Recommending Missing Sensor Values

ABSTRACT

Datasets gathered from sensor networks often suffer from a significant fraction of missing data, due to issues such as communication interference, sensor interference, power depletion, and hardware failure. Many standard data analysis tools such as classification engines, time-sequence pattern analysis modules, and even statistical tools are ill-equipped to deal with missing values—hence, there is a vital need for highly-accurate techniques for imputing missing readings prior to analysis.

In this paper, we present novel imputation methods that take a "Recommendation Systems" view of the problem: the sensors and their readings at each time step are viewed as products and user product ratings, with the goal of estimating the missing ratings. Sensor readings differ from product ratings, however, in that the former exhibit high correlation in both time and space. To incorporate this property, we modify the widely successful Matrix Factorization (MF) approach for recommendation systems to model inter-sensor and intra-sensor correlations and learn latent relationships among these dimensions. We evaluate the approach using two environmental sensor network datasets, one indoor and one outdoor, and two imputation scenarios, corresponding to intermittent readings and failed sensors.

Next, we consider sensor networks with multiple sensor types at each node. We present two techniques for extending our model to account for possible correlations among sensor types (e.g., temperature and humidity). Our results show that both techniques can significantly improve the accuracy.

Finally, we consider a popular data analysis task—building regression-based prediction models—and show that, compared to prior approaches for imputation, using our model leads to a much higher quality prediction model.

1. INTRODUCTION

Wireless sensor networks (WSNs) are especially susceptible to interference, battery depletion, hardware failures, and other environmental and communications ailments that lead to data loss. Datasets gathered from sensor networks[1] are often missing a significant fraction of the possible readings (e.g., the Intel Berkeley Research lab dataset [2] is missing roughly 50% of its readings). These missing values are problematic for data analysis tools such as classification engines, time-sequence pattern analysis modules, and other machine learning tasks, which are often ill-equipped to deal with missing values. Support Vector Machine (SVM) and Multiple Regression (MR) analysis, to name but a few examples, require complete datasets with no missing values. Popular statistical packages such as SAS, Stata, and R provide a few default options for handling missing data, as a preprocessing step, because the core algorithms require that all data be filled in. Typical options are (i) remove the entire "column" if there is a missing value or (ii) fill in the missing value (called *imputation*) using either simple defaults like the average of neighboring values or utilizing user-written code. The first option discards otherwise useful data, and in fact, may discard most of the columns in datasets with high data loss. Thus, imputation is a vital tool in the preparation of sensor data for subsequent analysis. Because the accuracy of the target data analysis depends on the accuracy of the imputation, improvements in sensor data imputation better serve sensor network deployment objectives.

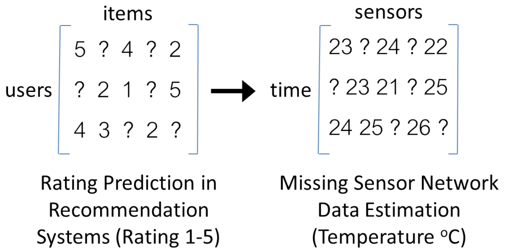

items
users
5 ? 4 ? 2
? 2 1 ? 5
4 3 ? 2 ?
sensors
time
23 ? 24 ? 22
? 23 21 ? 25
24 25 ? 26 ?
Rating Prediction in Recommendation Systems (Rating 1-5)
Missing Sensor Network Data Estimation (Temperature °C)

Figure 1: Bridge from Recommendation Systems to Sensor Data Imputation

1.1 Our Approach: Collaborative Filtering

Unlike prior work [1, 9–11, 13, 15, 16, 22, 23, 25], this paper presents a *Collaborative Filtering* (CF) approach to sensor data imputation, inspired by the field of Recommendation Systems. In typical CF approaches, the elements of interest are users and items (e.g., products), and the values are user ratings of those items (as in the left-hand side of Figure 1). Typically, most of the ratings are missing, and the goal is to predict (impute) the missing ratings in order to "recommend" items to users. By viewing sensors as items, users as time steps, and readings as ratings (as illustrated in Figure 1), we can

[1] We consider the common setting where sensor readings are collected in order to perform centralized analysis.

apply CF techniques to perform sensor data imputation. (Alternatively, sensors can be viewed as users and items as time steps—the mapping is irrelevant to the CF formulation.) In particular, we focus on the widely successful *Matrix Factorization* (MF) technique for CF.

Sensor readings differ from user ratings, however, in that the former often exhibit high correlation in both time and space. To incorporate this property, we first modify MF to model temporal correlations and learn latent relationships among sensors. Specifically, we add temporal-proximity terms to MF—we call this *Temporally-Regularized MF* (TR-MF). Similarly, we can also add spatial-proximity terms—we call this *Spatially-Temporally-Regularized* MF (STR-MF).

Second, we consider sensor networks with multiple sensor types at each node. We are readily able to exploit such heterogeneous sensor information in our solution, in contrast to most prior imputation methods that use more ad hoc means. We present two techniques for extending TR-MF to account for possible correlations among sensor types: *Multivariate-TR-MF* and *Temporally-Regularized Tensor Factorization* (TR-TF).

We evaluate our approaches using two environmental sensor network datasets, one indoor and one outdoor. We study two patterns for missing data: (i) covering *random* readings (modeling intermittent reading failures) and (ii) covering *consecutive* readings for some sensor nodes (modeling long temporal gaps such as with failed sensors).

Our study shows that TR-MF provides significantly higher estimation accuracy than both (i) state-of-the-art recommendation models and (ii) state-of-the-art sensor data imputation approaches (discussed in Section 2). Furthermore, our study shows that STR-MF, which adds spatial coordinate information into TR-MF, is useful *only* in the "consecutive" pattern—perhaps surprisingly, STR-MF is significantly *less* accurate than TR-MF in the "random" pattern. This is because TR-MF effectively leverages the latent relationships among sensors, including any spatial correlations, while avoiding the pitfalls of spatial-proximity biases (Section 2.2).

For the heterogeneous setting, our study shows that both Multivariate-TR-MF and TR-TF can significantly improve the accuracy over TR-MF, and each has its strengths, depending on the observed variance in the readings. Finally, we consider a popular data analysis task—building regression-based prediction models—and show that, compared to prior approaches for imputation, using TR-MF leads to a much higher quality prediction model.

These results validate our novel approach of equating sensor imputation with recommendation systems. Correlations are captured by grouping correlated sensor nodes and correlated time steps—unlike prior sensor data imputation approaches, our CF approaches use this *latent* information to impute values, and optimize the evaluation metrics directly. Moreover, our CF approaches are global, taking into account all collected observations, and not overly tied to spatial-proximity correlations.

1.2 Contributions

In summary, the main contributions of this paper are:

- We propose viewing sensor data imputation as a recommendation systems problem, and apply state-of-the-art collaborative filtering methods of recommendation systems (namely, matrix and tensor factorization) to the sensor network domain.
- We augment collaborative filtering with temporal regularization and multi-sensor signals, and provide efficient optimization methods to learn the inherent model parameters effectively.
- We present an empirical study on two sensor datasets, considering two missing data patterns corresponding to intermittent readings and failed sensors. The results show that our proposed approaches provide significantly higher estimation accuracy than state-of-the-art prior approaches, and moreover, such accuracy improvements can result in the generation of higher-quality prediction models.

2. RELATED WORK

Imputation techniques applied to sensor data can be divided into three categories by the information utilized: temporal methods, spatial methods, and spatio-temporal methods. They can be further categorized as hot-deck imputation and prediction models [8], as shown in Table 1.

2.1 Temporal Methods

Temporal methods leverage the temporal correlation among readings by the same sensor node; salient methods include observed data mean [20], last seen [9], and linear interpolation. These methods suffer, however, when there are long temporal gaps in observations for a given sensor; such gaps can be frequent in WSNs due to power depletion in energy-harvesting sensors, long-lived communication ailments, etc. As a result, the usefulness of temporal methods drops rapidly as the number of consecutively missing readings becomes large.

2.2 Spatial Methods

Spatial methods leverage the spatial correlation among readings by nearby sensor nodes; salient methods include associations rule mining (e.g., WARM [15] and FARM [10]) and weighted functions of nearby sensors (e.g., DEPM [16] and MI [25]).

Table 1: Sensor Data Imputation Methods

	Hot-Deck Imputation	Prediction Models
Temporal	Last-seen [9], Mean	Linear Interpolation
Spatial	WARM [15], FARM [10]	DEPM [16], MI [25]
Spatio-Temporal	STI [11]	DESM [16], ImM [18], AKE [23], BGP [22], EOF [1,13]

Window Association Rule Mining (WARM) [15] and Freshness Association Rule Mining (FARM) [10] study the estimation of missing data based on the association rules among spatially-correlated neighbors. Such methods enjoy the advantage of being able to handle categorical sensor data, but the estimation quality is limited in continuous sensor data by the association rules' requirement to first quantize the data.

The Data Estimation using Physical Model (DEPM) [16] method employs the basic laws of physics to design its prediction function. However, such models are applicable only to limited types of signals and environments, and generally require an accurate three-dimensional distance among sensors.

The Multi-Im (MI) method [25] imputes the missing data by replacing each missing value with a set of plausible values. Methods used in MI for plausible values include multiple linear regression, propensity score method, and Markov chain Monte Carlo method. This model suffers from high computation cost because multiple models have to be learned.

Generally speaking, there are two ways to obtain the spatial correlation: from the spatial coordinates or from the data itself. However, (only) the former requires accurate spatial coordinates, and more importantly, it suffers when ailments arise that affect entire spatial regions (such as large, temporary obstacles to sensing and/or communication). It also fails to account for barriers or other sources of sharp environmental gradients. For example, two nearby sensors, one near a stove and one beside a window, can produce very different values if the stove is in use and the outside temperature is low. In the end, using spatial coordinates can often lead to worse imputation results as non-existent or time-varying correlations are imposed between nearby sensors.

2.3 Spatio-Temporal Methods

Spatio-temporal methods consider both the temporal and spatial correlation; salient methods include STI [11], DESM [16], ImM [18], AKE [23], BGP [22], and EOF [1, 13].

The Spatial and Temporal Imputation (STI) method [17] favors spatial information over temporal information. For each missing reading, STI first checks if any nodes are within the sensing neighborhood (i.e., within a threshold distance), and utilizes the average of these neighbors to impute the missing value. If no sensors are within the neighborhood, the last seen value of the missing sensor is used for imputation. In Data Estimation using Statistical Model (DESM) [16], a missing reading is predicted using the linear combination of the previous reading of the sensor and the current reading of the neighboring sensor, weighted by the Pearson correlation between the two sensors. The Imputation Method (ImM) [18] learns to combine two temporal predictors (the last-seen predictor and an autocorrelation-based temporal linear predictor) and one spatial linear predictor.

The Applying K-nearest neighbor Estimation (AKE) method [23] adopts linear regression models to describe the spatial relationship among each pair of sensor nodes. The prediction of a target node is the weighted combination of the regression models from its K-nearest neighbors. If there is not enough spatial information to be exploited, AKE will use temporal information to help.

The Bayesian Gaussian Process (BGP) method [22] is a recently proposed random process method that assumes the current readings are Gaussian distributed given the past data. The parameter, mean and covariance are changeable from time to time based on domain knowledge on the mean and covariance function. In contrast, we focus in this paper on sensor data imputation that does not rely on domain knowledge.

The above methods suffer from making unverified assumptions about the data. For example, some models favor spatial correlation over temporal correlation (or vice versa), while some assume that sensors of the same distance should have a similar correlation regardless of their orientation or other factors. Such assumptions may or may not hold for various datasets, and hence imposing them a priori can lead to inaccurate results. The models we propose, however, try to rely less on such a priori knowledge, and learn the latent correlation directly from the data.

Finally, conceptually closer to our solution are the Singular Value Decomposition (SVD)-based methods. Conventional SVD has a significant limitation as it can only be applied to a complete matrix. Therefore, one needs to somehow first fill in the missing values before conducting such decomposition. The initial assignment of those missing values, unfortunately, can significantly affect the prediction accuracy [14]. Furthermore, SVD is computationally expensive in general. In contrast, our approach relies on the factorization process on an incomplete matrix.

A salient example of SVD-based approaches for imputation is the Empirical Orthogonal Functions (EOF) model, which has been applied to oceanographic applications to solve the problem of missing or unreliable satellite data [1]. EOF first fills in the missing values (e.g., using all zeros or the mean values) and then performs SVD to decompose the matrix. The top-K singular vectors in the SVD are used to reconstruct the matrix and update the estimation of missing values, where K is determined through cross validation over the data. An improved version of EOF [13] also considers the temporal ordering: The readings from one sensor node are copied M times and form M lag-shifted time series. The modification boosts the accuracy of EOF, but significantly slows down the method due to its M-fold data size increase. Moreover, our study shows that the improved EOF is still less accurate than our model.

To the best of our knowledge, ours is the first paper to propose applying matrix factorization techniques to sensor data imputation.

3. USING MATRIX FACTORIZATION FOR IMPUTATION

Matrix Factorization (MF) is arguably the most successful Collaborative Filtering (CF) technique in the area of Recommendation Systems [5, 14]. Compared with other recommendation models such as regression-based prediction models, graph-based random walk models, or simple statistical models, the MF model possesses the advantages of being more accurate and more scalable to data size. A key feature of MF models is its capability to learn *latent* factors from *relatively sparse* observations, and to leverage these factors to impute the missing elements in the matrix. In the following, we will first introduce the fundamentals of the MF methodology, present our novel sensor-data-specific modifications to the MF objective function, and finally provide the complete training procedure of the proposed method.

3.1 Introducing Matrix Factorization

As stated in section 2.3, conventional SVD is not appropriate for the task of missing data imputation. Matrix Factorization is designed to amend the limitations of SVD, and recently researchers have shown [14] that the matrix factorization model is indeed a better approach to learn the latent factors given sparse matrices, because during the factorization procedure only the observed entries are exploited. Utilizing numerical optimization procedures, for a partially observed matrix $\mathbf{R}$, MF produces two latent matrices $\mathbf{P}_{M\times K}$ and $\mathbf{Q}_{K\times N}$ whose multiplication seeks to approximate the observed entries in $\mathbf{R}_{M\times N}$:

$$\mathbf{R} \approx \mathbf{PQ}.$$

Given $\mathbf{R}$ being the sensor network readings, each row of $\mathbf{P}$ represents latent factors in the temporal dimension and each column of $\mathbf{Q}$ represents the latent factors in the dimension of correlation among sensors.

We adopt a biased-MF that includes row and column biases μ_m and μ_n. In a temperature monitoring system, the row bias can be understood as the average temperature at a given time, and the column bias reflects the average temperature at the location plus the systematic bias of the sensor node. The predictions of missing values can be obtained through $\hat{r}_{mn} = \mu_m + \mu_n + \mathbf{p}_m\mathbf{q}_n$. After adding the regularization term to constrain the scale of latent factors, the objective function of MF becomes to minimize:

$$\frac{1}{2}\sum_{m,n}(r_{mn} - \hat{r}_{mn})^2 + \frac{\beta}{2}(\sum_m(\mu_m^2 + ||\mathbf{p}_m||^2) + \sum_n(\mu_n^2 + ||\mathbf{q}_n||^2)),$$

where $\mathbf{p}_m$ are the row factors of $\mathbf{P}$ (for time m), and $\mathbf{q}_n$ are the column factors of $\mathbf{Q}$ (for sensor node n) respectively. β is the parameter that controls the strength of regularization.

3.2 Temporally-Regularized Matrix Factorization (TR-MF)

Although we map the sensor data imputation task to a CF-based recommendation task, there are indeed some major differences in the properties of the data. Normal CF or MF models assume the users (i.e., temporal dimension) are independent of each other. That is, we can randomly swap the rows in the matrix without affected the factorization outcome. However, such independence does not exist in sensor data as a sensor's signal in time t is highly dependent on that of time $t-1$. In other words, an ideal model should consider such ordering dependency, and under such a model reordering the rows would significantly affect the outcome of factorization.

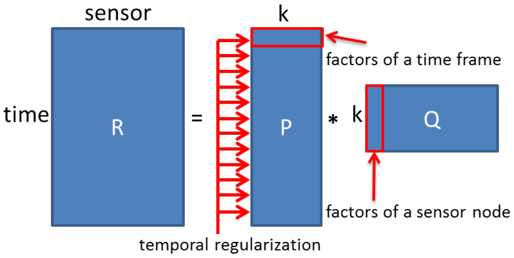


Figure 2: Illustration of TR-MF

With this observation, we propose a Temporally-Regularized Matrix Factorization (TR-MF) to better model the characteristic of sensor data. As the name may suggest, TR-MF adds a temporal regularization term to conventional MF (see Figure 2). The temporal regularization forces the latent factors of adjacent rows to be similar, which reflects the fact that readings in adjacent time steps should be similar. We also add a similar regularization term for adjacent row biases. The modified objective function (γ controls the strength of the regularization) looks like:

$$\frac{1}{2}\sum_{m,n}(r_{mn} - \hat{r}_{mn})^2 + \frac{\beta}{2}(\sum_m(\mu_m^2 + ||\mathbf{p}_m||^2) + \sum_n(\mu_n^2 + ||\mathbf{q}_n||^2))$$
$$+ \frac{\gamma}{2}\sum_m((\mu_m - \mu_{m+1})^2 + ||\mathbf{p}_m - \mathbf{p}_{m+1}||^2).$$

3.3 Spatio-Temporal-Regularized Matrix Factorization (STR-MF)

Although previously we argued that nearby sensors might not necessarily possess the highest correlation with each other, here we would like to show that our TR-MF model can easily be extended to accommodate spatial correlation if one decides to do so. Given the distance (or any kind of 'closeness' measure) between sensors, we can add spatial regularization terms to bias for possible spatial correlation. The objective function (γ_s controls the strength of the regularization) then becomes:

$$\frac{1}{2}\sum_{m,n}(r_{mn}-\hat{r}_{mn})^2+\frac{\beta}{2}\Big(\sum_{m}(\mu_m^2+||\mathbf{p}_m||^2)+\sum_{n}(\mu_n^2+||\mathbf{q}_n||^2)\Big)$$
$$+\frac{\gamma}{2}\sum_{m}((\mu_m-\mu_{m+1})^2+||\mathbf{p}_m-\mathbf{p}_{m+1}||^2)$$
$$+\frac{\gamma_s}{2}\sum_{\substack{n_i,n_j\\ \text{neighbors}}}((\mu_{n_i}-\mu_{n_j})^2+||\mathbf{p}_{n_i}-\mathbf{p}_{n_j}||^2).$$

We call this Spatio-Temporal-Regularized Matrix Factorization (STR-MF). Users should exploit spatial regularization with care, however, as our experimental study will show that biasing for spatial correlation can often produce inferior results. In the following discussion, we will focus mainly on TR-MF.

3.4 Optimization Procedure

Several methods to learn MF have been proposed, such as Stochastic Gradient Descent (SGD) [6, 14], Alternating Least Square (ALS) [14,26], Newton's method [3] and Wiberg Algorithm [21]. For sensor data, we suggest SGD for its efficiency and simplicity.

In SGD, we incrementally update our model by considering one reading at a time. Focused on one observed reading r_{mn} with the following objective function

$$\frac{1}{2}(r_{mn}-\hat{r}_{mn})^2+\frac{\beta}{2}(\mu_m^2+||\mathbf{p}_m||^2+\mu_n^2+||\mathbf{q}_n||^2).$$

It is not hard to derive the TR-MF update equations (η controls the learning rate) as

$$\begin{cases}\mu'_m=\mu_m-\eta((\hat{r}_{mn}-r_{mn})+\beta\mu_m)\\ \mu'_n=\mu_n-\eta((\hat{r}_{mn}-r_{mn})+\beta\mu_n)\\ \mathbf{p}'_m=\mathbf{p}_m-\eta((\hat{r}_{mn}-r_{mn})\mathbf{q}_n+\beta\mathbf{p}_m)\\ \mathbf{q}'_n=\mathbf{q}_n-\eta((\hat{r}_{mn}-r_{mn})\mathbf{p}_m+\beta\mathbf{q}_n)\end{cases}$$

For each reading, we update all the μ_m and $\mathbf{p}_m$. Then after a full scan of all observed readings, we perform temporal regularization by updating all μ_m and $\mathbf{p}_m$ simultaneously according to the following equations:

$$\begin{cases}\mu'_m=\mu_m-\eta\gamma((\mu_m-\mu_{m-1})+(\mu_m-\mu_{m+1}))\\ \mathbf{p}'_m=\mathbf{p}_m-\eta\gamma((\mathbf{p}_m-\mathbf{p}_{m-1})+(\mathbf{p}_m-\mathbf{p}_{m+1}))\end{cases}$$

The updating procedure is summarized in Procedure 1. Note that we propose to avoid updating the temporal regularization with the update of each reading, because doing so can bias the model toward the time steps that possess fewer missing readings, which contradicts the goal of data imputation.

```
Procedure 1 (Spatio-)Temporally-Regularized MF
Parameters: $\beta$, $\gamma$, ($\gamma_s$,) $\eta$, $K$
Input: training set, validation set
  Normalize the training set as $\mathcal{D}$
  Initialize $\mu_m$, $\mu_n$, $\mathbf{p}_m$, $\mathbf{q}_n$ to small random numbers
  repeat
      for each observed reading $r_{mn}$ in $\mathcal{D}$
          Update $\mu_m$, $\mu_n$, $\mathbf{p}_m$, $\mathbf{q}_n$
      Update $\mu_m$, $\mathbf{p}_m$ by temporal regularization
      (Update $\mu_n$, $\mathbf{q}_n$ by spatial regularization)
  until stopping criterion is met
  Output the imputation prediction model
```

For STR-MF, we can update all μ_n and $\mathbf{q_n}$ simultaneously as:

$$\begin{cases}\mu'_{n_i}=\mu_{n_i}-\eta\gamma_s\sum_{n_j}(\mu_{n_i}-\mu_{n_j})\\ \mathbf{q_{n_i}}'=\mathbf{q_{n_i}}-\eta\gamma_s\sum_{n_j}(\mathbf{q_{n_i}}-\mathbf{q_{n_j}})\end{cases}$$

where n_i and n_j are a pair of neighboring sensor nodes.

Data Normalization. Unlike the ratings in recommendation systems that are normally within a certain range (e.g., 1 to 5 stars), readings from WSNs are real-valued (rounded to a desired level of precision), and the range may vary with the sensors. Here we propose to normalize the training set to zero mean and unit variance before conducting MF learning, and once the missing values are produced by our model, we need to rescale the values to the original mean and variance. Although this procedure does not change the quality of outcomes theoretically, in practice we do find some benefits: First, when the global mean becomes zero, the origin of our model naturally becomes a fine initial point for MF training, which is very important for non-convex optimization techniques such as MF. Second, normalization forces different datasets to look similar, which simplifies the parameter tuning task.

Stopping Criterion. In addition to the training dataset used to learn the TR-MF (or STR-MF) model, we use a validation dataset to determine the model parameters and when to cease updating the model. More specifically, the training stops when the validation error fails to decrease for a certain amount of iterations (500 in our experiments).

Time Complexity. While training TR-MF, the time required for each update is $\Theta(KN)$, where K is the number of factors (≤ 54 in our experiments) and N is the number of observed readings. In other words, it is independent of the size of the data matrix. The total number of iterations varies with data quality, the missing rate and some parameters such as the learning rate and stopping criterion. In our experiments, it normally takes several minutes to train a TR-MF model on a computer with Xeon 2.53GHZ processor and 16 to 64G memory, but in some circumstances the training time can grow to 1 to 2 hours. To predict a missing value, it takes $2K$ multiplication operations, which can be done in real time. On a normal laptop, it takes less than one second to predict all values for our Berkeley data matrix ($270,000$ values). Note that the training time is not as important as the prediction time, as normally we need only to train our model once to learn the optimal parameters and factors for prediction.

4. MULTIVARIATE FACTORIZATION MODEL

A given sensor node may contain multiple sensor types and thus is capable of sensing various aspects of the environment (e.g., temperature and humidity) at the same time. These attributes can potentially be correlated [7], and being able to take advantage of such correlation would result in better imputation quality. This section proposes two models, Multivariate TR-MF and Temporally-Regularized Tensor Factorization (TR-TF), to leverage multivariate correlation for missing data recovery.

4.1 Multivariate TR-MF (MTR-MF)

In TR-MF, as we normally have many more time-steps than sensor nodes, the latent matrix $\mathbf{P}$ is much larger than $\mathbf{Q}$, and therefore being able to learn a faithful representation of latent factors in the temporal-dimension is critical.

For concreteness, assume there are two types of sensors in a node: temperature and humidity. Then, using TR-MF we can obtain $\mathbf{P}_{tem}$ and $\mathbf{Q}_{tem}$ from the temperature matrix $\mathbf{R}_{tem}$, and obtain $\mathbf{P}_{hum}$ and $\mathbf{Q}_{hum}$ from the humidity matrix $\mathbf{R}_{hum}$. These two $\mathbf{P}$s are identical in size, and it is not hard to imagine that they should be correlated because row factors $\mathbf{p}_m$ in both matrices represent the factors of time step m. Therefore, it might be beneficial to use both sides of information to learn a unified and better $\mathbf{P}$. This observation motivates us to design the Multivariate TR-MF (MTR-MF).

In MTR-TF, we let $\mathbf{R} = \begin{bmatrix}\mathbf{R}_{tem} & \mathbf{R}_{hum}\end{bmatrix}$, which is the horizontal concatenation of the temperature matrix and the humidity matrix. This enables the temperature and humidity models to share the common $\mathbf{P}$ matrix, so that they merge similar factors and communicate the observed information with each other. Note that the $\mathbf{Q}_{tem}$ and $\mathbf{Q}_{hum}$ matrices in the TR-MF model remain independent in the new $\mathbf{Q}$ in MTR-MF. Also, the spatial regularization can be freely added to yield a corresponding MSTR-MF.

The learning process of Multivariate TR-MF is very similar to that of TR-MF except for two differences: (1) for each row, we need distinct bias terms for temperature and humidity readings as they naturally are biased differently, and (2) $\mathbf{R}_{tem}$ and $\mathbf{R}_{hum}$ of $\mathbf{R}$ must be normalized independently with their own means and variances.

4.2 Temporally-Regularized Tensor Factorization (TR-TF)

One main concern for MTR-MF is that it does not fully exploit the mutual-dependency between the multiple sensor signals at a node. Namely, we did not specify that a pair of columns in $\mathbf{R}_{tem}$ and $\mathbf{R}_{hum}$ correspond to the same node. Here we propose a more complex tensor model to capture such relationships.

A tensor can be regarded as a high-dimensional matrix, and is usually exploited to represent multi-dimensional data. While TR-MF could model only 2-dimensional correlations such as sensor/time-step, a third-order tensor can model 3-dimensional correlations such as sensor1/sensor2/time-step. Tensor decomposition is a multi-dimensional extension of Singular Value Decomposition (SVD). Similar to SVD, conventional Tensor Decomposition methods assume a fully occupied matrix $\mathbf{R}$.

In the following, we will first introduce the tensor decomposition models and then describe how to modify one of them into a tensor factorization model for imputing missing data.

4.2.1 *Tensor Decomposition*

The long-standing Tucker decomposition model [24] factorizes a higher-order tensor into a core tensor $\mathbf{S}$ and one factor matrix for each dimension—however, it is computationally expensive. This led to the development of the more efficient Canonical Decomposition (CD) [4], which factorizes a tensor into a sum of K rank-one tensors. CD is the special case of the Tucker decomposition when $\mathbf{S}$ is super-diagonal. Formally, the CD of an $M \times N \times C$ tensor $\mathbf{T}$ is

$$\mathbf{T} = \sum_{k=1}^{K} \mathbf{p}_k \otimes \mathbf{q}_k \otimes \mathbf{w}_k \text{ or } t_{mnc} = \sum_{k=1}^{K} p_{mk} q_{nk} w_{ck}$$

where $\mathbf{p}_i$, $\mathbf{q}_i$ and $\mathbf{w}_i$ are a column of matrices $\mathbf{P}$, $\mathbf{Q}$ and $\mathbf{W}$, which are the factor matrices, and K is the number of columns.

4.2.2 *Tensor Factorization for Data Imputation*

Here we introduce Temporally-Regularized Tensor Factorization (TR-TF) model for missing data estimation. Similar to TR-MF, TR-TF learns the temporal correlation given the sparsity of data with the capability to take additional information such as spatial correlation and heterogeneous sensor readings into consideration.

We follow the idea of context-aware recommendation sytems [12], but we use Canonical Decomposition rather than Tucker Decomposition. Consider a 3-dimensional tensor (e.g., the temperature reading of a sensor given a certain time-stamp and a certain humidity reading), a tensor model can be described as:

$$\text{Factorization} := F_1 \times F_2 \times F_3 \to F_1 \times K, F_2 \times K, F_3 \times K$$

The tensor factorization model decomposes the three dimensional tensors T into three matrices. One of them represents the temporal dimension, and the rest can represent sensor nodes, sensor node coordinates, heterogeneous sensor readings, etc. In the experiments, we implemented a three dimensional tensor model and chose the sensor nodes as well as the multivariate sensor readings as the remaining two dimensions.

Similar to the TR-MF model, we also add bias terms to each dimension into the TR-TF model. The prediction function is:

$$\hat{t}_{mnc} = \mu_m + \mu_n + \mu_c + \sum_{k=1}^{K} p_{mk} q_{nk} w_{ck}$$

We extend the model with regularization terms designed to constrain the scale of latent factors and, solely in the temporal dimension, to force adjacent rows to be similar. The final objective function of our TR-TF is:

$$\begin{aligned}
&\sum_{m,n,c} \frac{1}{2}(\hat{t}_{mnc} - t_{mnc})^2 \\
&+\frac{\beta}{2}(\sum_m (\mu_m + \|\mathbf{p}_m\|^2) + \sum_n (\mu_n + \|\mathbf{q}_n\|^2) + \sum_c (\mu_c + \|\mathbf{w}_c\|^2)) \\
&+\frac{\gamma}{2}\sum_m ((\mu_m - \mu_{m+1})^2 + \|\mathbf{p}_m - \mathbf{p}_{m+1}\|^2)
\end{aligned}$$

4.2.3 Optimization

Minimizing the above objective function can be done using many different strategies. For efficiency and scalability, we suggest Stochastic Gradient Descent (SGD) as in TR-MF. Focusing on an observed reading data t_{mnc}, the update rules for all ks are:

$$\begin{aligned}
p'_{mk} &= p_{mk} - \eta(e \times q_{nk} \times w_{ck} + \beta \times p_{mk}) \\
q'_{nk} &= q_{nk} - \eta(e \times p_{mk} \times w_{ck} + \beta \times q_{nk}) \\
w'_{ck} &= w_{ck} - \eta(e \times p_{mk} \times q_{nk} + \beta \times w_{ck}) \\
\mu_m{}' &= \mu_m - \eta(e + \beta\mu_m) \\
\mu_n{}' &= \mu_n - \eta(e + \beta\mu_n) \\
\mu_c{}' &= \mu_c - \eta(e + \beta\mu_c)
\end{aligned}$$

where $e = \hat{t}_{mnc} - t_{mnc}$. After a round of updating, we then update the model according to temporal regularization:

$$\begin{aligned}
\mathbf{p}_m{}' &= \mathbf{p}_m - \eta\gamma((\mathbf{p}_m - \mathbf{p}_{m+1}) + (\mathbf{p}_m - \mathbf{p}_{m-1})) \\
\mu_m{}' &= \mu_m - \eta\gamma((\mu_m - \mu_{m+1}) + (\mu_m - \mu_{m-1}))
\end{aligned}$$

TR-TF applies similar normalization and stopping criterion techniques as in Section 3. Its time complexity is also similar to that of TR-MF, except TR-TF models can be trained even more efficiently because it usually consists of fewer factors and does not require many iterations to converge.

5. EXPERIMENTAL RESULTS

We conducted an experimental study comparing the accuracy of our proposed TR-MF, STR-MF, MTR-MF, and TR-TF methods to prior state-of-the-art methods.

5.1 Experimental Setup

We performed our study using two datasets: an indoor dataset, the "Berkeley" dataset, and an outdoor dataset, the "Traffic" dataset. For each dataset, we study two patterns for missing data: "random missing" and "consecutive missing." This section details these datasets and patterns, as well as the parameter settings for our methods. We follow the standard evaluation process in machine learning of dividing the data into training, validation, and testing sets.

5.1.1 Datasets

Berkeley Dataset. The Intel Berkeley Research lab dataset [2] records temperature, humidity, light, and voltage for 54 sensors (with one sensor completely missing) in an indoor lab environment from February 28th and April 5th, 2004. Sensor locations are shown in Figure 3. The dataset includes 2.3M sensor observations, with over 210K samples along the temporal dimension. Because the completeness of data degrades after 10000 samples, in our experiments only the first 2500, 5000, or 10000 samples were used. We found the relative performance of the compared methods to be similar in these three cases; thus, we report only results from 5000 samples. The inherent missing rate (i.e., before we introduce our missing value patterns) for these first 5000 samples is roughly 49%.

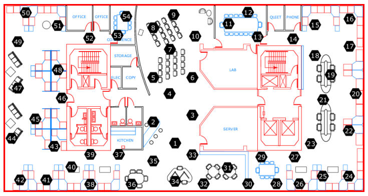


Figure 3: Intel Berkeley Research Lab Floorplan

Traffic Dataset. The Traffic Dataset [19] records the temperature, humidity, and voltage conditions of 20 sensor nodes and one gateway node. This dataset, collected by the Bio-industrial Department at National Taiwan

University, was recorded over a 2.5 year time period ending in 2011 in an outdoor location high traffic area in Taipei, Taiwan (see Figure 4). The sampling rate is roughly once every 30 minutes. Along the temporal dimension, we use the entire range which consists of roughly 43K time stamps. The inherent missing rate is roughly 58%.

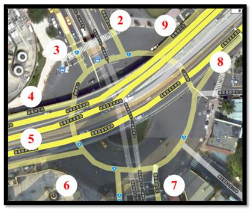


Figure 4: Traffic Sensor Deployment Configuration (8 sensors of 20 shown)

Note that for both Berkeley and Traffic datasets, simple preprocessing rules were applied to remove apparent outliers. In the Berkeley dataset, observations are removed if temperature $> 100\,°\mathrm{C}$, temperature $< 5\,°\mathrm{C}$, or humidity $< 16\%$. In the Traffic dataset, observations are removed if temperature $> 60\,°\mathrm{C}$, temperature $< 5\,°\mathrm{C}$, humidity $> 100\%$ or humidity $< 10\%$.

5.1.2 Missing Data Generation

Although both datasets intrinsically have missing readings, we cannot use those for evaluation because their true values are unknown. Instead we devised two strategies to produce artificial missing data.

Random Missing Pattern. This pattern reflects repeatedly choosing a random time and random sensor to be missing and hence removed from the training set. We define two variables x and y during our experiment, and the X-axis of the resulting plot varies with x.

- 10% of the existing readings are randomly selected (without replacement) to be the validation set
- y% of the existing readings are randomly selected (without replacement) to be the testing set
- The remaining readings (x%) are part of the training set. That is, x+y+10=100 that accounts for all the observed readings.

Consecutive Missing Pattern. This pattern reflects testing the effect of all data missing after a certain point in time. We define two values x and y as follows.

- Here, we have the last y% of time covered as missing, and the prior 10% to that is considered as the validation data.
- The sensor node numbers "covered up" in the validation and testing for the Berkeley and Traffic datasets are $4, 19, 45$ and $2, 4, 6, 8, 10, 14, 17, 19, 20, 21$, respectively. Note that node 21 of the Traffic dataset is the gateway node. We experimented with other combinations of covered up nodes, and the results were similar.

5.1.3 Parameter Setting

We exploit the commonly used metric of root-mean-square-error (RMSE) to measure the difference between the predicted values and the ground truth, and all parameters in our models and competitors' models are carefully optimized using the validation set.

Here are the resulting parameter values for our models. The number of factors K is 54 for MF and 30 for TF for the Berkeley dataset, and 21 for MF and 10 for TF for the Traffic dataset. The learning rate η is set to 0.04 to 0.004 for MF and 0.001 to 0.0001 for TF. A smaller η or a larger K could slow down the training process, but it would not degrade the model. Also, a reasonable choice of K should not be larger than the number of sensor nodes since the rank of the matrix is at most K. The temporal regularization γ is set to $0.2/\eta$ in all of our experiments. The conventional regularization β is 0.001 for consecutive missing and 0 for random missing in MF, while in TF, β is 0.005 for consecutive missing and 0.001 for random missing.

5.2 Experimental Results on TR-MF

We first compared TR-MF with its predecessor, the EOF model [1] and improved EOF model [13]. Table 2 shows a representative experiment (Berkeley dataset, random missing pattern, humidity readings). Improved EOF is significantly more accurate than EOF, but still worse than our TR-MF. Given this, and the fact that EOF-based methods require the execution of many rounds of SVD, which is computational expensive and much less efficient than the other models we study, we omit EOF from any further experiments.

Table 2: RMSE of TR-MF and EOF on (Berkeley, Random,Humid). Columns are labeled with the percentage of data used for training.

	10%	20%	40%	60%	80%	85%
TR-MF	0.142	0.114	0.092	0.082	0.076	0.075
EOF	2.423	1.385	1.000	0.734	0.656	0.645
Improved EOF	0.362	0.237	0.183	0.152	0.136	0.134

We compared our TR-MF with conventional MF (i.e., MF without temporal regularization), Linear Interpolation (LI), Applying K-nearest neighbor Estimation (AKE), Data Estimation using Statistical Model (DESM), Spatial Temporal Imputation (STI) and Multiple Imputation (MI). Note that sensor location information is avail-

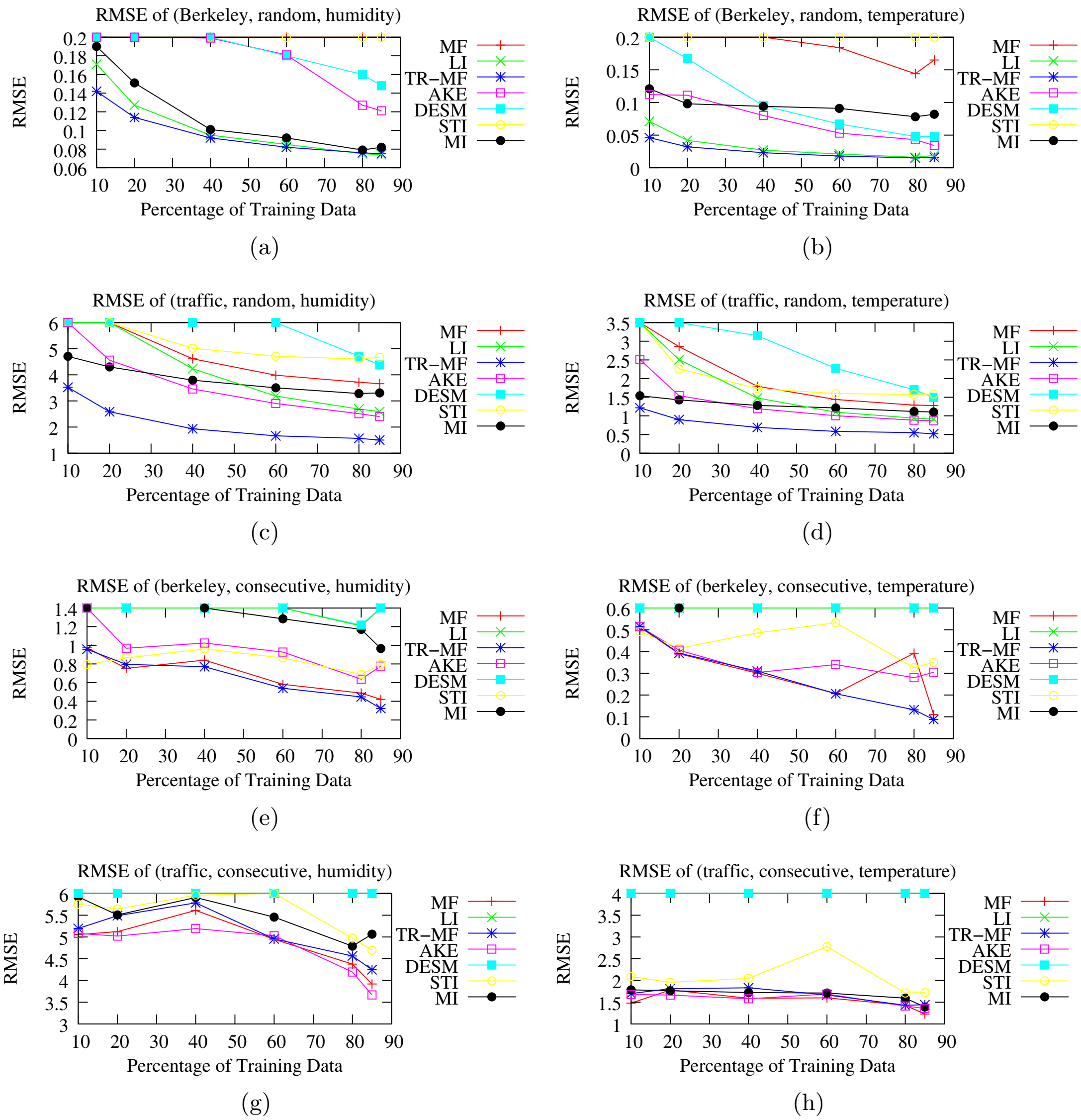


Figure 5: Accuracy of salient methods, varying the percentage of training data. (a) to (d) are for Random Missing data; (e) to (h) are for Consecutive Missing data.

able for only 8 out of 20 sensors in the Traffic dataset, but the pairwise distance values are required inputs for the DESM and STI models. In these cases, we simply assume the unknown distances are equal. Also, we omit the results for light sensors in the Berkeley dataset (in which TR-MF performs equally well as in other cases) due to space limitations.

Figures 5(a), 5(b), 5(c) and 5(d) show the results of our random missing experiments. The outcomes indicate several interesting facts. First, TR-MF outperforms the original MF significantly, which demonstrates the effectiveness of the temporal regularization. In general, TR-MF shows significant improvement over all the other methods. On the other hand, LI is quite competitive on the Berkeley dataset, especially for lower missing rate cases. This is because the sampling rate of the Berkeley dataset is fairly high, using only temporal correlations is sufficient to obtain decent results. In datasets with lower sampling rates such as the traffic dataset, the spatial-oriented methods such as AKE outperform LI.

Figures 5(e), 5(f) and 5(g) and 5(h) show the results for the consecutive missing pattern. Comparing it with the previous figure, we find that, not surprisingly, the consecutive missing task is more challenging, as the RMSE is much higher than that of the random missing case. Generally speaking, the incorporation of the temporal correlation is not very useful for the consecutive missing cases. Specifically, LI is no longer competitive and the performance between TR-MF and MF has become closer, especially when the missing rate is high. The AKE algorithm also performs competitively in this case. The results indicate that when there is less information from which a model can learn, providing it other kinds of information (e.g., the spatial information) can potentially improve the outcome. This conjecture is confirmed in the next experiment which shows that TR-MF can be further improved in consecutive missing cases when spatial information is included.

5.3 Experimental Results on STR-MF

Next we focus on using the Berkeley dataset to verify the STR-MF model as the sensor node location information for the Traffic data is incompletely specified and thus unable to be properly exploited.

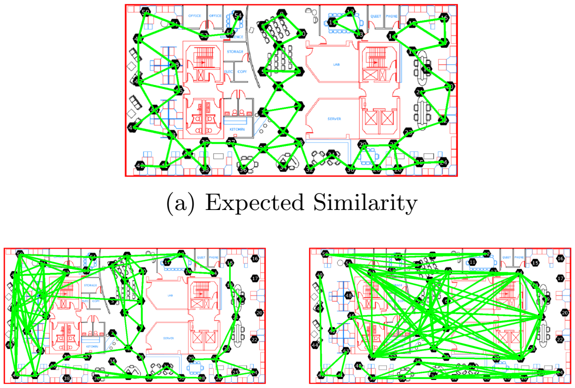
Table 3: RMSE of Berkeley, Random Missing

	Humidity			Light			Temperature		
train	TR-MF	STR-MF	sTR-MF	TR-MF	STR-MF	sTR-MF	TR-MF	STR-MF	sTR-MF
10%	**0.142**	0.484	0.173	**35.5**	97.4	38.3	**0.046**	0.154	0.061
20%	**0.114**	0.424	0.135	**28.2**	90.6	28.9	**0.032**	0.146	0.047
40%	**0.092**	0.352	0.104	**21.2**	85.8	22.8	**0.023**	0.145	0.037
60%	**0.082**	0.337	0.093	**17.2**	83.3	18.3	**0.018**	0.147	0.031
80%	**0.076**	0.324	0.084	**17.7**	84.4	18.1	**0.015**	0.148	0.027
85%	**0.075**	0.326	0.083	**14.4**	82.0	15.8	**0.016**	0.138	0.028

Table 4: RMSE of Berkeley, Consecutive Missing

	Humidity			Light			Temperature		
train	TR-MF	STR-MF	sTR-MF	TR-MF	STR-MF	sTR-MF	TR-MF	STR-MF	sTR-MF
10%	0.957	0.573	**0.547**	**220.0**	281.6	264.7	0.515	**0.242**	0.307
20%	0.796	0.657	**0.459**	**113.3**	236.8	230.0	0.392	**0.179**	0.187
40%	0.771	0.520	**0.455**	**58.0**	110.7	64.3	0.310	0.196	**0.189**
60%	0.540	**0.351**	0.708	**41.7**	150.1	69.2	0.206	**0.191**	0.243
80%	0.447	0.299	**0.261**	**21.4**	112.6	28.0	0.132	**0.108**	0.114
85%	0.323	**0.166**	0.256	**8.3**	85.4	12.0	0.088	**0.065**	0.082

Tables 3 and 4 compare STR-MF with TR-MF. The spatial regularization used for STR-MF was defined based on manually-determined neighborhood nodes established via inspection of the floor plan, as shown in Figure 6(a). Note that STR-MF and sTR-MF stand for TR-MF with strong and relatively weaker spatial regularization, respectively. The results show that for the random missing pattern, adding spatial regularization indeed hurts the performance. We believe this perhaps surprising finding holds for the random missing cases because the sensor observations alone are adequate for TR-MF to learn the correlations between sensors, hence enforcing similarity between nearby sensors can actually degrade accuracy because the degree of correlation between sensors is actually quiet low. On the other hand, for consecutive missing cases, adding spatial regularization shows significant improvement for humidity and temperature readings. Generally, the distinction between TR-MF and STR-MF is that the former is a pure data-driven model while the latter is slightly knowledge-driven. The experiments show that when we have sufficient data to learn the correlation between sensors, it is better to ignore the given knowledge because they might not be accurate; however, if the given data is less abundant, incorporating external spatial knowledge can be helpful.



(a) Expected Similarity

(b) Humidity: top 100 similar pairs

(c) Light: top 100 similar pairs

Figure 6: Expected vs. Actual Similarities

We also conduct an interesting experiment to see whether the correlations learned by TR-MF in random missing cases reflect our hypothesis of spatial correlation (i.e., Figure 6(a)). For the humidity and light data, we first use the TR-MF model to learn the latent factors of each sensor node given random missing under 85% training data. Based on the latent factors $\mathbf{q}$, we can then define the similarity between sensor nodes n_i and n_j as: $\frac{q_{n_i}^T q_{n_j}}{max(||q_{n_i}||^2, ||q_{n_j}||^2)}$,

Then we identify the top 100 similar pairs and draw a line between each of them on the floor plan. We can see that the manually hypothesized spatial correlation plot Figure 6(a) is more similar to the one learned from humidity data (Figure 6(b)) than to the one learned from light data (Figure 6(c)). This experiment shows two interesting insights. First, the manually-crafted spatial correlations do not necessary reflect the true correlations between sensor signals, because they might have not yet considered other factors such as barriers or long-distance correlations. Second, if the human knowledge of spatial correlation does to some extent reflect the true correlations between sensors, incorporating them in scenarios with limited information (i.e., the consecutive missing pattens) can be helpful (e.g., the humidity imputation in Table 4 improves with spatial information); otherwise it is useless (e.g., no improvement for the light imputation in Table 4).

5.4 Experimental Results on Multivariate Imputation Methods

To evaluate multivariate imputation models, in the experiments we allow temperature and humidity information to mutually enhance the predictions. Here we compare the two proposed multivariate imputation models: MTR-MF and TR-TF with the univariate model TR-MF, which has been demonstrated to outperform other models in the previous section. Due to space limitations, we report results for only the Traffic dataset.

There are two different scenarios we are concerned with in this experiment. For illustration purposes, assume the goal is to predict the missing entries in the temperature matrix. In the first scenario (denoted as MTR-MF-all and TR-TF-all), we do not cover up the humidity data and use this data in its entirety. The goal here is to evaluate whether using all the humidity information allows our model to improve the temperature predictions. In the second scenario (denoted as MTR-MF-part and TR-TF-part), we assume the humidity readings are missing together whenever temperature readings are missing. Note that such cases happen quite often in WSN due to communication loss or sensor node malfunction. In the second scenario, MTR-MF-part can be performed without problems. However, to predict an entry in the temperature matrix, the TR-TF model requires use of the corresponding entry in the humidity matrix. In the TR-TF-part, such issues can be addressed by first predicting the missing humidity readings based on the humidity TR-MF model, and then apply TR-TF.

The results are shown in Figure 7. Except MTR-MF-part, which remains similar performance as TR-MF, we see significant improvement on TR-TF-all, TR-TF-part and MTR-MF-all. From this we see the effect of heterogeneous information and we can conclude that: although MTR-MF is not helpful when multiple sensors are missing together, both of the temperature and humidity information can indeed be used to enhance the prediction of the other.

5.5 Designing Prediction Models after Imputation

As stated previously, one main reason to conduct data imputation is that most off-the-shelf data analysis tools cannot deal with inputs with missing values. Here one practical question to ask is: "Can an improved imputation outcome indeed lead to the development of a better analysis model?" Here we conduct an experiment using the *humidity* values of 20 sensors in the traffic dataset to build two regression models (linear regression and support-vector regression) in order to predict the *temperature* values of the gateway.

In the experiment, we first remove the gateway signal from the matrix, and use TR-MF as well as the other competitive models (including LI and AKE) to fill in all the missing humidity values in the remaining 20 sensors. Then for each imputation model, we use the filled-in readings of these 20 sensors as the inputs X and the gateway temperature values as the output Y to train the regression models for prediction. We divide the data randomly into 80% training and 20% testing, and show the results of both regression models in Table 5. We use Mean Square Error (MSE) to measure the error here and show the 95% confidence interval. The results confirm our hypothesis that a better imputation model does lead to the production of a better prediction models, and TR-MF again outperforms the other models in this respect.

Table 5: Accuracy of Temperature Models Built Using Filled Humidity Data

MSE	Linear Regression			Support Vector Regression		
train	LI	AKE	TR-MF	LI	AKE	TR-MF
10%	24.7 ±5.3	14.5 ±7.8	9.0 ±5.7	25.1 ±5.9	16.8 ±7.4	9.9 ±5.9
20%	19.2 ±1.1	13.9 ±3.1	8.7 ±1.9	20.7 ±1.5	15.8 ±2.4	7.9 ±2.2
40%	17.5 ±1.9	10.0 ±2.2	8.8 ±1.7	18.0 ±1.6	10.5 ±2.2	8.3 ±1.9
60%	16.6 ±0.6	8.0 ±1.3	6.6 ±0.8	17.4 ±0.7	9.4 ±1.0	7.2 ±0.9
80%	15.0 ±0.7	8.3 ±1.4	6.4 ±1.0	13.8 ±0.6	8.4 ±1.1	7.2 ±0.8
85%	14.4 ±0.6	7.8 ±1.3	6.3 ±0.7	16.1 ±0.6	8.4 ±0.9	5.8 ±0.8

6. CONCLUSION

This paper proposes usage of factorization-based models for missing data estimation. In contrast to many existing knowledge-driven approaches that make stronger assumptions about the data (e.g., assume that nearby sensor nodes have higher similarity; or assume the so-called "neighborhood area" is at the same radius away from the center in every direction), our data-driven factorization model learns the inter-sensor and intra-sensor correlations through exploiting their latent similarity. Furthermore, we show that the additional knowledge such as the spatial relationships among sensors can seamlessly be incorporated into our model through regularization terms (e.g., STR-MF) if desired.

Our experiments suggested that the temporal regularization is very helpful in general, while spatial information is useful only when the existing data are insufficient for the model to learn the inter-sensor relationships. Finally, we propose an update time $\Theta(KN)$ multivariate tensor factorization model whose training complexity does not grow with its order, which allows the users to add arbitrary more dimensions or features into the imputation model without significantly increasing the computation burden.

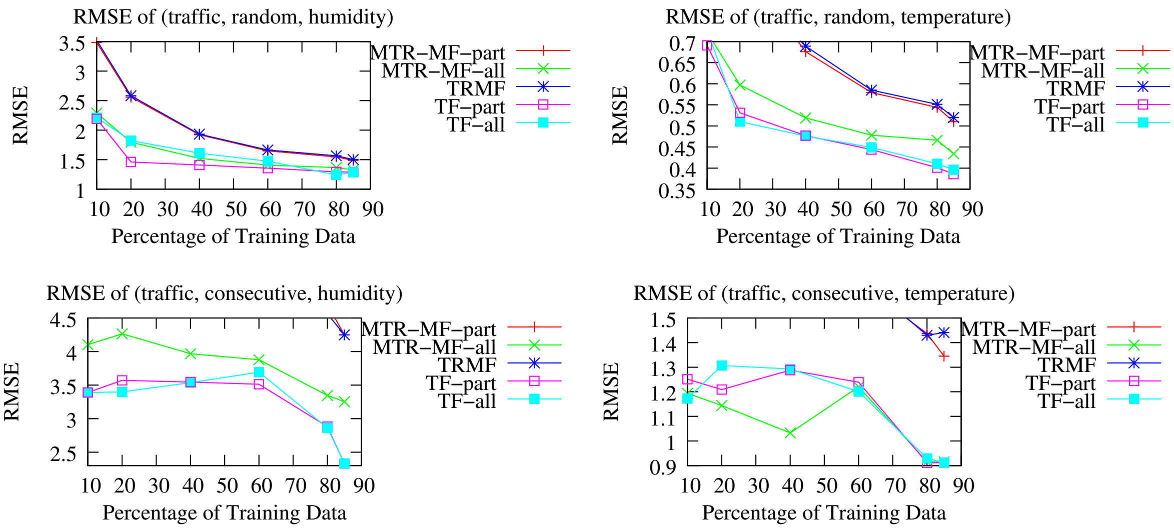


Figure 7: Accuracy of Multivariate models, varying the percentage of training data

We believe that factorization-based models will become a very important class of missing data estimation techniques for WSN in the near future, and we envision that our work can establish a foundation for more advanced research in this direction.

7. REFERENCES

[1] J. Beckers and M. Rixen. Eof calculations and data filling from incomplete oceanographic datasets. *Journal of Atmospheric and Oceanic Technology*, 2003.

[2] P. Bodik, W. Hong, C. Guestrin, S. Madden, M. Paskin, and R. Thibaux. Intel berkeley research lab dataset. `http://db.csail.mit.edu/labdata/labdata.html`, 2004.

[3] A. Buchanan and A. Fitzgibbon. Damped newton algorithms for matrix factorization with missing data. In *CVPR*, 2005.

[4] J. Carroll and J. Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of eckart-young decomposition. *Psychometrika*, 1970.

[5] P. Chen, C. Tsai, Y. Chen, K. Chou, C. Li, C. Tsai, K. Wu, Y. Chou, C. Li, W. Lin, et al. A linear ensemble of individual and blended models for music rating prediction. In *KDDCup Workshop*, 2011.

[6] M. Chih-Chao. Large-scale collaborative filtering algorithms. Master's thesis, National Taiwan University, 2008.

[7] A. Deshpande, C. Guestrin, W. Hong, and S. Madden. Exploiting correlated attributes in acquisitional query processing. In *ICDE*, 2005.

[8] P. García-Laencina, J. Sancho-Gómez, A. Figueiras-Vidal, and M. Verleysen. K nearest neighbours with mutual information for simultaneous classification and missing data imputation. *Neurocomputing*, 2009.

[9] E. Granger, M. Rubin, S. Grossberg, and P. Lavoie. Classification of incomplete data using the fuzzy artmap neural network. *IJCNN*, 2000.

[10] L. Gruenwald, H. Chok, and M. Aboukhamis. Using data mining to estimate missing sensor data. In *ICDM Workshops*, 2007.

[11] P. Jian-Zhong and L. Ji-Zhou. A temporal and spatial correlation based missing values imputation algorithm in wireless sensor networks. *Chinese J. of Computers*, 2010.

[12] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *RecSys*, 2010.

[13] D. Kondrashov, M. Ghil, et al. Spatio-temporal filling of missing points in geophysical data sets. *Nonlinear Processes in Geophysics*, 2006.

[14] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 2009.

[15] M. Le Gruenwald. Estimating missing values in related sensor data streams. *COMAD*, 2005.

[16] Y. Li, C. Ai, W. Deshmukh, and Y. Wu. Data estimation in sensor networks using physical and statistical methodologies. In *ICDCS*, 2008.

[17] Y. Li and L. Parker. A spatial-temporal imputation technique for classification with missing data in a wireless sensor network. In *IROS*, 2008.

[18] J. Lim and C. Bleakley. Robust data collection and lifetime improvement in wireless sensor networks through data imputation. In *ICSNC*, 2010.

[19] J. Liu, Y. Chen, T. Lin, D. Lai, T. Wen, C. Sun, J. Juang, and J. Jiang. Developed urban air quality monitoring system based on wireless sensor networks. In *ICST*, 2011.

[20] S. Madden, M. Franklin, J. Hellerstein, and W. Hong. Tinydb: An acquisitional query processing system for sensor networks. *TODS*, 2005.

[21] T. Okatani and K. Deguchi. On the wiberg algorithm for matrix factorization in the presence of missing components. *IJCV*, 2007.

[22] M. Osborne, S. Roberts, A. Rogers, and N. Jennings. Real-time information processing of environmental sensor network data. *ACM Trans. on Sensor Networks*, 2012.

[23] L. Pan and J. Li. K-nearest neighbor based missing data estimation algorithm in wireless sensor networks. *Wireless Sensor Network*, 2010.

[24] L. Tucker. Implications of factor analysis of three-way matrices for measurement of change. *Problems in measuring change*, 1963.

[25] Y. Yuan. Multiple imputation for missing data: concepts and new development (version 9.0). In *25th SAS® Users Group International Conference*, 2000.

[26] Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan. Large-scale parallel collaborative filtering for the netflix prize. *AAIM*, 2008.