then validate the results and our claims by conducting experiments on synthetic and real industrial data obtained from diverse domains.

Keywords Regression, Hierarchical, Coarse-to-fine

1 Introduction

In many industries such as consumer products and manufacturing, we have a supply chain consisting of a manufacturer who produces and sends goods to various distribution centers (DC), who further redistribute the goods amongst the stores. In this supply chain, we observe a delay between the time that the goods are produced and the stores

Amit Dhurandhar
IBM T.J. Watson
E-mail: adhuran@us.ibm.com

receiving the goods. This delay also corresponds to information being available at different levels of granularity. What we mean by this is that, initially, the manufacturer produces a certain amount of goods in bulk which he knows about but doesn't know exactly how these goods will be distributed amongst the various DCs until he receives the corresponding orders. Once the orders are received, the manufacturer knows how much to send each DC but doesn't know how these goods will be distributed amongst the various stores. Once the stores place their orders, this last piece of information is finally known. Thus, as time goes by, finer and finer pieces of information become available. From a strategic point of view, however, the manufacturer may want to know initially how his goods are going to be distributed among the various DCs and stores with as much accuracy as possible. Based on his past experience, he may be able to come up with predictions of how much each store or DC might order. However, the question we ask is the following: Is it at all possible to improve the quality of these predictions if we know the total amount that will be distributed for the current time period? In this paper, we answer this question affirmatively. That is, we provide a simple method to provably improve predictions obtained based on past observations by using estimates or actual values of the target at a coarser level of granularity for the current time period.

It is easy to see that if we have the true values (or accurate estimates) at a particular level of granularity, we can sum them up to get estimates at a coarser level of granularity. For example, if we have predictions for a coarser level of granularity and true values at a finer level, we can improve the predictive accuracy at the coarser level by aggregating the finer estimates and using them as predictions. This is seen in Fig. 1a. However, if we have the converse problem then the solution is not obvious. What we mean by this is that, if we knew coarser values and were trying to improve the quality of our predictions at a finer level, it's not clear if there is in fact a provable way of improving the accuracy. In other words, given predictions of the target for the current time period based on past experience – this could be the output of a regression model or something else – we improve (more precisely never worsen) the quality of these predictions using *aggregated target* information, i.e. using information about the sums of different sets of targets we are trying to predict. This will become clearer if we consider Fig. 1b, where we have predictions for the three datapoints (denoted by circles). The sum of the true targets is 9 and the method we suggest in this paper uses this value 9 to improve the accuracy of the predictions. In fact, as you will see, even if the value 9 is not the exact sum of the targets but an "accurate enough" estimate, our method still guarantees that the new predictions obtained by its application will be no less accurate than the old predictions.

Such a method can be used not only for supply chain problems but in any problem where good quality aggregate information is available, and we want to predict at a finer level of granularity. A good example is census data where information at a national or state level may be more reliable than data at a city or county level since there might be missing data as some people may not turn in the survey. Predicting the missing values can be done more effectively knowing the aggregate information. If aggregate information is available at multiple levels of granularity with varying accuracy, choosing the right level of granularity so as to maximize the improvement in predictions is not obvious. In this paper, we provide a criterion for choosing this optimal level of granularity. Moreover, using this criterion we also illustrate a way of choosing multiple levels that can potentially assist in further enhancing predictive accuracy.

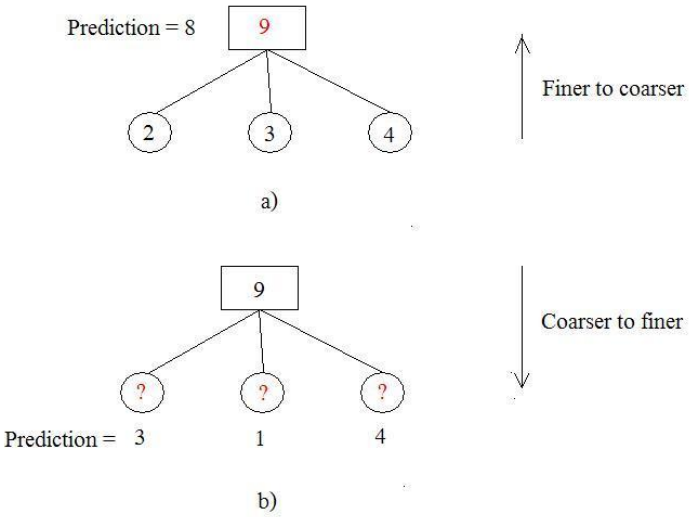

Fig. 1 a) Using finer estimates to improve coarser predictions. b) Using coarser estimates to improve finer predictions.

Information at a coarser level of granularity may not always be available as is the case in standard machine learning settings. In this case, we can build regression models on the historical data by aggregating it at various levels of granularity and use the "best" model to give us estimates of the aggregated targets at that level of granularity. These estimates can then be used in conjunction with our method to improve the predictions at the finest level of granularity that we care about. The "best" model is not necessarily the most accurate model amongst those built at the various coarser levels of granularity since, as we will see later, the amount of improvement in predictive accuracy at the finest level by using our method is a function of both the accuracy of the models and the level of granularity that they are built at. Consequently, we provide an algorithm for choosing the model that is most likely to elevate the accuracy of the predictions. In the trivial case, the best model might be the model at the finest level, which would suggest that aggregating the data isn't too helpful. An algorithm of this nature however, can be used for a wide variety of machine learning tasks such as predicting time series data where the aggregate models would predict the potential drift, if any, over time. This drift, if accurately captured, can assist in improving individual predictions [7, 1, 2]. Another example is microarray data which is sparse and hence aggregating it can help predictive accuracy [9]. Moreover, when aggregate information is not available for fixed sets of datapoints and the data has no known groupings (viz. underlying community structure) or order (viz. time series data), one has to decide which sets of points to aggregate to get the best results. Interestingly, the solution to this problem depends on how we obtain the aggregated targets. The strategy differs if the aggregate information is obtained by just summing

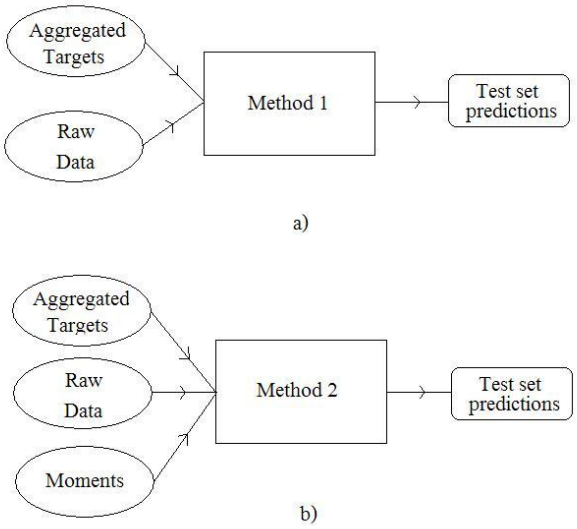

Fig. 2 The basic method in this paper represented in a) takes as input the raw data and the aggregated targets (exact or estimates) at a single or multiple levels and outputs the test predictions. The enhanced method represented in b) takes in addition the moment information of the target distrbution to give further improved predictions.

up the available data as opposed to obtaining it from an independent source, say through querying. We suggest methods for both these cases.

In addition to this, we enhance our method of using aggregated targets to improve finer predictions to be able to use distribution information of the target if available. In particular, we find the optimal weighting based on the mean and variance of the distribution of the target that will maximize the impact on the quality of the predictions in expectation. If this information is not available one may estimate these moments from the data, if it seems appropriate. The input-output of the original method and the enhanced method are pictorially depicted in Fig. 2.

The rest of the paper is organized as follows: In the next section we discuss relevant literature. In Section 3, we first describe our method. We then formally state this through lemmas and theorems (proofs in appendix) that show that our method in fact works. We then show that the predictions can be further improved if the mean and variance of the target are accurately known. Based on the proofs of these previous results, in cases where we may have estimates at multiple (coarser) levels of granularity, we provide a criterion that chooses a level that using our method is most likely to maximize the improvement in the quality of predictions at the finer level of granularity. In traditional settings where these estimates may not be available apriori, we suggest an algorithm where we build regression models at multiple levels of granularity on the training set in order to obtain the corresponding estimates. We then using our criterion, decide on the level and hence the regression model to be used to improve the quality of predictions obtained from a regression model built at the (finer) level of granularity

that we care about. We also suggest a way of using multiple levels and methods to decide which datapoints to group together when neither aggregate information for fixed sets of datapoints nor any groupings or order in the data is known. In Section 4, we present results of experiments performed on synthetic and real datasets and empirically validate the efficacy of our methods. In Section 5, we discuss further extensions and summarize the major findings in this paper.

2 Related Work

Using coarser information to improve predictions has been of some interest lately [11, 8, 13, 4, 15, 6]. In [11, 8, 4, 15] the authors employ this philosophy to improve performance of models in certain computer vision tasks (viz. pose estimation, face recognition, etc.). In [13] however, the idea is used to improve the performance of NLP models. The primary difference between this past literature and our work is that the results in this paper are for the regression setting while the past literature mainly considers the classification and the structured prediction setting. In the structured prediction setting, tasks included predicting a categorical attribute vector as opposed to a real valued one. This is equivalent to multidimensional classification, while we are focused on predicting real values. Moreover, in this paper, we explicitly provide a provable method to improve predictions at a finer level of granularity using aggregate information.

With regards to the regression setting, there has been extensive work in statistics related to multilevel modeling [10, 12]. The work most relevant to us is the literature on hierarchical linear models (HLMs). HLMs are used to model nested hierarchies explicitly known in the data. For example, educational research data often consists of pupils nested within classrooms which in turn are nested within schools and so on. In such situations, HLMs have been shown to be superior to other standard regression methods viz. simple linear regression, multiple linear regression, etc. A possible reason for their superior performance is that these models effectively capture the interdependencies between the different levels. There are extensions of these models such as generalized hierarchical linear models (GHLMs) which capture more complex non-linear dependencies between the various levels and the target. These non-linear extensions however, are usually significantly more computationally expensive to learn than the already expensive to learn HLMs [10].

The question now is what are the benefits gained using our method compared to this past work. First, HLMs estimate only a linear model at each stage. Extensions of this approach such as GHLM also have restrictions on the class of models used at each level. No such restriction exists in our case, as the proposed method is a meta-learning method applicable to any base regression method. Second, HLMs and its extensions model only nested hierarchies. In our approach, however, this constraint does not necessarily have to be enforced. For example, our method permits hierarchies of the form shown in Fig. 3. Here level 2 and level 3 are definitely not nested. Hence, we can model more general hierarchies in our framework. Third, our method is robust to errors in estimates at coarser levels as is shown in the lemmas and theorems proved later. In contrast, HLM's are sensitive to small group sizes even if we have exact estimates at higher levels [10].

Other notable work related to using aggregate information in the regression setting is with respect to the problem of identifying socioeconomic factors that may lead to hospital admissions for heart and circulatory diseases [6]. The proposed model in this

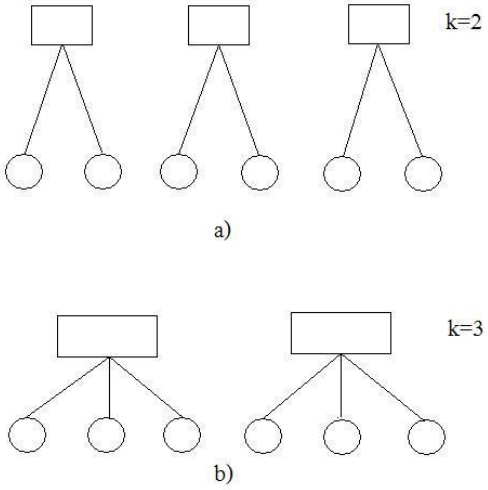

Fig. 3 a) Aggregation granularity is 2. b) Aggregation granularity 3.

work is based on specific aspects of the domain to which it is applied and is not easily generalizable. Our method, however, is not necessarily restricted to any particular domain. In fact, as we will see in the experimental section, it is applicable in multiple diverse domains.

3 Trickling Down Aggregates

In this section we first describe a simple method to trickle down aggregate information in order to improve finer predictions. We formally state the relevant results with proofs provided in the appendix. If estimates of the aggregated target are available at multiple (coarser) levels of granularity, we then – based on our proofs for the previous results – suggest a criterion to choose the level at which using our method is most likely to maximize the improvement in the quality of predictions at the (finest) level of granularity that we want to predict. If these estimates are not available, we suggest an algorithm for obtaining them and using the criterion to decide the appropriate level.

Before we start formally describing our results we define a couple of terms.

Aggregation granularity: We define aggregation granularity k as the number of values at the finest level of granularity (i.e. at the level of the original dataset) summed together to form coarser estimates. For example, in Fig. 3a, the aggregation granularity is 2 since, if the circles represent datapoints at the finest level, then the rectangles which denote coarser estimates are sums of pairs of these circles. Similarly, in Fig. 3b, the aggregation granularity is 3 since the rectangles are sums of triplets of the circles.

More formally, given a sample $X = \{(x_1^1, x_1^2, ..., x_1^d), ..., (x_N^1, x_N^2, ..., x_N^d)\}$ of N datapoints in d dimensional space, we obtain a sample $A^d = \{(a_1^1, a_1^2, ..., a_1^d), ..., (a_T^1, a_T^2, ..., a_T^d)\}$ at aggregation granularity r i.e. $k = r$ by setting each $a_p^q = \sum_{i \in S_p} x_i^q$, where S_p contains indices of r datapoints in the original sample, with $S_i \cap S_j = \Phi \; \forall i, j \in \{1, ..., T\}$ and $i \neq j$. Moreover, $S_1 \cup S_2 \cup ... \cup S_T = \{1, ..., N\}$.

Aggregated targets: These are sums of the individual targets in each set, in which the sets form a partitioning of the individual targets in the dataset. Note that the rectangles in Fig. 3 would denote aggregated targets if the circles denoted individual target values. In the above formal definition of aggregation granularity, if we instantiate X to denote the target (hence, $d = 1$), then A^1 – which can be written as simply A – would denote the aggregated targets at aggregation granularity r.

3.1 Method and Results

An informal description of our method where the aggregation granularity is k is as follows: We first sum up the various disjoint sets of k predictions corresponding to the aggregated targets which are already available or obtained by techniques described before. With this we have each of the aggregated targets associated with its own sum of k predictions. We now subtract each of these sums from the corresponding aggregated targets which gives us the corresponding differences. We then divide each of these differences by k and uniformly add them to the corresponding k predictions. For instance, if linear regression is used to obtain predictions for N datapoints, i.e., $\bar{Y} = [\bar{y_1}, \cdots, \bar{y_N}]^\top = \begin{bmatrix} x_1^1 & \cdots & x_1^d \\ \vdots & \dots & \vdots \\ x_N^1 & \cdots & x_N^d \end{bmatrix} [\beta_1, \cdots, \beta_d]^\top$, where $A = \{a_1, \cdots, a_{\frac{N}{k}}\}$[1] are the aggregated targets, then the new modified predictions are given by, $\hat{Y} = [\hat{y_1}, \cdots, \hat{y_N}]^\top = [\bar{y_1}, \cdots, \bar{y_N}]^\top + \frac{1}{k}[\epsilon_{\lceil \frac{1}{k} \rceil}, \cdots, \epsilon_{\lceil \frac{N}{k} \rceil}]^\top$. Here $\epsilon_i = a_i - \sum_{j=(i-1)k+1}^{ik} \bar{y_j}$, where $i \in \{1, \cdots, \frac{N}{k}\}$.

Thus in Fig. 1b, where 9 is the aggregated target with 3, 1 and 4 being the predictions, we would first add 3, 1 and 4 which gives us 8, then subtract 8 from 9 which gives us 1 and then finally add $\frac{1}{3}$ to the original predictions which would give us $\frac{10}{3}$, $\frac{4}{3}$ and $\frac{13}{3}$ as the new predictions. If additional information regarding the distribution (mean and variance) of the target is available, then rather than distributing the differences uniformly amongst the predictions an optimal convex weighting scheme is derived.

With this, we now present five results which include a formal description of the method we described above:

- In lemma 1 we show that knowing the *true or exact* values of the aggregated targets and predictions of the individual targets, our method can produce new modified predictions that are never worse in terms of mean squared error (MSE) than the original predictions.
- In theorem 1 we show that, knowing *approximate* values (within a certain error bound) of the aggregated targets and predictions of the individual targets, our method can produce new modified predictions that are never worse in terms of MSE than the original predictions.

[1] Here we assume N is divisible by k to avoid clutter and drive home the basic point.

- In lemma 2 we show that the approach of uniformly distributing the differences stated in lemma 1 not only enhances the predictive accuracy but is also the optimal way of distributing the differences.
- Theorem 2 shows that, even if we know the exact values of the aggregated targets and have predictions of the individual targets and if we alter our method slightly where we distribute the differences non-uniformly amongst the predictions, then the claim made in lemma 1 no longer holds. In other words, the MSE of the new predictions might be greater than the old predictions if all the differences are not distributed uniformly. This theorem shows that there is *gap* between the optimal solution stated in lemma 1 (and proved in lemma 2) and the other solutions since not only are the other solutions inferior to the one in lemma 1, they are also inferior to using the old predictions.
- Lastly, in lemma 3 we show that, in addition to knowing the aggregated targets and having predictions of the individual targets, if we also know the mean and variance of the target distribution then we can derive optimal weights for distributing the differences which may be non-uniform.

Lemma 1 *Consider two sets of N real numbers $Y = \{y_1, y_2, ..., y_N\}$ and $\bar{Y} = \{\bar{y_1}, \bar{y_2}, ..., \bar{y_N}\}$ (estimates). Let $A = \{a_1, ..., a_m\}$ and $\bar{A} = \{\bar{a_1}, ..., \bar{a_m}\}$ such that if, k is the aggregation granularity, $l_i = min(ik, N) - (i-1)k$, $m = \lceil \frac{N}{k} \rceil$, then $a_i = \sum_{j=(i-1)k+1}^{min(ik,N)} y_j$ and $\bar{a_i} = \sum_{j=(i-1)k+1}^{min(ik,N)} \bar{y_j}$. If $\epsilon_i = a_i - \bar{a_i}$ then,*

$$\sum_{j=1}^{N} (y_j - \bar{y_j})^2 \geq \sum_{j=1}^{N} (y_j - \hat{y_j})^2$$

where $\hat{y_j} = \bar{y_j} + \frac{\epsilon_{\lceil \frac{j}{k} \rceil}}{l_{\lceil \frac{j}{k} \rceil}}$

The result below shows that even if the values at the coarser level of granularity are not known exactly but with "some" error, they still can be used to enhance accuracy.

Theorem 1 *Consider two sets of N real numbers $Y = \{y_1, y_2, ..., y_N\}$ and $\bar{Y} = \{\bar{y_1}, \bar{y_2}, ..., \bar{y_N}\}$ (estimates). Let $A = \{a_1, ..., a_m\}$ and $\bar{A} = \{\bar{a_1}, ..., \bar{a_m}\}$ where if k is the aggregation granularity, then $l_i = min(ik, N) - (i-1)k$, $m = \lceil \frac{N}{k} \rceil$, $a_i = \sum_{j=(i-1)k+1}^{min(ik,N)} y_j$ and $\bar{a_i} = \sum_{j=(i-1)k+1}^{min(ik,N)} \bar{y_j}$. If $\epsilon_i = a_i - \bar{a_i}$ and $\delta_i \in [0, 2\epsilon_i]$ then,*

$$\sum_{j=1}^{N} (y_j - \bar{y_j})^2 \geq \sum_{j=1}^{N} (y_j - \hat{y_j})^2$$

where $\hat{y_j} = \bar{y_j} + \frac{\delta_{\lceil \frac{j}{k} \rceil}}{l_{\lceil \frac{j}{k} \rceil}}$

Lemma 2 *Consider two sets of N real numbers $Y = \{y_1, y_2, ..., y_N\}$ and $\bar{Y} = \{\bar{y_1}, \bar{y_2}, ..., \bar{y_N}\}$ (estimates). Let $A = \{a_1, ..., a_m\}$ and $\bar{A} = \{\bar{a_1}, ..., \bar{a_m}\}$ where if k is the aggregation granularity, $l_i = min(ik, N) - (i-1)k$, $m = \lceil \frac{N}{k} \rceil$, then $a_i = \sum_{j=(i-1)k+1}^{min(ik,N)} y_j$ and $\bar{a_i} = \sum_{j=(i-1)k+1}^{min(ik,N)} \bar{y_j}$. If $\epsilon_i = a_i - \bar{a_i}$ and $\forall i \sum_{j=(i-1)k+1}^{min(ik,N)} \alpha_j = 1$ where $\forall j \ \alpha_j \geq 0$ then,*

$$\sum_{j=1}^{N}(y_j - \tilde{y_j})^2 \leq \sum_{j=1}^{N}(y_j - \hat{y_j})^2$$

where $\tilde{y_j} = \bar{y_j} + \frac{\epsilon_{\lceil \frac{j}{k} \rceil}}{l_{\lceil \frac{j}{k} \rceil}}$ *and* $\hat{y_j} = \bar{y_j} + \alpha_j \epsilon_{\lceil \frac{j}{k} \rceil}$.

Theorem 2 *Consider two sets of* N *real numbers* $Y = \{y_1, y_2, ..., y_N\}$ *and* $\bar{Y} = \{\bar{y_1}, \bar{y_2}, ..., \bar{y_N}\}$ *(estimates). Let* $A = \{a_1, ..., a_m\}$ *and* $\bar{A} = \{\bar{a_1}, ..., \bar{a_m}\}$ *where if* k *is the aggregation granularity,* $l_i = min(ik, N) - (i-1)k$, $m = \lceil \frac{N}{k} \rceil$, *then* $a_i = \sum_{j=(i-1)k+1}^{min(ik,N)} y_j$ *and* $\bar{a_i} = \sum_{j=(i-1)k+1}^{min(ik,N)} \bar{y_j}$. *If* $\epsilon_i = a_i - \bar{a_i}$ *and* $\forall i \sum_{j=(i-1)k+1}^{min(ik,N)} \alpha_j = 1$ *where* $\forall j \ \alpha_j \geq 0$ *with all* α_j *(for any* i*) not being equal then there always exists a* Y *and* $\bar{Y}$ *such that,*

$$\sum_{j=1}^{N}(y_j - \bar{y_j})^2 \leq \sum_{j=1}^{N}(y_j - \hat{y_j})^2$$

where $\hat{y_j} = \bar{y_j} + \alpha_j \epsilon_{\lceil \frac{j}{k} \rceil}$

Lemma 3 *Consider two sets of* N *real numbers* $Y = \{y_1, y_2, ..., y_N\}$ *and* $\bar{Y} = \{\bar{y_1}, \bar{y_2}, ..., \bar{y_N}\}$ *(estimates). Let* $A = \{a_1, ..., a_m\}$ *and* $\bar{A} = \{\bar{a_1}, ..., \bar{a_m}\}$ *where if* k *is the aggregation granularity,* $l_i = min(ik, N) - (i-1)k$, $m = \lceil \frac{N}{k} \rceil$, *then* $a_i = \sum_{j=(i-1)k+1}^{min(ik,N)} y_j$ *and* $\bar{a_i} = \sum_{j=(i-1)k+1}^{min(ik,N)} \bar{y_j}$. *If* $\epsilon_i = a_i - \bar{a_i}$ *and it is known that* $Y \sim D$ *where* μ *is the mean of the distribution* D *(i.e.* $E[Y]$*) and* σ^2 *is the variance then,*

$$E[\sum_{j=1}^{N}(y_j - \bar{y_j})^2] \geq E[\sum_{j=1}^{N}(y_j - \hat{y_j})^2] \tag{1}$$

where $\hat{y_j} = \bar{y_j} + \alpha_{\lceil \frac{j}{k} \rceil}^{(j \mod l_{\lceil \frac{j}{k} \rceil} + 1)} \epsilon_{\lceil \frac{j}{k} \rceil}$, $\alpha_{\lceil \frac{j}{k} \rceil}^{(j \mod l_{\lceil \frac{j}{k} \rceil} + 1)} \geq 0$ *and* $\sum_{i=1}^{l_p} \alpha_p^{(i)} = 1 \ \forall p \in \{1, ..., \lceil \frac{N}{k} \rceil\}$. *The optimal alphas that minimize the expectation on the right side of the inequality in equation 1 are given by,*

$$\alpha_p^{(i)} = \frac{1}{l_p \epsilon_p^2}[l_p \epsilon_p (\mu - \bar{y}_{i+k(p-1)}) - (2l_p - 1)(\sigma^2 + \mu^2)]; i \neq l_p$$
$$\alpha_p^{(l_p)} = \frac{1}{l_p \epsilon_p^2}[(2l_p - 1)(l_p - 1)(\sigma^2 + \mu^2) - l_p \epsilon_p ((l_p - 1)\mu + \bar{y}_{i+k(p-1)} - a_p)]$$

A high level description of the methods formally described in lemma 1 and in lemma 3 are shown in Fig. 4.

3.2 Choosing *between* Multiple levels

Based on the proofs (in the appendix) of the results in the previous subsection, we observe that the reduction in MSE by applying our method is a function of the aggregation granularity and the accuracy of the estimates of the aggregated targets. In particular, the smaller the aggregation granularity and the lower the error, the more

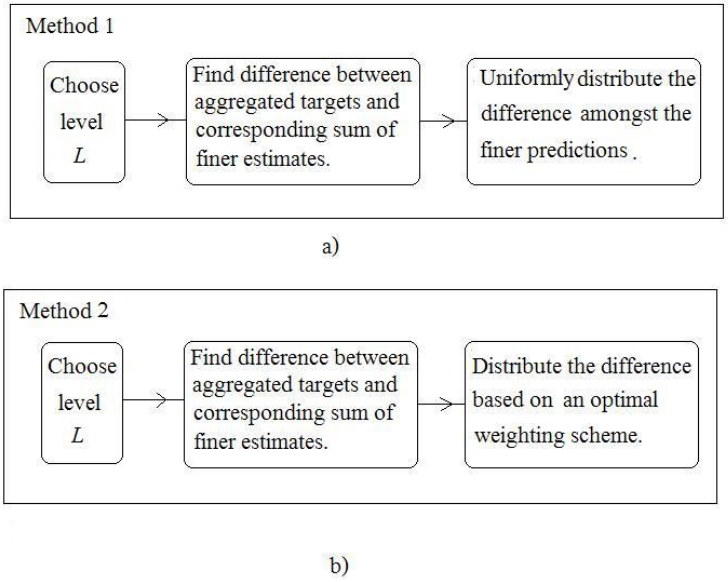

Fig. 4 The two methods represented in Fig. 2 are illustrated above. The first one evenly distributes the differences while the second one distributes the differences based on a convex weighting scheme.

significant the improvement. However, if we have estimates of the aggregated targets at multiple levels of granularity with varying accuracy, it is generally not clear as to which level will lead to the most improvement. For example, at $k = 2$ we might have an error of 0.2 and at $k = 3$ we might have an error of 0.15. In this case, it is not clear whether to use the estimates of the aggregated target at level 2 or level 3. Note that if the error at level 3 was more than that at level 2 then the choice is obvious and we would choose level 2. Hence, we see that in choosing the appropriate level there is a trade-off between the aggregation granularity and the error of the estimates of the aggregated targets.

Criterion: If k is the aggregation granularity and MSE_k denotes the mean squared error of the aggregated targets at aggregation granularity k then the level that is most likely to lead to maximum improvement in the predictions of the target is given by:

$$L = \min_k \operatorname*{argmin}_k k\mathrm{MSE}_k \tag{2}$$

Hence, if there are multiple k values with the same value of the objective, we choose the minimum k. If $L = 1$ is the answer, that means that the aggregated targets at coarser levels will most likely not help in improving the predictive accuracy.

One may ask what the thought process is in using the estimates of the aggregated targets if one is able to compute the MSE at that level which implies one knows true values of the aggregated targets. The point of this exercise is to come up with a criterion that in a traditional machine learning setting can be used to choose a regression

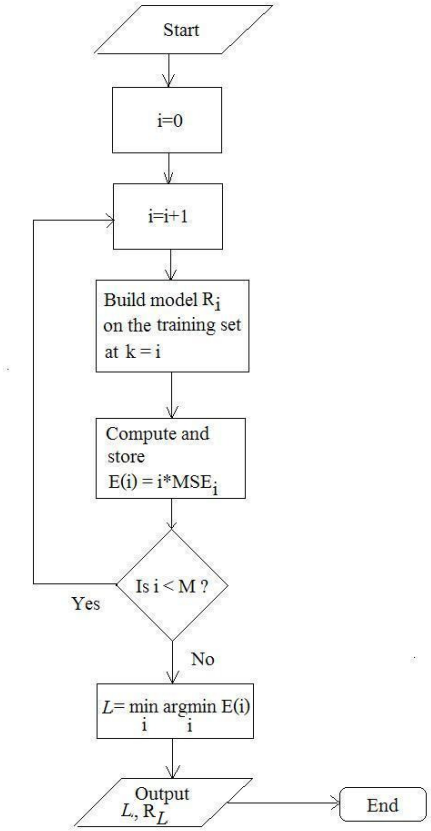

Fig. 5 Algorithm for choosing the appropriate level and the corresponding regression function given that models are built at M levels i.e from $k = 1$ to $k = M$.

model at a certain level of aggregation based on the training set followed by using this model to estimate aggregated targets on the test set, which then can be used to improve predictions of the target on the test set. An algorithm for the same is described below:

Algorithm for choosing level/model in traditional settings: In a standard machine learning setting where we have a training and a test set, we can train M models – one for each of the M levels of aggregation – on the training set and use the criterion mentioned in equation 2 to decide the best level and the corresponding model to be used to improve the predictions on the test set (i.e. of a model built at $k = 1$). If the test set size is N, potentially M could be N. However it makes little sense to build models beyond a certain level for mainly two reasons: 1) the corresponding dataset sizes (due to aggregation) at or beyond that level or aggregation granularity may be insufficient to train a model and 2) beyond a certain aggregation granularity even if

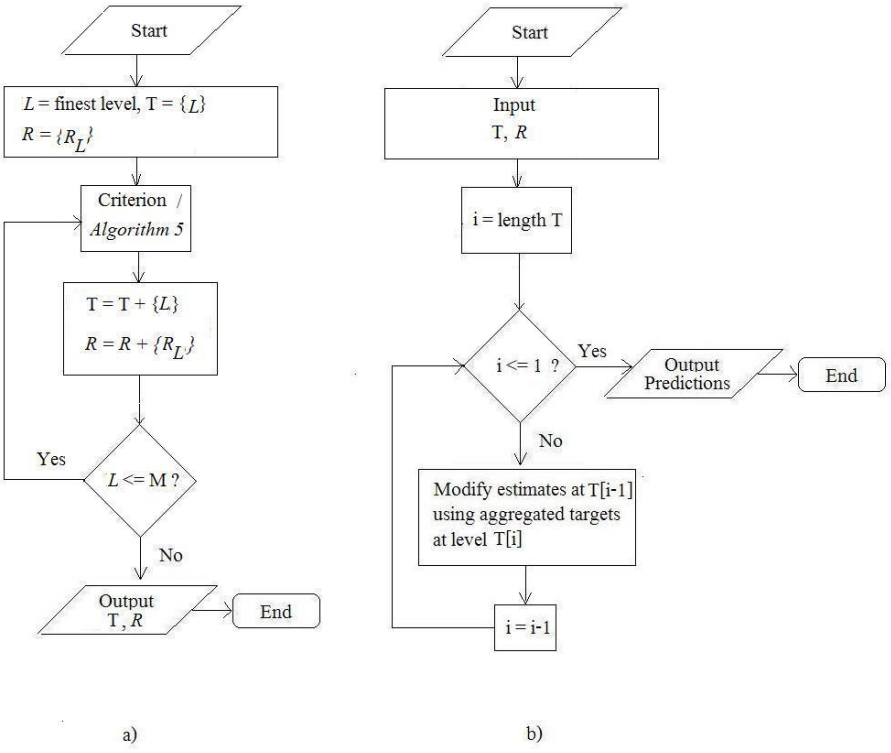

Fig. 6 The text in italics (viz. R) denotes the additional inputs/outputs/processing that is needed when the aggregated targets are not available but have to be estimated from the data. a) illustrates the learning phase to choose multiple levels, while b) illustrates the testing/application phase after the learning has been completed.

we knew the exact values of the aggregated targets at those levels, the enhancement they would produce in the quality of predictions is minuscule.

A flowchart describing the algorithm to decide the level and model is given in Fig. 5. As per the flowchart, we end up using the model R_L to predict aggregated targets on the test set. The estimates produced by R_L can then be used to improve the predictions of the model built to predict the target by using the strategy we outlined before. Note that the strategy changes depending on whether, in addition to these estimates, we also use the estimates (or actuals if available) of the moments of the target distribution.

3.3 Choosing Multiple levels

In the previous section we stated a criterion to choose a level that will most likely lead to maximum improvement in the quality of predictions. A natural question is can we choose multiple levels to further boost the predictive accuracy? The answer to this question depends on the quality of the estimates of the aggregated targets. If we have exact estimates, then using any level other than the lowest for which we know aggregated targets seems superfluous. The reason for this is, when we have exact estimates, the information available at lower aggregation granularities is never less than that at higher aggregation granularities. Hence, in this case, using the lowest available aggregated target information seems to be the best choice. Notice that our criterion for selecting the appropriate level would also result in the same choice.

When the estimates for the aggregated targets are not exact, however, the above argument is not applicable. In this case, one may potentially use multiple levels to improve predictions. The manner in which this might be accomplished is shown in Fig. 6. The idea is to recursively apply either the level selection criterion directly or algorithm 5 depending on if we already have the aggregated target information or if we have to estimate it. For example, we first choose the best level, say L_1, for the finest/lowest aggregation granularity, say L_0. We then find the best level, say L_2, for level L_1. We continue doing this until we reach level M – the highest/coarsest level that we wish to consider. We keep track of all the levels we choose $T = \{L_0, L_1, L_2, ..., L_p, M\}$. We then use the aggregated targets at level M to improve estimates at level L_p. This is followed by using the improved estimates at level L_p to improve the estimates at the previous level recorded in the list T. This process continues until the improved estimates at L_1 are used to improve our predictions at the finest level.

The greedy approach described above can thus be used to choose multiple levels that can potentially further enhance predictive accuracy.

3.4 Choosing Datapoints to Aggregate

In the previous sections we assumed that – when aggregated targets are available – they are for fixed sets of datapoints, and we have no control over the formation of these groups/sets. Though this may be true in many real applications, it's still worth exploring how we would form the sets if we were able to query their aggregated targets given a large but constant number of total allowed queries. The other more realistic problem is in the standard machine learning setting, where we do not have aggregated targets and the data has no explicit order (viz. time series) or groupings. In this case, it is not clear how to group the data in order to build models that estimate the aggregated targets.

We now discuss these two scenarios and, given our methods, argue that the best solutions to aggregate in these scenarios are, interestingly, diametric.

3.4.1 Aggregates from Independent Source

Consider the situation where we could query the aggregated targets for groups of datapoints of our choice. Given that our base method – the one without moment information – uniformly distributes the differences between the aggregated targets and sums of sets of individual predictions, we would want to group together points on

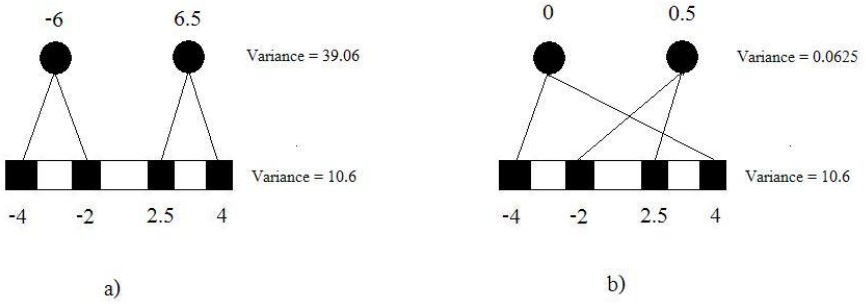

Fig. 7 a) shows the variance if we aggregate based on standard metrics i.e. the closest set of points. b) shows the variance if we aggregate based on reciprocals of the standard metrics.

which we suspect that the deployed regression method would give, if not the same, at least similar errors. A good bet for this to happen would probably be if we somehow cluster datapoints based on their similarity and later query their aggregated targets. The intuition behind this approach is that any reasonable regression method would probably predict similar values for similar datapoints which in turn would imply that the errors it makes on them would also be similar.

The clustering algorithm that we would want to use in such a scenario would probably be an agglomerative hierarchical clustering approach [14, 5]. Such an approach would form aggregates of increasing size as we go up the hierarchy, with the sizes of the different aggregates within each level in the hierarchy being almost the same[2]. One could then query the aggregated targets based on the obtained clusterings.

3.4.2 Aggregates from Input Data

Consider the standard machine learning setting where we do not have access to the aggregated targets nor does the data have any explicit order or grouping. If the data were ordered or had groupings, one could build models using the algorithms in Fig. 5 or Fig. 6 by aggregating successive datapoints or datapoints in groups respectively. However, without these pieces of information we need to form our own groups to aggregate.

As in the previous case, an agglomerative hierarchical clustering approach would be suitable here too. Though at a high level the approach would be the same as before, the details such as the nature of the distance metric used would be different. In the previous case, we would use standard distance metrics such as Euclidean distance, Manhattan distance, Mahalanobis distance, etc. In this case however, we would want to use a metric that is *inversely proportional* to these standard distance metrics. For example, we may use the reciprocal of the euclidean distance or the reciprocal of the manhattan or mahalanobis distance. Hence, if our dataset has N datapoints $X = \{x_1, \cdots, x_N\}$ we would compute their pairwise distances and form disjoint groups of the farthest pairs of points $x_{ij} = (x_i, x_j)$ by sorting and forming groups without replacement. Thus level 2, X^2, would have $\frac{N}{2}$ points and would be formed by summing the respective pairs. The

[2] The clusters within every level may be of slightly different sizes but we have accounted for that in our methods and analysis by denoting the size of each partition p by l_p rather than a constant k.

next level would be formed by replacing X with X^2 and repeating this aforementioned procedure. If the N datapoints form the training set, we would learn a regression model at each level and choose the appropriate levels based on algorithm 6a. On the other hand, if the N datapoints form the test set we would use algorithm 6b.

The reason for using reciprocals of standard metrics is as follows: In the previous case, since we obtained the (estimates of the) aggregated targets from a source independent of our data, the errors in the predictions of the regression method would be independent of the errors in the estimates of the aggregated targets. In this case however, these two errors would be correlated since we are using the same data source and maybe even the same regression method at both levels. The correlation would be higher if we aggregated similar datapoints and then built regression models to obtain estimates of the aggregated targets. One of the main reasons for aggregating data is to reduce the variance with the hope that the reduction in variance will more than offset the reduction in the dataset size and will assist in building regression models that are more accurate than the regression models at the finest level. However, if we group together similar datapoints, the reduction in variance as we aggregate would be minimal. For example, as shown in Fig. 7, if we aggregate as in b) the reduction in variance is significantly higher than if we aggregate as in a). Hence, in this case, it makes more sense to group together datapoints based on the reciprocals of the standard metrics.

4 Experiments

In this section, we evaluate our proposed solutions on synthetic data as well as on 4 real industrial datasets obtained from diverse domains. We report results for two state-of-the-art regression methods – which are the base regression methods – namely ridge regression and support vector machine (SVM) regression with RBF kernel. We use these methods to predict the target as well as the aggregated targets when they are unknown. Moreover, whenever the algorithms in Fig. 6 are used, we build models until M, which is equal to the test set size. In experiments on real data, we also compare our procedure with HLMs. HLMs broadly have two classes of parameters that have to be estimated; fixed factors and random factors. We estimate the fixed factors using least squares and the random factors using maximum likelihood. Whenever a nested hierarchy is unknown we create it until level M by grouping successive pairs of points to form level 2, followed by grouping successive pairs of groups at level 2 to form the next level and so on. An example two-level HLM is represented below:

$$Level - 1$$
$$y_{ij} = \beta_{0j} + \beta_{1j} x_{ij} + \epsilon_{ij}$$

$$Level - 2$$
$$\beta_{0j} = \gamma_{00} + \gamma_{01} W_j + \eta_{0j}$$
$$\beta_{1j} = \gamma_{10} + \gamma_{11} W_j + \eta_{1j}$$

where i's index the datapoints and j's the groups. The β are first-level fixed factors while the γ are second-level fixed factors. The ϵ and η are first and second-level random factors respectively. The first-level random factors are assumed to be distributed normally with mean zero and variance σ^2. On the other hand, the second-level random

$\sigma\downarrow,\rho\rightarrow$	0.8	0.6	0.4	0.2
0.8	0.21,0.16,0.18	0.15, 0.08,0.19	0.23, 0.11,0.27	0.44, 0.39,0.41
0.6	0.19,0.14,0.19	0.08,**0.03**,0.09	0.10,**0.03**,0.33	0.41,0.44,0.42
0.4	0.18,0.12,0.20	0.07,**0.01**,0.14	0.09,**0.02**,0.15	0.39,0.37,0.53
0.2	0.22,0.24,0.35	0.32,0.28,0.34	0.42,0.35,0.36	0.47,0.49,0.54

Table 1 For dataset of size *100* and using ridge regression we see above the $p-values$ of the paired t-test. The first entry in each cell is the $p-value$ for testing hypothesis G_0, the second entry is the $p-value$ for testing hypothesis H_0 and the last entry is the $p-value$ for testing the hypothesis T_0. The entries in bold are cases where the corresponding null hypotheses are rejected.

$\sigma\downarrow,\rho\rightarrow$	0.8	0.6	0.4	0.2
0.8	0.23,0.21,0.22	0.19,0.16,0.19	0.17,0.19,0.24	0.38,0.31,0.35
0.6	0.18,0.15,0.16	0.07,**0.02**,**0.02**	0.06,**0.02**,**0.03**	0.39,0.36,0.41
0.4	0.22,0.16, 0.22	**0.04**,**0.02**,**0.03**	0.08,**0.01**,**0.02**	0.33,0.36,0.57
0.2	0.42,0.40,0.37	0.15,0.10,0.33	0.27,0.28,0.44	0.53,0.49,0.58

Table 2 For dataset of size *1000* and using ridge regression we see above the $p-values$ of the paired t-test. The first entry in each cell is the $p-value$ for testing hypothesis G_0, the second entry is the $p-value$ for testing hypothesis H_0 and the last entry is the $p-value$ for testing the hypothesis T_0. The entries in bold are cases where the corresponding null hypotheses are rejected.

$\sigma\downarrow,\rho\rightarrow$	0.8	0.6	0.4	0.2
0.8	0.28,0.24,0.26	0.21,0.15,0.26	0.13,0.09,0.22	0.31,0.36,0.43
0.6	0.16,0.14,0.25	0.11,**0.03**,0.14	0.06,**0.03**,0.23	0.55,0.47,0.51
0.4	0.23,0.27,0.18	0.08,**0.02**,0.16	0.08,**0.02**,0.13	0.38,0.37,0.49
0.2	0.33,0.29,0.32	0.24,0.22,0.26	0.39,0.34,0.31	0.54,0.58,0.66

Table 3 For dataset of size *100* and using SVM we see above the $p-values$ of the paired t-test. The first entry in each cell is the $p-value$ for testing hypothesis G_0, the second entry is the $p-value$ for testing hypothesis H_0 and the last entry is the $p-value$ for testing the hypothesis T_0. The entries in bold are cases where the corresponding null hypotheses are rejected.

$\sigma\downarrow,\rho\rightarrow$	0.8	0.6	0.4	0.2
0.8	0.28,0.25,0.24	0.12,0.10,0.14	0.16,0.17,0.27	0.21,0.23,0.29
0.6	0.23,0.18,0.15	0.09,**0.02**,**0.02**	0.07,**0.03**,**0.02**	0.37,0.39,0.35
0.4	0.16,0.14, 0.23	0.06,**0.02**,**0.01**	0.05,**0.01**,**0.03**	0.36,0.30,0.45
0.2	0.36,0.36,0.38	0.34,0.31,0.33	0.35,0.34,0.42	0.51,0.45,0.56

Table 4 For dataset of size *1000* and using SVM we see above the $p-values$ of the paired t-test. The first entry in each cell is the $p-value$ for testing hypothesis G_0, the second entry is the $p-value$ for testing hypothesis H_0 and the last entry is the $p-value$ for testing the hypothesis T_0. The entries in bold are cases where the corresponding null hypotheses are rejected.

factors have a multivariate normal distribution again with mean zero. W is just an indicator matrix, which selects certain aggregated datapoints at level 2. This two-level model can easily be extended to three or more levels using the same basic idea as is described in [10].

4.1 Synthetic Data Experiments

Setup: We generate synthetic datasets from an 11-dimensional Gaussian distribution, of which 10 are explanatory attributes and the last one is the target. We set the mean of this Gaussian to be the origin while the correlation matrix takes different values so as to generate different types of datasets. The general form of this correlation matrix is fixed, however, in that all entries corresponding to correlation between two different explanatory attributes are set to ϕ, while the standard deviation of all of the 11 attributes is set to σ and the correlation between the explanatory attributes and target is set to ρ. In order to generate datasets with different amounts of correlation between the explanatory attributes and target, we vary ρ. To generate datasets with different variances of the individual attributes we vary σ. Moreover, to observe the behavior of our method on different dataset sizes, we create datasets of size 100 and 1000 for each ρ and σ combination that we consider. In particular, we create datasets for $\rho = \{0.8, 0.6, 0.4, 0.2\} \times \sigma = \{0.8, 0.6, 0.4, 0.2\}$ as seen in tables 1, 3 (dataset size = 100), 2 and 4 (dataset size = 1000). For each of the combinations of these parameters, we sample 100 datasets where in 25 of these datasets $\phi = 0$, in another 25 $\phi = 0.2$, then in another 25 $\phi = 0.4$ and in the remaining 25 $\phi = 0.6$.

Each dataset that is obtained, which is of a particular size and sampled with particular ϕ, ρ and σ, is randomly split into train (70%) and test sets (30%). This is done 50 times to generate 50 different train and test splits on the same dataset. On each training dataset we first learn a model using only ridge or SVM regression (R). We then learn using each of the regression methods and aggregated targets formed by randomly grouping together datapoints (R+G). This is followed by training each of the regression methods along with our algorithm in Fig. 6 (R+A). Finally we learn using each of the regression methods along with our algorithm and moment information (R+A+M) estimated from the training set which gives us an estimate of the optimal weights as per lemma 3. On the corresponding test sets we compute the MSE of each of these four techniques in predicting the target. Thus, for each combination of parameters we now get 50×100=5000 MSE values for each of the four techniques, once using ridge regression and then using SVM regression. The aggregates for R+A and R+A+M are formed using agglomerative hierarchical clustering as described in section 3.4.2.

Using these MSE values we primarily want to decipher the parameter settings when R is worse than R+G and/or R+A, and when R+A is worse than R+A+M. To arrive at these conclusions in a statistically sound manner, we use the paired t-test [3] to confirm that the reductions in MSE from R to R+G/R+A and R+A to R+A+M are statistically significant. Hence, we carry out three hypothesis tests: the first one to see if the differences in MSE between R and R+G are significant, the second one to see if the differences in MSE between R and R+A are significant and the third one to see if the differences in MSE between R+A and R+A+M are significant. The first two studies would also indicate the benefit, if any, of using our aggregation algorithm as opposed to stochastic aggregation. The null hypothesis in the first case is:

G_0 : The models R and R+G are equivalent.

The null hypothesis in the second case is:

H_0 : The models R and R+A are equivalent.

The null hypothesis in the third case is:

T_0 : The models R+A and R+A+M are equivalent.

Observations: We first discuss the results for hypothesis H_0, as understanding the results for G_0 is easier relative to H_0. From the results in tables 1, 2, 3 and 4, we see that the hypothesis H_0 is rejected (i.e. $p-value < 0.05$) when the correlation between explanatory attributes and the target is neither too high nor too low and when the standard deviation of the attributes is moderate. A possible reason for this to happen is as follows: At high correlations the explanatory attributes are excellent predictors of the target and, hence, a simple method is sufficient to be able to predict the target accurately. At low correlations the explanatory attributes contain very little information that can be used to predict the target irrespective of the modeling method used. At moderate correlations the explanatory attributes are reasonable predictors of the target. In this case, however, when the variance of the target is high, aggregating using algorithm in Fig. 6 does not reduce the variance sufficiently until we have aggregated by more than a few levels and, as mentioned before, the higher the aggregation granularity the lesser the improvement. When the variance of the target is low, further aggregating keeps the variance in the same regime, which is low, and hence the predictive power of the attributes does not change significantly at lower levels. At mid variances, however, aggregation causes the variances to fall in the low regime and hence, at even low levels of aggregation we predict accurately which then enhances the performance of the model predicting the target.

From the same four tables mentioned before, we see that G_0 is almost always rejected. However the $p-values$ are closer to being significant when the dataset size is larger. These observations imply two things. First, forming aggregated targets using our algorithm is definitely more useful than randomly forming aggregated targets. Second, the randomized groupings work better with larger datasets since there is a higher chance of uncorrelated points being grouped together.

From the results in tables 1, 2, 3 and 4, we see that the hypothesis T_0 is rejected when the dataset size is large and the other conditions are the same as that for hypothesis H_0 to be rejected. For R+A+M to work well it is required that R+A works at least reasonably since, if our estimates of the aggregated targets using algorithm in Fig. 6 are not accurate, then irrespective of how we distribute these estimates the prediction quality will not improve by much. When R+A works really well there isn't much room for improvement. Moreover, a larger dataset size aids in obtaining more accurate estimates of the moments. Hence, the scenarios where R+A+M shows improvement over R+A is a subset of the cases where R+A shows improvement over R.

4.2 Real Data Experiments

We now test the efficacy of our approach on 4 real industrial datasets obtained from varied domains. Since the records in each of these datasets are ordered by time, we respect this time ordering and use the first 70% of the records for training and the remaining 30% (i.e. most recent 30%) records for testing. We then report the MSE on the test set as a measure of evaluation for the different techniques with the results shown in Fig. 8, Fig. 9, Fig. 10 and Fig. 11.

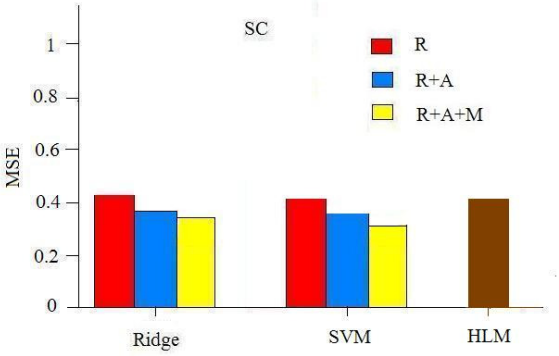

Fig. 8 SC denotes the supply chain dataset. The figure shows the relative performance of the two regression methods and HLM on the SC dataset where we know the aggregated targets.

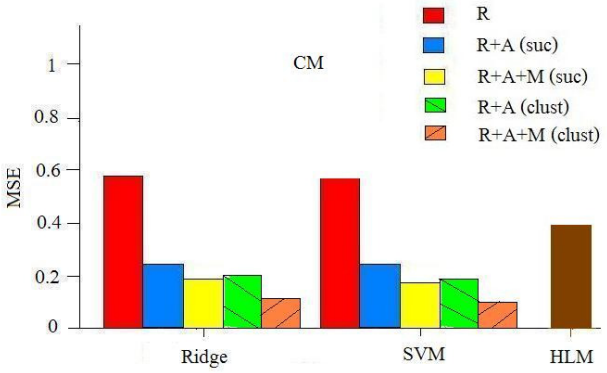

Fig. 9 CM denotes the chip manufacturing dataset. R+A (suc) and R+A+M (suc) denote the respective strategies deployed on data aggregated using successive points in the time series. R+A (clust) and R+A+M (clust) denote the respective strategies deployed on data aggregated using our agglomerative hierarchical clustering method outlined in section 3.4.2.

Supply Chain Dataset: This dataset is obtained from an actual manufacturer and contains data at two levels, namely at the (finer) distribution center (DC) level and at the (coarser) manufacturer level. The goal is to predict the inventory position at a DC given past inventory positions and other attributes such as age of the inventory and product type (viz. egg beaters, pasta etc.). In addition to this, we also have information about the total amount shipped (aggregate information) from the manufacturer to meet the demands of the DC. Hence in this case, we do not need to use algorithm in Fig. 5 or Fig. 6 since we already have aggregate information. We just need to use results in lemma 1 and lemma 3 (estimating the moments from the data). In our dataset, there are 7 distribution centers and the data was collected daily for about a year (dataset

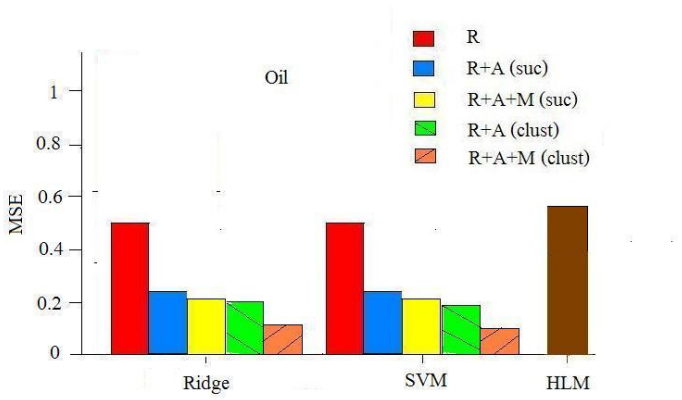

Fig. 10 Oil denotes the oil production dataset. R+A (suc) and R+A+M (suc) denote the respective strategies deployed on data aggregated using successive points in the time series. R+A (clust) and R+A+M (clust) denote the respective strategies deployed on data aggregated using our agglomerative hierarchical clustering method outlined in section 3.4.2.

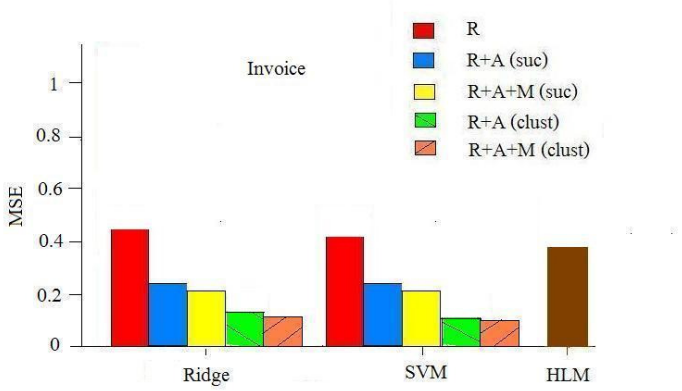

Fig. 11 Invoice denotes the dataset with invoice amounts of various BU-CC combinations. R+A (suc) and R+A+M (suc) denote the respective strategies deployed on data aggregated using successive points in the time series. R+A (clust) and R+A+M (clust) denote the respective strategies deployed on data aggregated using our agglomerative hierarchical clustering method outlined in section 3.4.2.

size is 357).

Chip Manufacturing Dataset: In the chip manufacturing industry, predicting speed of the wafers (collections of chips) accurately ahead of time can be crucial in choosing the appropriate set of wafers to send forward for further processing. Eliminating faulty wafers can save the industry a huge amount of resources in terms of time and money.

This dataset [2] has 175 features including wafer speed. The other features are a combination of physical measurements and electrical measurements made on the wafer. The dataset size is 2361. In this case, aggregate information is unavailable and, hence, we use the algorithm in Fig. 6 to estimate the aggregated targets, which can be viewed as estimating drift in the time series.

Oil Production Dataset: Oil companies periodically launch production-logging campaigns to get an idea of the overall performance as well as to assess their individual performance at particular oil wells and reservoirs. These campaigns are usually expensive and laden with danger for the people involved in the campaign. Automated monitoring of oil production equipment is an efficient, risk-free and economical alternative to the above solution.

The dataset we perform experiments on is obtained from a major oil corporation [7]. There are a total of 9 attributes in the dataset. These attributes are obtained from the sensors of a 3-stage separator which separates oil, water and gas. The 9 attributes are composed of 2 measured levels of the oil water interface at each of the 3 stages and 3 overall attributes. Our target is Daily production, which indicates the amount of oil produced every day at the well. The dataset size is 992. In this case also, aggregate information is unavailable and thus, we use the algorithm in Fig. 6 to estimate the aggregated targets.

Invoice Amounts Dataset: Large corporations have many of business units (BUs) viz. travel, marketing, auditing, information technology etc. with each BU consisting of various commodity councils (CCs) viz. office supplies, tech services, communication services, etc. Throughout a calendar year, each of the commodity councils carry out multiple transactions and record the corresponding invoices. It is extremely useful for these CCs and BUs at large to estimate in advance the invoice amounts that are likely to be registered in the near future.

This dataset has 8 BUs with each BU consisting of 150 CCs, leading to a total of $150 \times 8 = 1200$ attributes. The data was collected daily for a year and, hence, the dataset has only 365 datapoints. Our target is the CC communication services under the BU information technology, which has one of the highest invoice amounts and therefore is critical to business.

In this case, we cannot directly apply HLMs since the number of features is larger than the dataset size. Hence, we learn HLMs by considering only those CCs that lie under information technology.

Observations: In Fig. 8, Fig. 9, Fig. 10 and Fig. 11, we observe the behavior of the three variants, namely: 1) R, 2) R+A and 3) R+A+M, on the three datasets. First we see that the performance improves consistently as we go from R to R+A and from R+A to R+A+M. However, the extent of the improvement differs in the 3 cases. A possible reason that the improvement from R+A to R+A+M is more significant on CM than on the other datasets could be that the larger dataset size leads to more accurate estimates of the moments as compared to the other datasets. The improvement from R to R+A is more pronounced on CM, Oil and Invoice than on SC since the L returned by algorithm in Fig. 6 for the finest level are 3, 4 and 3 respectively, which is much less than 7 – the aggregation granularity for SC – and the inaccuracies in the estimates of the aggregated targets for these two cases are only slight.

From Fig. 9, Fig. 10 and Fig. 11, we also see that our clustering approach to form aggregates gives the best results, even when we know and use the explicit time-ordering to aggregate the data. A possible explanation for this is that successive datapoints in a time series are likely to be more similar than others and grouping them together leads to issues similar to those discussed in section 3.4.2.

On all the 4 datasets we also see that our approach is better than HLMs. The performance gain is more on CM, Oil and Invoice than on SC, indicating that being able to model a more general set of hierarchies than just nested is beneficial. In most cases (i.e. except Oil), the HLM's performance is somewhere between R+A and R, indicating that aggregating in these cases seems to have merit irrespective of the multilevel method used. In the Oil dataset, the relationships between the explanatory attributes and target are likely to be highly non-linear which leads to HLMs performing worse than R.

5 Discussion

In this paper, we proposed a provable way of improving prediction quality with the help of accurate aggregate or coarser information. In cases where we may have (estimates of) aggregated targets at multiple levels, we provided a way of choosing the optimal level so as to maximize the improvement in prediction quality. In standard machine learning settings, we have provided an algorithm for the same where aggregate information may not be available from an independent source. We also provided algorithms that can exploit aggregated targets at multiple levels and can group together datapoints when no explicit order or groupings are known. Moreover, using estimates of the moments of the target distribution, we have provided a way of better distributing the aggregate information so as to further enhance the predictive accuracy.

A different way of incorporating aggregate information would be to add this information as constraints in the formulations of optimization-based regression methods. In this case, however, we would have to describe for each of the regression methods how the modified objective could be effectively optimized. The constraints would range from linear to non-linear (with moment information). Such a procedure might be more effective for certain methods than our proposed approach. However, the extent and ease of applicability to different regression methods would be contingent on the specific methods. In contrast, our meta-learning approach can be readily applied to almost any regression method with relative ease, since it is agnostic to its inner workings. In other words, our approach should at the very least be a competitive baseline that can be used to validate the creation of such enhanced regression methods.

In the future, it would be interesting to expand the current line of work where we assume other types of information. For example, we may assume that we do not know the exact or approximate aggregated targets but rather distributions over them. We may also assume that rather than knowing the aggregated targets, we know lower (or upper) bounds on them. A realistic scenario with information of this nature could be envisioned. An electricity supply company, for instance, may have multiple transformers that provide power to hundreds of households, with each household being serviced by more than one transformer. In such a scenario, the power generated at a transformer would lower bound the total energy consumption of the households it serves. It would be interesting to decipher the optimal way in which this information could be used to accurately predict household consumption. Another promising direction would be to combine the methods in this paper with other advanced methods that are used

in relevant applications. For instance, we could amalgamate the ideas in this paper with ideas in [7,2] and possibly create more effective methods for multiple time series prediction problems. The creation of such sophisticated methods might be of some interest in the future.

Acknowledgement

I would like to thank Jayant Kalagnanam for providing data and exposing me to such problems. I would like to thank Naoki Abe and Aurelie Lozano for helpful discussions. I would also like to thank Katherine Turner for proofreading the paper.

Appendix

Proof of Lemma 1

Proof Consider two sets of N real numbers $Y = \{y_1, y_2, ..., y_N\}$ and $\bar{Y} = \{\bar{y_1}, \bar{y_2}, ..., \bar{y_N}\}$ (estimates). Let $A = \{a_1, ..., a_m\}$ and $\bar{A} = \{\bar{a_1}, ..., \bar{a_m}\}$ where if k is the aggregation granularity, $l_i = min(ik, N) - (i-1)k - 1$, $m = \lceil \frac{N}{k} \rceil$, then $a_i = \sum_{j=(i-1)k+1}^{min(ik,N)} y_j$ and $\bar{a_i} = \sum_{j=(i-1)k+1}^{min(ik,N)} \bar{y_j}$. Let $\epsilon_i = a_i - \bar{a_i}$ and $\hat{y_j} = \bar{y_j} + \frac{\epsilon_{\lceil \frac{j}{k} \rceil}}{l_{\lceil \frac{j}{k} \rceil}}$.

Let the mean squared error based on the original estimates i.e. $(\bar{y_i})$ be given by,

$$MSE_{old} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \bar{y_i})^2 \tag{3}$$

Hence, the mean squared error based on new estimates i.e. $(\hat{y_i})$ is given by,

$$\begin{aligned} & MSE_{new} \\ & = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y_i})^2 = \frac{1}{N} \sum_{i=1}^{N} ((y_i - \bar{y_i}) - \frac{\epsilon_{\lceil \frac{i}{k} \rceil}}{l_{\lceil \frac{i}{k} \rceil}})^2 \\ & = \frac{1}{N} \sum_{i=1}^{N} (y_i - \bar{y_i})^2 - \frac{1}{N} [\sum_{i=1}^{N} (\frac{2}{l_{\lceil \frac{i}{k} \rceil}} (y_i - \bar{y_i}) \epsilon_{\lceil \frac{i}{k} \rceil} - \frac{1}{l^2_{\lceil \frac{i}{k} \rceil}} (\epsilon_{\lceil \frac{i}{k} \rceil})^2)] \\ & = MSE_{old} - \frac{1}{N} A \end{aligned} \tag{4}$$

where $A = \sum_{i=1}^{N} (\frac{2}{l_{\lceil \frac{i}{k} \rceil}} (y_i - \bar{y_i}) \epsilon_{\lceil \frac{i}{k} \rceil} - \frac{1}{l^2_{\lceil \frac{i}{k} \rceil}} (\epsilon_{\lceil \frac{i}{k} \rceil})^2)$. Now to prove our result we have to show that $A \geq 0$.

$$
\begin{aligned}
A &= \sum_{i=1}^{N}\left(\frac{2}{l_{\lceil \frac{i}{k} \rceil}}(y_i - \bar{y_i})\epsilon_{\lceil \frac{i}{k} \rceil} - \frac{1}{l^2_{\lceil \frac{i}{k} \rceil}}(\epsilon_{\lceil \frac{i}{k} \rceil})^2\right) \\
&= \sum_{p=1}^{\lceil \frac{N}{k} \rceil}\left(\frac{2}{l_p}\left(\sum_{i=(p-1)k+1}^{min(pk,N)} y_i - \sum_{i=(p-1)k+1}^{min(pk,N)} \bar{y_i}\right)\epsilon_p - \frac{1}{l_p^2} l_p \epsilon_p^2\right) \\
&= \sum_{p=1}^{\lceil \frac{N}{k} \rceil}\left(\frac{2}{l_p}\epsilon_p^2 - \frac{1}{l_p}\epsilon_p^2\right) = \sum_{p=1}^{\lceil \frac{N}{k} \rceil} \frac{1}{l_p}\epsilon_p^2 \geq 0
\end{aligned} \tag{5}
$$

Proof of Theorem 1

Proof The proof of the theorem follows from the proof of lemma 1 where we substitute ϵ_i with $\delta_i \in [0, 2\epsilon_i]$ in equation 4. With this we have,

$$
MSE_{new} = MSE_{old} - \frac{1}{N}B \tag{6}
$$

where $B = \sum_{i=1}^{N}(\frac{2}{l_{\lceil \frac{i}{k} \rceil}}(y_i - \bar{y_i})\delta_{\lceil \frac{i}{k} \rceil} - \frac{1}{l^2_{\lceil \frac{i}{k} \rceil}}(\delta_{\lceil \frac{i}{k} \rceil})^2)$. Now to prove our result we have to show that $B \geq 0$.

$$
\begin{aligned}
B &= \sum_{i=1}^{N}\left(\frac{2}{l_{\lceil \frac{i}{k} \rceil}}(y_i - \bar{y_i})\delta_{\lceil \frac{i}{k} \rceil} - \frac{1}{l^2_{\lceil \frac{i}{k} \rceil}}(\delta_{\lceil \frac{i}{k} \rceil})^2\right) \\
&= \sum_{p=1}^{\lceil \frac{N}{k} \rceil}\left(\frac{2}{l_p}\left(\sum_{i=(p-1)k+1}^{min(pk,N)} y_i - \sum_{i=(p-1)k+1}^{min(pk,N)} \bar{y_i}\right)\delta_p - \frac{1}{l_p^2} l_p \delta_p^2\right) \\
&= \sum_{p=1}^{\lceil \frac{N}{k} \rceil}\left(\frac{2}{l_p}\delta_p\epsilon_p - \frac{1}{l_p}\delta_p^2\right) = \sum_{p=1}^{\lceil \frac{N}{k} \rceil} \frac{-1}{l_p}\delta_p(\delta_p - 2\epsilon_p)
\end{aligned} \tag{7}
$$

The above quadratic equation has 2 roots $\delta_p = 0$ and $\delta_p = 2\epsilon_p$ and we already know that $B \geq 0$ when $\delta_p = \epsilon_p$. Since, $\epsilon_p \in [0, 2\epsilon_p]$ and the function is a quadratic in δ_p we have $B \geq 0\ \forall \delta_p \in [0, 2\epsilon_p]$.

Proof of Lemma 2

Proof In equation 5 we substitute alphas as the fractions for the $\frac{1}{l_p}$ in each aggregated set since, we want to find the optimal weighting scheme. Optimizing each aggregated set will give the overall optimal solution and hence without loss of generality (w.l.o.g.) we have the following optimization problem,

$$
\begin{aligned}
&max \sum_{i=1}^{l_p} \epsilon_p^2 \alpha_i^2 \\
&subject\ \ to : \sum_{j=1}^{l_p} \alpha_j = 1
\end{aligned} \tag{8}
$$

Forming the lagrangian and setting the first derivative to zero we get, $\alpha_j = -\frac{\lambda}{2\epsilon_p^2}$, where λ is the lagrange parameter. Substituting this back into the constraint we get, $\lambda = -\frac{2\epsilon_p^2}{l_p}$. Combining the previous two results we get, $\alpha_j = \frac{1}{l_p}$.

Proof of Theorem 2

Proof In A substituting the alphas we have,

$$A = \sum_{p=1}^{\lceil \frac{N}{k} \rceil} 2\epsilon_p \sum_{i=(p-1)k+1}^{min(pk,N)} [(y_i - \bar{y_i})\alpha_i - \frac{\epsilon_p}{2}\alpha_i^2] \tag{9}$$

We thus have to show that when all alphas for a particular p are not equal then there always exist Y and $\bar{Y}$ such that $A < 0$. We can show this by proving that there always exist $\{y_{(p-1)k+1}, ..., y_{min(pk,N)}\}$ and $\{y_{(p-\bar{1})k+1}, ..., y_{min(\bar{pk},N)}\}$ such that the above equation for any particular p is less than zero and hence, if we replicate this case for all p then their sum is less than zero which implies $A < 0$. With this we have to show that for any p (in our setting), $2\epsilon_p \sum_{i=(p-1)k+1}^{min(pk,N)} [(y_i - \bar{y_i})\alpha_i - \frac{\epsilon_p}{2}\alpha_i^2] \leq 0$.

W.l.o.g. we will prove the above result for $p = 1$ and the proof should be valid for all p. Thus, we will show that when all alphas for $p = 1$ are not equal then there always exist $\{y_1, ..., y_k\}$ and $\{\bar{y_1}, ..., \bar{y_k}\}$ such that, $2\epsilon_1 \sum_{i=1}^{k} [(y_i - \bar{y_i})\alpha_i - \frac{\epsilon_1}{2}\alpha_i^2] \leq 0$.

Since all alphas are not equal, w.l.o.g. assume that $\alpha_1 > \alpha_2$ where $\alpha_1 \geq \alpha_i \ \forall i \in \{1, ..., k\}$ and $\alpha_2 \leq \alpha_i \ \forall i \in \{1, ..., k\}$. We will prove the result by dividing it into 2 cases. Case 1 is $\epsilon_1 \geq 0$ and case 2 is $\epsilon_1 \leq 0$. Notice that we have freedom to choose values for Y and $\bar{Y}$ to prove our result.

Case 1: We choose y_i and $\bar{y_i}$ such that $y_i = \bar{y_i} \ \forall i \in \{3, ..., k\}$ and $y_2 - \bar{y_2} \geq \bar{y_1} - y_1 \geq 0$. This forces $\epsilon_1 \geq 0$ as desired. Hence, for the previous equation to be true, a sufficient condition is, $\alpha_1(y_1 - \bar{y_1}) + \alpha_2(y_2 - \bar{y_2}) \leq 0$ which implies $\bar{y_1} - y_1 \geq \frac{\alpha_2}{\alpha_1}(y_2 - \bar{y_2})$. We can always find $y_1, \bar{y_1}, y_2$ and $\bar{y_2}$ such that $y_2 - \bar{y_2} \geq \bar{y_1} - y_1 \geq \frac{\alpha_2}{\alpha_1}(y_2 - \bar{y_2}) \geq 0 \ \forall \alpha_i$ where $i \in \{1, ..., k\}$.

Case 2: This is analogous to case 1. All the inequalities in case 1 can be reversed and thus, we need to find $y_1, \bar{y_1}, y_2$ and $\bar{y_2}$ such that $y_2 - \bar{y_2} \leq \bar{y_1} - y_1 \leq \frac{\alpha_2}{\alpha_1}(y_2 - \bar{y_2}) \leq 0$ $\forall \alpha_i$ where $i \in \{1, ..., k\}$, which is definitely possible.

Proof of Lemma 3

Proof Since we take expectations with respect to the underlying distribution for the result of this lemma the objective we have to maximize to get the optimal alphas is the expected value of equation 9 given a_p and $\bar{a_p}$, i.e. $E[A|a_p, \bar{a_p}]$. This function is concave in the alphas and hence by forming the lagrangian and maximizing the objective given the constraints on the alphas we get, $\alpha_p^{(i)} = \frac{1}{2\epsilon_p^2}[2\epsilon_p(\mu - \bar{y}_{i+k(p-1)} + \lambda)]; i \neq l_p$ and $\lambda = \frac{-2}{l_p}(2l_p - 1)(\sigma^2 + \mu^2)$, where λ is the lagrange parameter. Notice that $\alpha_p^{(l_p)}$ is uniquely defined since the alphas sum to 1. With this the optimal alphas are given by,

$$\alpha_p^{(i)} = \frac{1}{l_p\epsilon_p^2}[l_p\epsilon_p(\mu - \bar{y}_{i+k(p-1)}) - (2l_p - 1)(\sigma^2 + \mu^2)]; i \neq l_p$$

$$\alpha_p^{(l_p)} = \frac{1}{l_p\epsilon_p^2}[(2l_p - 1)(l_p - 1)(\sigma^2 + \mu^2) - l_p\epsilon_p((l_p - 1)\mu + \bar{y}_{i+k(p-1)} - a_p)]$$

References

1. A. Arnold, Y. Liu, and N. Abe. Temporal causal modeling with graphical granger methods. In *Knowledge Discovery and Data Mining*. ACM, 2007.
2. A. Dhurandhar. Multistep time series prediction in complex instrumented domains. In *Large-scale Analytics for Complex Instrumented Systems workshop, in International Conference on Data Mining*. IEEE, 2010.
3. T. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10:1895–1923, 1998.
4. F. Fleuret and D. Geman. Coarse-to-fine face detection. *Int. J. Comput. Vision*, 41:85–107, 2001.
5. T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2 edition, 2009.
6. C. Jackson, N. Best, and S. Richardson. Hierarchical related regression for combining aggregate and individual data in studies of socio-economic disease risk factors. *Journal Of The Royal Statistical Society Series A*, 171(1):159–178, 2008.
7. Y. Liu, J. Kalagnanam, and O. Johnsen. Learning dynamic temporal graphs for oil-production equipment monitoring system. In *Knowledge Discovery and Data Mining*, pages 1225–1234. ACM, 2009.
8. D. Munoz, J. Bagnell, and M. Hebert. Stacked hierarchical labeling. In *ECCV*, 2010.
9. M. Park, T. Hastie, and R. Tibshirani. Averaged gene expressions for regression. *Biostatistics*, pages 212–227, 2007.
10. S. Raudenbush and A. Bryk. *Hierarchical Linear Models*. Sage, 2 edition, 2002.
11. B. Sapp, A. Toshev, and B. Taskar. Cascaded models for articulated pose estimation. In *ECCV*, 2010.
12. J. Singer and J. Willett. *Applied Longitudinal Data Analysis: Modeling Change and Event Occurrence*. Oxford University Press, 1 edition, 2003.
13. P. Slav. Coarse-to-fine natural language processing. Phd Thesis UC Berkeley, 2009.
14. Jr. Ward. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58:236–244, 1963.
15. D. Weiss and B. Taskar. Structured prediction cascades. In *Proc. AISTATS*, 2010.