A probabilistic graphical model for evolving truth discovery

Fan Yang, Yisi Liu, Zheyi Zhu

University of Illinois at Urbana Champaign

Abstract

With the development of science and technology, data in our life is getting bigger and bigger. On the one hand, a large amount of data implies that the efficient and sophisticated way of utilizing data might provide huge benefits to our life. On the other hand, the conflicts among multi-source information produce a lot of difficulties in using the data efficiently. To tackle this challenge, truth discovery is emerging over time. A lot of truth discovery methods based on probabilistic graphical models are proposed for various scenarios. Here, we consider the scenario with evolving truth. In many real world applications, the information may come sequentially, and as a consequence, the truth of objects may be dynamically evolving. Most Existed truth discovery methods, unfortunately, cannot handle such scenarios. In this work, we propose a probabilistic graphical models to deal with evolving numerical data. Specifically, a hidden Markov model is proposed to model the evolving truth and efficient EM algorithm with Kalman filter and smoothers is designed with fully theoretical guarantees. Compared with previous methods, it has great power to deal with missing data, model the correlation between objects and sources. Furthermore, parameters in the model can be either automatic learned from model or predefined with explicit meanings. The experiments on two real world applications demonstrate its effectiveness over several available truth discovery methods.

Keywords: Truth discovery, Streaming data, Numerical data, Hidden Markov model, Kalman Filter
1. Introduction

In this Internet era, data is everywhere. A lot of websites provide up to date data, like flight information, hotel information, stock markets information, weather forecasting. It makes our life comfortable and full of useful information.

However, a large amount of data doesn’t mean the information implied by data is always correct. If we collect information about the same object from different sources, it is very common that data may conflict with each other due to errors, missing records, different scales, etc. Truth discovery methods are designed to find the more trustworthy information collected from various sources.

One naive approach to resolve the conflicts of the data is voting. That is, truth can always come with the highest votes. Actually, this is not always the case. For the more reliable source, its information should be more reliable even its result is not the most majority one. Thus, general principles are constructed in this area: if the piece of information is from a reliable source, then it is more trustworthy, and the source that provides trustworthy information is more reliable.

Most of the existing truth discovery algorithms are proposed to work on static data. They can not handle the scenarios where the collected information comes sequentially, which happen in many real world applications. Consider the aforementioned applications: The weather condition, the stock information, and the flight status are collected in real-time. These applications reveal the necessity to develop truth discovery methods for such scenarios.

For the dynamic data, unique features are found. First, truths might not be a constant. It could evolve with time going. Figure 1 shows the evolving truths of market capitalization and flight arrival time. Three firms market capitalization data is shown here (Figure 1(a), 1(b) and 1(c)). One firm has overall increasing market capitalization; another one has decreasing market capitalization; while, 'akam' first has stable market capitalization, then has a downhill jump. For market capitalization, it should vary all the time. Assuming it to be constant is definitely not reasonable. Another example is about the flight arrival time. Though the scheduled time of the same flight should be constant. However, in reality, the arrival time will vary for a lot of reasons. Figure 1(d), 1(e) and 1(f) shows three flights (AA-404-MIA-MCO, AA-3379-CVG-ORD, UA-1500-IAH-GUA) information about arrival
time over a month. The arrival time is not stable at all. Sometimes, the flight arrives early. Sometimes, delay might happen.

If we look further, we will find the second future of the dynamic data. For a specific object, the temporal smoothness exists among its information at different timestamps. Figure 2 shows the auto-correlation function of market capitalization and flight arrival data. We can find auto-correlation from the graph and for a lot of times, this effect is statistical significant. Thus, it implies that Markov process might be a potential candidate for us to model the truth dynamics. This is extremely useful when observations or source number is not big at every timestamp. This smoothness feature can help us to estimate or predict the truth in the future.

In this work, a new truth discovery method for evolving numerical data based on hidden Markov model is developed for dynamic scenarios. We take into account evolving truth, source quality, source correlation, objects correlation in our model. The case study shows its effectiveness compared with previous methods.

In summary, our contributions in this work are:

- Take into account evolving truth features based on rigorous probability and statistics methods with theoretical guarantees.
- First use hidden Markov model to deal with evolving numerical truths.
- \(O(T)\) one pass algorithm is developed.
- Case study shows its effectiveness over several real dataset.

2. The model

We first formulate the problem of evolving truth discovery with clear descriptions of the notation in this work. Then, we will build the hidden Markov model for truth discovery with streaming data. Based on it, efficient blocked Kalman filtering algorithm is proposed. We show that our model has several advantages including missing data handling, dynamic parameter learning compared with previous methods.

2.1. Problem Formulation

We follow the notations in [1].
2.1.1. Input

Consider a set of objects $O$ that we are interested in, and for each of them $o \in O$, related information can be collected from $S$ sources at each timestamp $t \in \{1, 2, 3, \ldots\}$. Let $v_{o,t}^s$ represent the information from the $s$-th source about the object $o$ at the $t$-th timestamp. For convenience, we denote all the information from source $s$ at time $t$ as $X_t^s$, that is, $X_t^s = \{v_{o,t}^s\}_{o \in O}$. Further, the size of this set is denoted as $c_t^s = |X_t^s|$.

2.1.2. Output

After collecting information from different sources, our goal is to aggregate these information and output trustworthy ones (truth). Let $\mu_{o,t}$ be the aggregated result (truth) for object $o$ at time $t$, and $X_t^*$ be the whole set of aggregated results at time $t$.

Besides the aggregated results, truth discovery methods can also estimate source reliability degrees. Let $\Sigma_s$ denote the source covariance matrix. We use its diagonal element $\sigma_s^2$ to model the source quality. Its off-diagonal elements $\sigma_{s_i,s_j}^2$ can be used to measure the source correlation information.

We use $T^f$ to represent the total time or terminal time. Whenever we use $a : b$ where $a, b$ are arbitrary integers and $b \geq a$, it represents the set
Task definition

The studied task is formally defined as follows. For a set of objects we are interested in, at timestamp $T$, related information is collected from $S$ sources. Our goal is to find the most trustworthy information $\mu_{o,1:T}$ for each object $o$ by resolving the conflicts among information from different sources $\{v_{o,T}^s\}_{s=1}^S$. Meanwhile, source quality information, source correlation information, objects correlation information and truth evolving dynamics are able to be inferred. To guarantee the efficiency, a modified algorithm is proposed which should not re-visit the information at previous timestamps to achieve $O(T)$ time complexity.

Hidden Markov model

We first build a hidden Markov model for evolving truth discovery. Figure 3 shows the dynamics of truths and observations. We focus on numerical data here. $X_t^*$ is the whole set of aggregated results at time $t$. Markov process is used to model the dynamics of the truths. Assume objects will not change as time going and $\mu_t$ denote the vector of truths in $X_t^*$. Then, the dynamics
can be written in the following form:

\[ \mu_{t+1} = A\mu_t + \epsilon_t^\mu \quad (1) \]

\( \mu_t \) includes all truths at timestamp \( t \) which is the latent variable we want to estimate. Equation (1) represents the Markov process for latent variables (states). \( A \) is the transition matrix of latent states. If \( A \) is a diagonal matrix, truth of each object will depend on its own previous state. Matrix \( A \) could be either given or estimated from the model. We assume \( \epsilon_t^\mu \) follows multivariate normal distribution:

\[ \epsilon_t^\mu \sim \text{Normal}(0, \Sigma_t^\mu) \quad (2) \]

\( \Sigma_t^\mu \) is the covariance matrix of the truths of all objects at timestamp \( t \). It could be either predefined or estimated.

\( v_{o,T}^s \) is the information for object \( o \) from source \( s \) at time \( t \). We assume it fully depends on the truths at time \( t \) following multivariate normal distribution:

\[ v_t = \vec{\mu}_t + \epsilon_t^s. \quad (3) \]

\( v_t \) here is the observation vector including all observed objects from their sources at timestamp \( t \). It is firstly ordered by sources, then, by objects. Specifically, if the observations of all sources and all objects are available, \( v_t = (v_{o=1,t}^{s=1}, v_{o=1,t}^{s=2}, \ldots, v_{o=1,t}^{s=n_s}, v_{o=2,t}^{s=1}, \ldots, v_{o=1, t}^{s=n_s})^T \). We assume the (unbiased) mean of the observations are the truths. To match the dimension and setting of vector \( v_t \), \( \vec{\mu} \) is the further vectorized \( \mu_t \) in the form \( (\mu_1 \otimes 1_{n_i=1}, \ldots, \mu_{n_o} \otimes 1_{n_t=\sigma_t^s})^T \). \( \otimes \) is the Kronecker product. \( n_o \) is the number of objects. \( n_s \) is the number of sources. \( n_t^{o=1} \) represents the number of observed values of object \( i \) at time \( t \). \( 1_n \) is the vector of \( n \) repeated element 1. \( \epsilon_t^s \sim \text{Normal}(0, \Sigma_t^s) \)

If all objects from all sources are available at time \( t \), \( \Sigma_t^s \) has the form:

\[
\Sigma_t^s = \begin{pmatrix}
\Sigma_s & * & * & \cdots & * \\
* & \Sigma_s & * & \cdots & * \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
* & \cdots & \cdots & \cdots & \Sigma_s
\end{pmatrix}
\quad (4)
\]
\[ \Sigma_s = \begin{pmatrix} \sigma_{1,1}^2 & \sigma_{1,2} & \sigma_{1,3} & \cdots & \sigma_{1,n_s} \\ \sigma_{2,1} & \sigma_{2,2}^2 & \sigma_{2,3} & \cdots & \sigma_{2,n_s} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \sigma_{n_s,1} & \cdots & \cdots & \sigma_{n_s,n_s-1} & \sigma_{n_s,n_s}^2 \end{pmatrix} \] (5)

In most cases, not all objects or all sources information is available at time \( t \), \( \Sigma_s^t \) will be the sub-matrix of Equation (4). Furthermore, "*" in Equation (4) can be assumed 0 if there is no evidence that observed values for different objects are correlated. If we assume sources are not correlated to each other, \( \sigma_{i,j} \) can also be assumed 0 in Equation (5). Otherwise, it can be estimated based on the data and model.

\( \sigma_s^2 (\sigma_1^2, \sigma_2^2 \ldots) \) can be regarded as source quality parameter because large variability of observed value could most likely be from unreliable source. Off-diagonal elements in \( \Sigma_s \) can be used to model the correlation of different sources.

![Figure 3: Model dynamics](image)

2.3. Blocked Kalman Filtering algorithm with EM

Here, we choose to use EM algorithm to solve the model.

In the M step, the parameter's in the system are re-estimated by taking the corresponding partial derivative of the expected log likelihood, setting to zero, and solving.

In the E step, we use Kalman filter and Kalman smoother to get \( E[\mu_t | v_{1:T}] \), \( E[\mu_t \mu'_t | v_{1:T}] \) and \( E[\mu_t \mu'_{t-1} | v_{1:T}] \). Then, those quantities will be used in E step.
If many objects are considered in the model, the dimension of $\mu_t$ will be high. The matrix calculation in E-step (Kalman filter and smoother) will take huge memory and turns out to be computationally intensive. Thus, in the E step, we implement Kalman filter and smoother in the blocked way following the theorem 1 below.

**Theorem 1.** If three conditions below satisfied:

- The blocks of $v_t$ and $\mu_t$ have no overlaps.
- Given truths at time $t-1$, $\mu_{t-1}$, truths in different blocks are independent at timestamp $t$.
- Observations in different blocks are independent given truths information at any timestamp $t$.

Then, truth information in each block $b$, $P(\mu^b_t|v_{1:t})$ or $P(\mu^b_t|v_{1:T})$, are independent of each other. That is, we can implement Kalman filter and Kalman smoother in E step to decomposed blocked objects independently.

One good advantage of the EM algorithm here is that analytical solution for all parameters are available. Detailed algorithm with analytical solution of parameters will be updated soon.

### 2.4. Data preprocessing

We implement data preprocessing step before EM algorithm following the similar way of [2]. The reason for us to conduct data preprocessing is that:

- Reduce outliers effect so that normal distribution assumption in the model makes sense.
- Make all objects in same scale. This is extremely helpful for us to define prior information.

Based on robust statistics, the sample mean can be shifted infinitely by outliers and therefore is not a good prior, but some measures, such as the median or the output of non-numerical truth finding algorithms, would be more robust and suitable to serve as the prior. With the prior given, any claimed values deviating too far from the prior can be treated as outliers,
and there are various ways to measure the deviation, such as relative errors, absolute errors, Gaussian p-values, etc.

To detect outliers of each object’s observed value at time $t$, we use the median absolute deviation to find outliers ([3]). After detecting and deleting all outliers, we normalized all observed values to its $z$-scores for the later truth-finding EM algorithm. Specifically, for each object $o$ at time $t$, normalized $v_{o,t}^s$ is $(v_{o,t}^s - \text{mean}(v_{o,t}^s))/\text{std}(v_{o,t}^s)$. If there are not many data at time $t$ for object $o$, we can aggregate several times together by setting time window to normalize the data.

2.5. Other available algorithms

Other than EM algorithm with blocked kalman filter algorithm, we have several other choices to deal with different conditions.

- Large scale of objects without block structure: Localized ensemble Kalman filtering and smoothing. (O(T) time complexity)

- Non-linear or non-Gaussian setting: particle filtering (O(T) time complexity): Liu-west filter, Stovrik filter, particle learning.

- Other advanced Monte Carlo methods (O($T^2$) time complexity): MCMC (Gibbs, Metropolis, Gibbs-EM), particle MCMC, Variational Bayes (EM)......

ensemble kalman filter and smoother can be used in the case of large scale of data which can not be decomposed into smaller space. More advanced sequential Monte Carlo methods can be used to deal with the non-linear case in O(T) complexity. Other advanced Monte Carlo methods are potential tools for us to do parameter estimation and latent states estimation more accurately.

2.6. Missing data

One advantage of our model is that it has simple way to deal with missing data. Missing data can be regarded as unknown parameters to be estimated by using maximum likelihood estimation. That is, all unavailable information from source $s$ about object $o$ at time $t$ can be treated as missing data and our model can be converted to a standard hidden Markov model with equal number of observations from all sources, all objects at every time $t$ with
several missing data. Then, typical kalman filter and smoother algorithm can be implemented.

What’s more, to deal with missing data, you can just adjust the kalman filter and smoother algorithm by putting the parameter before $\mu_{0,t}$ in Equation 3 to be zero and delete $y_t$ related terms. In other words, those truths without observed values will fully depends on the dynamics of the truths in Equation (1). Detailed algorithm with missing data will be updated soon.

2.7. Comparison with previous methods

If we assume object truths are independent of each other; source are independent of each other, and, observed values of objects given the truth of objects are independent, our model will be simplified with similar solution to the model in [1]. We compare our simplified model with the model in [1] here. If we assume object truths are independent at all time, $\Sigma_t^{\sigma}$ in Equation (2)
will be diagonal matrix. $\Sigma_s$ and $\Sigma_s$ are also be diagonal matrix in Equation (4) and Equation (5).

The M step of E-M algorithm gives us following expression about source quality $\sigma_s^2$ at time $T$:

$$
\sigma_s^2 = \frac{2\beta_s + \sum_{t=1}^{T} \sum_{o \in \chi_t^s} E((v_{o,t}^s - \mu_{o,t})^2|v_{1:T})}{2(\alpha_s + 1) + \sum_{t=1}^{T} c_t^s}
$$

(6)

where $(\alpha_s, \beta_s)$ is the parameter in the prior distribution, Inv-Gamma, of $s$'s source quality $\sigma_s^2$.

In [1], source quality is modeled as source weight in the following way:

$$
w_s = \frac{2(\alpha_s - 1) + \sum_{t=1}^{T} c_t^s}{2\beta_s + \sum_{t=1}^{T} \sum_{o \in \chi_t^s} ((v_{o,t}^s - \mu_{o,t})^2)}
$$

(7)

where $(\alpha_s, \beta_s)$ is the parameter in the prior distribution, Gamma, of $s$'s source weight $w_s$. $\mu_{o,t}$ is estimated by using source weighted summation of observed values.

From Equation (6) and (7), we can see that even if, our model is based on probability and [1]'s model is based on optimization, our source quality parameter $\sigma_s^2$ is similar to the inverse of source weight $w_s$ in [1]. That is, $\sigma_s^2 \approx 1/w_s$.

To update the truth information, in our E-step, we have the following expression:

$$
E(\mu_{o,t}|v_{1:t}) = A_o \times E(\mu_{o,t-1}|v_{1:(t-1)}) + K_{o,t}(v_{o,t}^s - A_o \times E(\mu_{o,t-1}|v_{1:(t-1)}))
$$

(8)

$K_{o,t}$ is kalman Gain matrix related to source quality matrix $\Sigma_s$. $E(\mu_{o,t}|v_{1:T})$ will be similar to $E(\mu_{o,t}|v_{1:t})$ based on Kalman smoothing updating.

In [1]'s model, the updated $\mu_{o,t}$ is:

$$
\mu_{o,t} = \frac{\sum_{s=1}^{n_s} w_s v_{o,t}^s + \lambda \hat{\mu}_{o,(t-1)}^s}{\sum_{s=1}^{n_s} w_s + \lambda}
$$

(9)

We can see both Equation (8) and (9) balance the new observed value and previous state estimation. But, ours balance is dynamic based on Kalman Gain. In [1]'s model, the balancing parameter $\lambda$ need to be predefined and fixed.
2.8. An $O(T)$ version of algorithm

- Data Preprocessing: reduce outliers effect and make all objects in the same scale

- For timestamp $t$ from 0 to $T$:
  - Conduct EM algorithm sequentially.
  - E-step: Given parameters value, obtain truth information $E(\mu^t|o^{1:t})$ and $Var(\mu^t|o^{1:t})$ (Kalman Filter)
  - M-step: Given truth information, update all parameters (source quality, transition matrix, initial state, Object covariance matrix.....) by using $E(\mu^t|o^{1:t})$ and $Var(\mu^t|o^{1:t})$.

Time complexity: $O(T)$

Initial parameter setting: 1. Equal source quality setting ($\sigma_i = 1$ for every source based on scaled data) 2. Estimated parameter from training set.

3. Case study

In this section, we conduct case study with market capitalization data and flight arrival data. Both of data can be found in [4].

3.1. Market capitalization data

The market capitalization information is from the trading data of 1000 stock symbols from 55 sources on every work day in July 2011.

- Objects: 1000 stocks (Scale: $\times 10^8$).
- Sources: 55 sources.
- Timestamp: 20 days.
- Missing data: None.
- Truth data (for test): Information on NASDAQ100 stocks and another 100 randomly selected stocks collected by taking the majority values provided by five stock data providers: nasdaq.com, yahoo finance, google finance, bloomberg and MSN finance.
• Methods: DynaTD+ALL in (Li et al. 2015), Dymedian (Median at each timestamp) and our method

3.2. Flight arrival data
The flight arrival time information is from the flight data of over 3000 flights from 38 sources over 1-month period (December 2011).

• Data source: Truth Finding on the Deep Web: Is the Problem Solved?[4]
• Objects: Around 3000 flights.
• Sources: 38 sources.
• Timestamp: 31 days.
• Missing data: Yes. At some timestamps, no source information available for some flights arrival time.
• Truth data (for test): 100 randomly selected flights provided by corresponding airline websites.
• Methods: DynaTD+ALL in (Li et al. 2015), Dymedian (Median at each timestamp) and our method.

3.3. Result
We use mean of absolute error (MAE) and root of mean squared error (RMSE) to measure the performance of truth discovery algorithm. In this case study, we just assume that we have source independence and objects independence assumption. Actually, our algorithm can deal with more general cases. Our methods (i,j) means that we use first i days to train the model and conduct O(T) version algorithm in the remaining j − i days.

Table 1 shows the performance comparison. DybaTD+ALL is the methods mentioned in [1]. Dymedian is the method to calculate median at each timestamp t.

Our methods always have a better performance compared with DynaTD+ALL and Dymedian. More interestingly, we find no training set is a little better than the methods with all training set. Training set is primarily used to estimate the parameter. If we use all training set, that means, source quality parameter will be fixed and constant at each timestamp. If no training set specified, algorithm will learn and update the parameter when new data
Table 1: Performance Comparison

<table>
<thead>
<tr>
<th>Methods</th>
<th>Stock Dataset MAE</th>
<th>Stock Dataset RMSE</th>
<th>Flight Dataset MAE</th>
<th>Flight Dataset RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our methods (20,20)</td>
<td>7.66</td>
<td>35.95</td>
<td>3.44</td>
<td>8.68</td>
</tr>
<tr>
<td>Our methods (10,20)</td>
<td><strong>7.60</strong></td>
<td>33.77</td>
<td>3.42</td>
<td>8.66</td>
</tr>
<tr>
<td>Our methods (0,20)</td>
<td>7.64</td>
<td><strong>33.30</strong></td>
<td><strong>3.04</strong></td>
<td><strong>8.49</strong></td>
</tr>
<tr>
<td>DynaTD+All</td>
<td>12.29</td>
<td>37.03</td>
<td>25.76</td>
<td>42.95</td>
</tr>
<tr>
<td>Dymedian</td>
<td>8.43</td>
<td>42.74</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>

coming in. Thus, parameter will change all the time. It seems implies that a constant source quality is not a good assumption based on the data we use. Because, a constant source quality parameter implied by the model "Our methods (20,20)" doesn’t have a better performance. Data set with longer time span is needed to study for this phenomenon in the future.

Another point we find is that data-preprocessing is necessary in the truth discovery framework. DynaTD+All sometimes doesn’t have a good performance because it is really hard to specify the prior information for all objects because they are in different scales. This might be one reason DynaTD+All doesn’t have a bad performance for flight dataset.

- MAE: Mean of Absolute Error, RMSE: Root of Mean Squared Error.
- Our methods (i,j): 1. Objects and source independence assumption. 2. Use first $i$ days to train the model and conduct $O(T)$ version algorithm in the remaining $j - i$ days.

4. Advantages and future directions

Our methods have several advantages and future directions as follows:

- Advantages:
  - Analytical solution with reasonable expressions.
  - Handle limited source number and missing data.
  - Dependency modeling.
  - Flexible setting and combination with other models.
<table>
<thead>
<tr>
<th>Table 2: Source quality</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Stock Dataset</strong></td>
</tr>
<tr>
<td><strong>Ranking</strong></td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td><strong>Flight Dataset</strong></td>
</tr>
<tr>
<td><strong>Ranking</strong></td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
</tbody>
</table>

- Future directions
  - Evolving source quality.
  - Hidden Markov model for truth discovery in categorical streaming data.
  - Truth discovery in unstructured streaming data.

References


[2] B. Zhao, J. Han, A probabilistic model for estimating real-valued truth from conflicting sources (2012).