Spatially Regularized Streaming Sensor Selection

Changsheng Li[†], **Fan Wei**[‡], **Weishan Dong**[†], **Xiangfeng Wang**[§],
Junchi Yan[§*], **Xiaobin Zhu**[¶], **Qingshan Liu**[♯], **Xin Zhang**[†]
[†]IBM Research-China, China, {lcsheng,dongweis,zxin}@cn.ibm.com
[‡]Department of Mathematics, Stanford University, USA, fanwei@stanford.edu
[§]East China Normal University, China, {xfwang, jcyan}@sei.ecnu.edu.cn
[¶]Beijing Technology and Business University, China, brucezhucas@gmail.com
[♯]Nanjing University of Information Science and Technology, China, qsliu@nuist.edu.cn

Abstract

Sensor selection has become an active topic aimed at energy saving, information overload prevention, and communication cost planning in sensor networks. In many real applications, often the sensors' observation regions have overlaps and thus the sensor network is inherently redundant. Therefore it is important to select proper sensors to avoid data redundancy. This paper focuses on how to incrementally select a subset of sensors in a streaming scenario to minimize information redundancy, and meanwhile meet the power consumption constraint. We propose to perform sensor selection in a multivariate interpolation framework, such that the data sampled by the selected sensors can well predict those of the inactive sensors. Importantly, we incorporate sensors' spatial information as two regularizers, which leads to significantly better prediction performance. We also define a statistical variable to store sufficient information for incremental learning, and introduce a forgetting factor to track sensor streams' evolvement. Experiments on both synthetic and real datasets validate the effectiveness of the proposed method. Moreover, our method is over 10 times faster than the state-of-the-art sensor selection algorithm.

1. Introduction

In recent years, sensor networks have become very popular for collecting continuous data streams, such as audio and visual data. In many real-world scenarios, a sensor network may consist of a large number of sensor nodes; The large quantity of sensors may cause bottlenecks in the areas such as battery supplies, communication, and information overload (Kollios et al. 2005; Zhang and Ji 2005). In order to overcome these limitations, sensor selection becomes a promising and effective way to reduce the number of sensor nodes used. Up to now, sensor selection has been widely studied for various applications, such as traffic flow forecasting (Chan et al. 2012), robotics (Hovland and McCarragher 1997), target tracking (Wang et al. 2004), wireless sensor networks (Abrams, Goel, and Plotkin 2004), and sensor network management (Rowaihy et al. 2007). However, many of the approaches above are based on a simplified assumption that the "sensors can perfectly observe a particular sensing region, and nothing outside the region" (Golovin, Faulkner, and Krause 2010). By this assumption, the data collected would have no redundancy. However, in many real-life applications, the sensor network is redundant, i.e., a small fraction of the sensor nodes in the whole network are necessary to collect the data and describe the underlying phenomenon. An example of redundant sensor network is that different cameras are installed in a room monitoring people's activities. In order to stress sensor selection in the real-life scenarios, we do not make the assumption that the sensor network is not redundant. On the contrary, we follow the method (Aggarwal, Xie, and Yu 2011) to use the assumption that the sensor network is inherently redundant. In such a case, the goal of sensor selection is to select a suitable subset of sensors so as to minimize the loss of information (e.g., prediction errors of inactive sensors), subject to the battery power consumption constraint.

In a redundant sensor network, data streams collected by different sensor nodes usually have predictable relationships, and such relationships can be used to determine which subset of sensors are necessary and thus be in active mode (Aggarwal, Bar-Noy, and Shamoun 2011). Recently, the distributed online greedy (DOG) method proposed in (Golovin, Faulkner, and Krause 2010) aims to take advantage of utility-feedback to repeatedly select sensors online. The method in (Aggarwal, Bar-Noy, and Shamoun 2011) uses external domain-specific linkage knowledge for sensor selection. However, such linkage information is often hard to acquire beforehand in practice. The key idea of the technique in (Aggarwal, Xie, and Yu 2011) is to use regression analysis to determine an active sensor subset with minimal power consumption. This small active sensor set is used to predict the observation values of the other inactive sensors.

In the methods above, the most relevant previous work to this paper is (Aggarwal, Xie, and Yu 2011), where the authors address a similar problem for sensor selection. However, their method has the following limitations. i) The connection between sensor selection and multi-variate interpolation is not fully utilized in their method. But these two parts are actually interleaved and can benefit from each other, so it is natural to incorporate multi-variate interpolation into not only prediction, but also sensor selection. ii) The location information of sensor nodes is not considered in their sensor selection method. However, sensors with smaller distances usually collect more similar data, which provides additional information that can be exploited. iii) The time window based strategy when handling streaming data may lose too much historical information. A method with no information loss is desired in such streaming applications.

In this paper, we propose an efficient algorithm, called *Spatially Regularized Streaming Sensor Selection* (SRSSS), which is an effort to bridge the above gaps. The main contributions of this paper are:

- We propose a framework to realize sensor selection in virtue of multi-variate interpolation, which can help obtain an informative sensor subset to better predict inactive sensors.
- We leverage spatial information for sensor selection. Our motivation is based on an observation that geographically closer sensors tend to generate more redundant information. Thus, we intuitively should select sensors well-separated geographically to reduce redundancy. Meanwhile, when predicting inactive sensors, the nearby active sensors in their neighborhood have higher weights than the distant ones. To the best of our knowledge, this is the first work to incorporate spatial information into sensor selection to accurately predict inactive sensors.
- We enable SRSSS to select sensors in an incremental fashion by defining a statistical variable, which stores sufficient information of historical observation with no information loss. In addition, we introduce a time-forgetting factor such that more recent data have higher weights in data streams; thus our algorithm also responds fast if the underlying distribution changes.
- We design an efficient algorithm to optimize the proposed objective function. Extensive experiments demonstrate the effectiveness and efficiency of our approach.

2. Proposed Method

Notations. In this paper, matrices are written as boldface uppercase letters and vectors as boldface lowercase letters. Given a matrix $\mathbf{A}$, we denote its (i, j)-th entry, i-th row, j-th column as A_{ij} , $\mathbf{A}^i$, and $\mathbf{A}_j$, respectively.

2.1 Problem Formulation

We formulate the sensor selection problem as follows. The sensors work in two types of modes: the *power-efficient* mode (or equivalently, *inactive* mode) and the *active* mode. In the power-efficient mode, the sensors do not collect data, which is battery power-efficient. In the active mode, the sensors use a higher sampling rate to collect data, which is expensive in terms of battery power consumption and storage costs. Therefore, although in the active mode sensors can collect the most accurate data for decision-making, it is not desirable to keep all the sensors in active mode. We further assume that the sensors can switch between the two modes at any time. The question we are interested in is the follows: how to select a subset of sensors to be on power-efficient (or inactive) mode and the rest on active mode so that we can save energy as well as not losing much information compared to when all sensors in active mode.

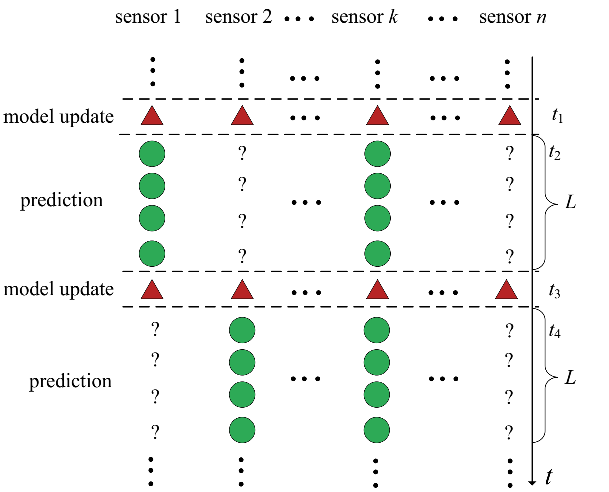


Figure 1: Schematic illustration of algorithm formulation. Time flow goes vertically; same horizontal level denotes the same instant time. Algorithm alternates between *model update* and *prediction*. A green circle means a data collected from an active sensor; Red triangles mean all the sensors are collecting the data. A question mark denotes that at that moment, the sensor is in power-efficient mode and is not collecting data; and we need to predict what the data are supposed to be from the data collected by the active sensors.

To make the problem formulation clearer, we use Figure 1 to illustrate. Our algorithm repeatedly alternates between two phases: the *model update* phase and the *prediction* phase. In the model update phase, all the sensors are collecting data. In the prediction phase, only a subset of sensors, the active sensors, are collecting data, and the data of inactive sensors are predicted from the data collected by the active sensors. At time t_1, the algorithm is under model update phase. Each sensor collects one data sample. We use these data and the previous data collected in model update phase to determine which subset of sensors to be turned on in the coming model prediction phase. For example the algorithm decides to turn sensor 1 and sensor k on active mode and the rest in inactive mode. At this phase the algorithm also learns a mapping relation between output from sensor 1 and sensor k to the ones from the rest of the sensors. This phase costs 1 unit of time. Algorithm now immediately enters the prediction phase. At time $t_2 = t_1 + 1$, we use the active sensors (e.g., sensor 1 and sensor k) to collect data and the inactive sensors are not collecting data. We can predict the data of the inactive sensors based on the output from active sensors and the mapping relation we learned at time t_1. This phase lasts for L units of time. Our model switches back to model update phase immediately at time $t_3 = t_2+L$, all the sensors start to collect the data again. Again by the mapping relation we are able to see how far our approximation differs from the true values. The model update phase redetermines which subset of sensors to be in the active and relearns the mapping relationship from the current and previous data collected in the model update phase. This phase again costs 1 unit of time. And suppose now sensors 2 and

k are selected as active sensors. We enter the next phase of prediction at time $t_4 = t_3 + 1$; the new active sensors (e.g., sensors 2 and k) frequently sample data, and output from new power-efficient sensors are predicted based on the new active sensors. This phase will again last for L units of time. And our model switches back to model update phase. The process continues.

Mathematically, let $\Phi = \{1, 2, \dots, n\}$ be the index set of the sensor nodes. Each prediction mode takes L units of time and each model update phase takes 1 unit of time. Suppose that the model update phase starts at time $kL, k = 0, 1, \dots$. And the prediction phase starts at time $kL + 1$ and ends at $kL + (L - 1)$. Let $\mathbf{X}_k = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \in \mathbb{R}^{(k+1)\times n}$ denote the data streams collected by all the sensors in the model update phase up to time kL; reader can think of k as the number of rounds k our model has gone through model update phase. Based on $\mathbf{X}_k$, our goal is to find a sensor subset $\phi_k = \{l_1, \dots, l_s\} \subset \Phi$ to be active, where s is also learned, and online learn a project matrix $\mathbf{W}_k$ mapping from $\mathbf{X}_k^{\phi} = \{\mathbf{x}_{l_1}, \dots, \mathbf{x}_{l_s}\}$ to $\mathbf{X}_k^{\Delta} = \mathbf{X}_t \setminus \mathbf{X}_k^{\phi}$ [1], subject to the power consumption constraint. When the sensors ϕ_k are working actively, we expect that the data of the inactive sensors $\varphi_k = \Phi \setminus \phi_k$ can be predicted as accurately as possible based on $\mathbf{W}_k$ and the future data collected by ϕ_k. Based on all the data collected in the next round of model update, i.e., at time $(k+1)L$, we will incrementally choose a new subset ϕ_{k+1} to be active. In a word, we will design an incremental algorithm to perform sensor selection.

2.2 Objective Function

We aim to realize sensor selection in a multi-variate interpolation framework. In the meantime, we hope that the selected active sensors are not too close to each other, thus the data streams are with less redundancy. In addition, for each power-efficient sensor, we claim that its data output should rely more on the active sensors closer to it. In light of these points, we formulate our objective function for model update as follows:

$$\begin{aligned}
&(\mathbf{W}_{k+1}, \mathbf{z}_{k+1}) = \\
&\arg\min_{\mathbf{W},\mathbf{z}} \sum_{i=1}^{k} \mu^{k-i} \|\mathbf{X}_k^i \mathbf{D}_\mathbf{z} \mathbf{W}(\mathbf{I} - \mathbf{D}_\mathbf{z}) - \mathbf{X}_k^i (\mathbf{I} - \mathbf{D}_\mathbf{z})\|_2^2 \\
&+ \alpha \sum_{i,j=1}^{n} \|\mathbf{y}_i - \mathbf{y}_j\|_2 |W_{ij}| - \beta \sum_{i,j=1}^{n} \|\mathbf{y}_i - \mathbf{y}_j\|_2 z_i z_j + \lambda \|\mathbf{W}\|_F^2 \\
&s.t.\ \mathbf{z} = [z_1, \dots, z_n] \in \{0, 1\}^n, \mathbf{c}^T \mathbf{z} \le P \qquad (1)
\end{aligned}$$

where $\mathbf{X}_k^i$ denotes the i-th row of matrix $\mathbf{X}_k$, i.e., the observation values of the i-th tick collected by all the sensors. $\mathbf{W} \in \mathbb{R}^{n\times n}$ is the project matrix to be learned. W_{ij} denotes the (i, j)-the element of $\mathbf{W}$. Let $\mathbf{z} = [z_1, \dots, z_n] \in \mathbb{R}^n$ be an indicator vector to represent which sensors are selected to be active. $z_i = 1$ (or 0) indicates that the i-th sensor is selected as active (or not). $\mathbf{D}_\mathbf{z}$ is a diagonal matrix with $(\mathbf{D}_\mathbf{z})_{ii} = z_i$. Let $\mathbf{y}_i$ be the location coordinates of the i-th sensor, and $\mathbf{I}$ is the identity matrix. $\|\cdot\|_2$ denotes the l_2 norm of a vector, and $\|\cdot\|_F$ denotes the Frobenius norm of a matrix. $\mathbf{c} = [c_1, \dots, c_n]^T$ are the power costs required to activate the corresponding sensors. $\mu \in [0, 1]$ is a time-forgetting factor to favor more recent data. $\alpha, \beta, \lambda \ge 0$, are three non-negative regularization parameters.

In the first term of the objective function (1), $\mathbf{X}_k^i \mathbf{D}_\mathbf{z}$ turns the values of $\mathbf{X}_k^i$ to be zeros when the corresponding sensors are not active. $\mathbf{X}_k^i \mathbf{D}_\mathbf{z} \mathbf{W}$ predicts data from all sensors based on the ones from active sensors, while $\mathbf{X}_k^i \mathbf{D}_\mathbf{z} \mathbf{W}(\mathbf{I} - \mathbf{D}_\mathbf{z})$ considers only predictions for inactive sensors by putting zero predictions for active sensors. Hence the goal of the first term in (1) is to minimize the prediction loss on the inactive sensors. The following two terms incorporate spatial information. The second term in (1) aims at penalizing the coefficients W_{ij} with large distances between sensors, so that the prediction relies more on local information from the closer active sensors. The third term maximizes the distances among the selected active sensor nodes, in order to make data streams collected by the active sensors less redundant. The last term in (1) controls the complexity of the learned projection matrix $\mathbf{W}$. The inequality constraint in (1) is to restrict the battery power consumption of sensors to meet the energy budget requirement.

However, (1) is a quartic problem on variable $\mathbf{z}$, which is computational expensive.

Theorem 2.1: *The objective function (1) is equivalent to*

$$\begin{aligned}
(\mathbf{W}_{k+1}, \mathbf{z}_{k+1}) = \arg\min_{\mathbf{W},\mathbf{z}} \sum_{i=1}^{k} \mu^{k-i} \|\mathbf{X}_k^i \mathbf{D}_\mathbf{z} \mathbf{W} - \mathbf{X}_k^i (\mathbf{I} - \mathbf{D}_\mathbf{z})\|_2^2 \\
+ \alpha \sum_{i,j=1}^{n} \|\mathbf{y}_i - \mathbf{y}_j\|_2 |W_{ij}| - \beta \sum_{i,j=1}^{n} \|\mathbf{y}_i - \mathbf{y}_j\|_2 z_i z_j + \lambda \|\mathbf{W}\|_F^2 \\
s.t.\ \mathbf{z} = [z_1, \dots, z_n] \in \{0, 1\}^n, \mathbf{c}^T \mathbf{z} \le P, \qquad (2)
\end{aligned}$$

Problem (2) now is quadratic about $\mathbf{z}$. The complete proof of Theorem 2.1 is provided in Appendix.

Since in the context of data streams for sensor network, it is impractical to load all the historical data into memory or scan a sample multiple times for model update. Therefore, we introduce a statistics to store sufficient information of the historical data with respect to our objective function, without storing the individual data. It can be regarded as a way of information compression, and there is no information loss with respect to (1) (Li et al. 2015). Notice that only the first term in the objective function (2) involves historical data $\mathbf{X}_k$, hence we focus on the first term.

Let $\mathbf{A} := \mathbf{D}_\mathbf{z}(\mathbf{W} + \mathbf{I}) - \mathbf{I}$; we can rewrite the first term in (2) as:

$$\begin{aligned}
\sum\nolimits_{i=1}^{k} \mu^{k-i} \|\mathbf{X}_k^i \mathbf{A}\|_2^2 &= \sum\nolimits_{i=1}^{k} \mu^{k-i} \mathbf{X}_k^i \mathbf{A}\mathbf{A}^T (\mathbf{X}_k^i)^T \\
&= \sum\nolimits_{i=1}^{k} \mu^{k-i} tr(\mathbf{X}_k^i \mathbf{A}\mathbf{A}^T (\mathbf{X}_k^i)^T) \\
&= \sum\nolimits_{i=1}^{k} \mu^{k-i} tr((\mathbf{X}_k^i)^T \mathbf{X}_k^i \mathbf{A}\mathbf{A}^T) \\
&= tr(\mathbf{C}_{\mathbf{xx}}^k \mathbf{A}\mathbf{A}^T) \qquad (3)
\end{aligned}$$

[1] For two sets A, B, we use $A \setminus B$ to denote all the elements in A but not B

where $\mathbf{C}_{\mathbf{xx}}^k = \sum_{i=1}^{k} \mu^{k-i} (\mathbf{X}_k^i)^T \mathbf{X}_k^i$, and $tr(\cdot)$ denotes the trace of a matrix. Notice that the statistics $\mathbf{C}_{\mathbf{xx}}^k$ can be updated online by $\mathbf{C}_{\mathbf{xx}}^{k+1} = \mu \mathbf{C}_{\mathbf{xx}}^k + (\mathbf{X}_{k+1}^{k+1})^T \mathbf{X}_{k+1}^{k+1}$, whose memory complexity is a constant: $O(n^2)$. After that, the first term of (2) can be calculated by (3). Therefore, we can rewrite the objective function (2) using the following matrix form:

$$
\begin{aligned}
&(\mathbf{W}_{k+1}, \mathbf{z}_{k+1}) = \\
&\arg\min_{\mathbf{W},\mathbf{z}} tr\left(\mathbf{C}_{\mathbf{xx}}^k (\mathbf{D}_{\mathbf{z}}(\mathbf{W}+\mathbf{I}) - \mathbf{I})(\mathbf{D}_{\mathbf{z}}(\mathbf{W}+\mathbf{I}) - \mathbf{I})^T\right) \\
&+ \alpha \|\mathbf{S} \odot \mathbf{W}\|_1 - \beta \mathbf{z}^T \mathbf{S} \mathbf{z} + \lambda \|\mathbf{W}\|_F^2 \\
&\quad s.t.\ \mathbf{z} = [z_1, \dots, z_n] \in \{0,1\}^n, \mathbf{c}^T \mathbf{z} \le P, \qquad (4)
\end{aligned}
$$

where $\mathbf{S}_{ij} = \|\mathbf{y}_i - \mathbf{y}_j\|_2$, and $\odot$ denotes the element-wise multiplication of two matrices.

2.3 Optimization Algorithm

The main formula (4) is not convex with respect to $\mathbf{W}$ and $\mathbf{z}$ simultaneously, and $\mathbf{z}$ has binary constraint. Hence it is unrealistic to expect a suitable algorithm to easily find the global minimum. In order to solve the optimization problem fast, we propose an efficient optimization algorithm that is based on the popular alternating direction method of multipliers (ADMM) (Boyd et al. 2011; Lin, Liu, and Su 2011).

Before applying ADMM, we first transform (4) into the following equivalent formulation by introducing three variables $\widehat{\mathbf{W}}$, $\mathbf{v}$, and ξ,

$$
\begin{aligned}
&\min_{\mathbf{W},\mathbf{z}} tr\left(\mathbf{C}_{\mathbf{xx}}^k (\mathbf{D}_{\mathbf{z}}(\mathbf{W}+\mathbf{I}) - \mathbf{I})(\mathbf{D}_{\mathbf{z}}(\mathbf{W}+\mathbf{I}) - \mathbf{I})^T\right) \\
&\qquad + \alpha \|\mathbf{S} \odot \widehat{\mathbf{W}}\|_1 - \beta \mathbf{z}^T \mathbf{S} \mathbf{z} + \lambda \|\mathbf{W}\|_F^2 \\
&s.t.\ \mathbf{W} = \widehat{\mathbf{W}}, \mathbf{z} = \mathbf{v}, \mathbf{c}^T \mathbf{z} + \xi = P, \qquad (5) \\
&\qquad \mathbf{v} \in \{0,1\}^n, \xi \ge 0.
\end{aligned}
$$

It is clear that (5) is to minimize a smooth non-convex function plus separable convex regularization functions subject to linear coupling constraints; thus we can apply the ADMM type algorithm. We first present the related augmented Lagrangian function of (5),

The augmented Lagrange function of (5) is

$$
\begin{aligned}
&\mathcal{L}(\mathbf{W}, \widehat{\mathbf{W}}, \mathbf{z}, \mathbf{v}, \xi, \Lambda_1, \Lambda_2, \Lambda_3) := \\
&tr\big(\mathbf{C}_{\mathbf{xx}}^t (\mathbf{D}_{\mathbf{z}}(\mathbf{W}+\mathbf{I}) - \mathbf{I})\ (\mathbf{D}_{\mathbf{z}}(\mathbf{W}+\mathbf{I}) - \mathbf{I})^T\big) \\
&+ \alpha \|\mathbf{S} \odot \widehat{\mathbf{W}}\|_1 - \beta \mathbf{z}^T \mathbf{S} \mathbf{z} + \lambda \|\mathbf{W}\|_F^2 + \frac{\rho_1}{2} \|\mathbf{W} - \widehat{\mathbf{W}} + \frac{\Lambda_1}{\rho_1}\|_F^2 \\
&+ \frac{\rho_2}{2} \|\mathbf{z} - \mathbf{v} + \frac{\Lambda_2}{\rho_2}\|_2^2 + \frac{\rho_3}{2} (\mathbf{c}^T \mathbf{z} + \xi - P + \frac{\Lambda_3}{\rho_3})^2,
\end{aligned}
$$

where parameters Λ_1, Λ_2, and Λ_3 are the Lagrange multipliers and ρ_1, ρ_2, and ρ_3 are the constraints violation penalty parameters with respect to the linear constraints respectively. Based on the basic Gauss-Seidel structure in a ADMM-type algorithm, we will introduce how to solve these subproblems separately in detail.

i) Compute the subproblem of $\mathbf{W}^{t+1}$:

$$
\begin{aligned}
\mathbf{W}^{k+1} =& \arg\min_{\mathbf{W}}\ tr\big(\mathbf{C}_{\mathbf{xx}}^k (\mathbf{D}_{\mathbf{z}^k}(\mathbf{W}+\mathbf{I}) - \mathbf{I})(\mathbf{D}_{\mathbf{z}^k}(\mathbf{W}+\mathbf{I}) \\
&- \mathbf{I})^k\big) + \lambda \|\mathbf{W}\|_F^2 + \frac{\rho_1}{2} \|\mathbf{W} - \widehat{\mathbf{W}}^k + \frac{\Lambda_1^k}{\rho_1}\|_F^2.
\end{aligned}
$$

Taking the partial derivative of the equation above with respect to $\mathbf{W}$, and setting it to zero, we obtain:

$$
\begin{aligned}
&(\mathbf{D}_{\mathbf{z}^k} \mathbf{C}_{\mathbf{xx}}^k \mathbf{D}_{\mathbf{z}^k} + (\lambda + \frac{\rho_1}{2}) \mathbf{I}) \mathbf{W} \\
&\qquad = \mathbf{D}_{\mathbf{z}^k} \mathbf{C}_{\mathbf{xx}}^k (\mathbf{I} - \mathbf{D}_{\mathbf{z}^k}) + \frac{\rho_1}{2} (\widehat{\mathbf{W}} - \frac{\Lambda_1^k}{\rho_1}).
\end{aligned}
$$

Therefore, the optimal $\mathbf{W}^{k+1}$ can be obtained by

$$
\begin{aligned}
\mathbf{W}^{k+1} =& (\mathbf{D}_{\mathbf{z}^k} \mathbf{C}_{\mathbf{xx}}^k \mathbf{D}_{\mathbf{z}^k} + (\lambda + \frac{\rho_1}{2}) \mathbf{I})^{-1} \\
&\times (\mathbf{D}_{\mathbf{z}^k} \mathbf{C}_{\mathbf{xx}}^k (\mathbf{I} - \mathbf{D}_{\mathbf{z}^k}) + \frac{\rho_1}{2} (\widehat{\mathbf{W}}^k - \frac{\Lambda_1^k}{\rho_1})). \qquad (6)
\end{aligned}
$$

ii) Further we calculate the subproblem about $\widehat{\mathbf{W}}^{k+1}$, i.e.,

$$
\widehat{\mathbf{W}}^{k+1} = \arg\min_{\widehat{\mathbf{W}}} \alpha \|\mathbf{S} \odot \widehat{\mathbf{W}}\|_1 + \frac{\rho_1}{2} \|\widehat{\mathbf{W}} - \mathbf{W}^{k+1} - \frac{\Lambda_1^k}{\rho_1}\|_F^2.
$$

We can obtain a closed-form solution of $\widehat{\mathbf{W}}^{k+1}$ by the matrix shrinkage operation Lemma (Lin et al. 2009):

$$
\begin{aligned}
\widehat{\mathbf{W}}_{ij}^{k+1} =& \max\left\{ |(\mathbf{W}^{k+1} + \frac{\Lambda_1^k}{\rho})_{ij}| - \frac{\alpha \mathbf{S}_{ij}}{\rho_1}, 0 \right\} \\
&\cdot sgn\left((\mathbf{W}^{k+1} + \frac{\Lambda_1^k}{\rho_1})_{ij} \right), \qquad (7)
\end{aligned}
$$

where $sgn(t)$ is the signum function of t.
iii) $\mathbf{z}^{k+1}$ is the minimizer for

$$
\begin{aligned}
&\min_{\mathbf{z}} tr\big(\mathbf{C}_{\mathbf{xx}}^k (\mathbf{D}_{\mathbf{z}}(\mathbf{W}^{k+1}+\mathbf{I}) - \mathbf{I})(\mathbf{D}_{\mathbf{z}}(\mathbf{W}^{k+1}+\mathbf{I}) - \mathbf{I})^T\big) \\
&\quad - \beta \mathbf{z}^T \mathbf{S} \mathbf{z} + \frac{\rho_2}{2} \|\mathbf{z} - \mathbf{v}^k + \frac{\Lambda_2^k}{\rho_2}\|_2^2 \\
&\quad + \frac{\rho_3}{2} \left(\mathbf{c}^T \mathbf{z} + \xi^k - P + \frac{\Lambda_3^k}{\rho_3} \right)^2.
\end{aligned}
$$

This is an unconstrained quadratic programming problem. By some straight-forward computation, we can rewrite it as:

$$
\min_{\mathbf{z}}\ \mathbf{z}^T \mathbf{H} \mathbf{z} + \mathbf{b}^T \mathbf{z}, \qquad (8)
$$

where

$$
\begin{aligned}
\mathbf{H} =& \mathbf{C}_{\mathbf{xx}}^k \odot \left((\mathbf{W}^{k+1} + \mathbf{I})(\mathbf{W}^{k+1} + \mathbf{I})^T \right) \\
&- \beta \mathbf{S} + \frac{\rho_2}{2} \mathbf{I} + \frac{\rho_3}{2} \mathbf{c} \mathbf{c}^T, \\
\mathbf{b} =& -2 diag((\mathbf{W}^{k+1} + \mathbf{I}) \mathbf{C}_{\mathbf{xx}}^k) \\
&+ \rho_2 (-\mathbf{v}^k + \frac{\Lambda_2^k}{\rho_2}) + \rho_3 (\xi - P + \frac{\Lambda_3^k}{\rho_3}) \mathbf{c},
\end{aligned}
$$

where $diag(\mathbf{A})$ is a vector whose entries are the diagonal elements of $\mathbf{A}$. Thus, the optimal $\mathbf{z}^{k+1}$ can be obtained by:

$$
\mathbf{z}^{k+1} = \frac{1}{2} \mathbf{H}^{-1} \mathbf{b}. \qquad (9)
$$

iv) Computing the subproblem about $\mathbf{v}^{k+1}$ is to solve

$$\min_{\mathbf{v}} \|\mathbf{z}^{k+1} - \mathbf{v} + \frac{\Lambda_2^k}{\rho_2}\|^2, \quad s.t. \quad \mathbf{v} \in \{0,1\}^n. \tag{10}$$

This is a simple integer programming problem. We can easily obtain the optimal $\mathbf{v}^{k+1}$ by:

$$\mathbf{v}_i^{k+1} = \begin{cases} 1 & \text{if } (\mathbf{z}^{k+1} + \frac{\Lambda_2^k}{\rho_2})_i \geq 0.5; \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

v) As for the subproblem about ξ^{k+1}, we also can easily obtain a closed-form solution:

$$\xi^{k+1} = \max(P - \mathbf{c}^T \mathbf{z}^{k+1} - \frac{\Lambda_3^k}{\rho_3}, 0). \tag{12}$$

The key steps of the proposed SRSSS algorithm are summarized in Algorithm 1. When initializing Algorithm 1, we first use the typical mini-batch version ADMM to optimize (5). It means we run the SRSSS Algorithm with respect to a specific $\mathbf{C}_{\mathbf{xx}}^0$ that is constructed by the mini-batch data.

Algorithm 1 The SRSSS Algorithm

```
Input: Data arrive sequentially. Set $\alpha$, $\beta$, and $\gamma$;
Initialize:
$(\mathbf{W}^0, \widehat{\mathbf{W}}^0, \mathbf{z}^0, \mathbf{v}^0, \Lambda_1^0, \Lambda_2^0, \Lambda_3^0, \rho_1, \rho_2, \rho_3, \tau, \max_\rho)$
are initialized by adopting ADMM on a mini-batch data;
for $t = 0, 1, 2, \dots,$ do
  if $t = kL$, $k = 0, 1, 2, \cdots$, enter model update :
    1. update $\mathbf{W}^{k+1}$ by (6);
    2. update $\widehat{\mathbf{W}}^{k+1}$, $\mathbf{z}^{k+1}$ by (7) and (9);
    3. update $\mathbf{v}^{k+1}$ and $\xi^{k+1}$ by (11) and (12);
    4. let $\phi_k$, the set of active sensors in the next phase,
         be the indices of value 1 in $\mathbf{v}^{k+1}$;
    5. update the multipliers
       $\Lambda_1^{k+1} = \Lambda_1^k + \rho_1(\mathbf{W}^{k+1} - \widehat{\mathbf{W}}^{k+1})$;
       $\Lambda_2^{k+1} = \Lambda_2^k + \rho_2(\mathbf{z}^{k+1} - \mathbf{v}^{k+1})$;
       $\Lambda_3^{k+1} = \Lambda_3^k + \rho_3(\mathbf{c}^T\mathbf{z}^{k+1} + \xi - P)$;
    6. update the parameter $\rho_1$, $\rho_2$ and $\rho_3$ by
       $\rho_i = \min(\tau\rho_i, \max_\rho), i = 1, 2, 3$;
  end if
  if $t > kL$ and $t < (k+1)L$, remain in prediction phase:
    7. predict output from sensors $[n] - \phi_k$ from the
         output by sensors in $\phi_k$ and $\mathbf{W}^{k+1}$;
  end if
end for
Output: Vector $\mathbf{v}^{k+1}$.
```

3. Experiments

In this section, we empirically evaluate the proposed sensor stream selection algorithm, SRSSS, on two data sets, one challenging synthetic dataset and one real-world dataset.

3.1 Experimental Setting

Compared methods To evaluate the performance of SRSSS, we compare it with PES MULTIVARIATE (RL) (Aggarwal, Xie, and Yu 2011) that is the most related work to ours. We call it PMRL here for convenience. We also compare our method with some feature selection and active learning algorithms, since this sensor selection problem can be alternatively viewed as a feature selection problem or an active learning problem by treating each sensor as a feature or an instance. We choose two unsupervised feature selection methods and two representation based active learning methods as alternatives in the experiments: Laplacian score (LS) (He, Cai, and Niyogi 2005), JELSR (Hou et al. 2011), TED (Yu, Bi, and Tresp 2006), and RRSS (Nie et al. 2013). After obtaining the active sensors by the methods above, we use multi-variate interpolation to infer the output from the inactive sensors for fair comparisons.

Parameter setting There are some parameters to be set in advance. For PMRL, LS, JELSR, TED, and RRSS, we set the parameters *window size* to 10 to load the data for each update, same as in (Aggarwal, Xie, and Yu 2011). The parameter *maxlap* is set to 5 for PMRL. For SRSSS, the forgetting factor μ is set to 0.9 throughout the experiments. For all the methods, the power constraint P is set to 10, and the prediction interval L is set to 5 throughout the experiments, unless otherwise stated. The regularization parameters for all the methods are tuned by cross-validation on a mini batch dataset. In the experiments, we use the first 100 observation values to tune parameters. In addition, we need to generate a synthetic power requirement for each sensor. In the experiments, the power requirements of different sensors are uniformly distributed in the range $[0, 1]$.

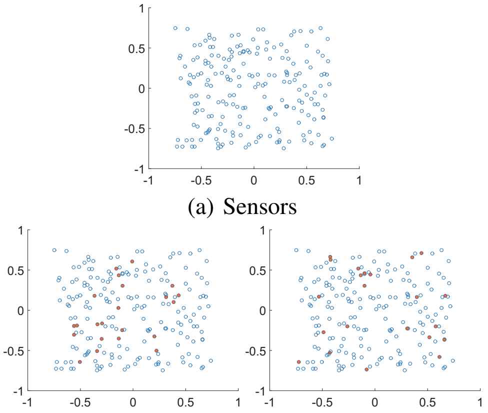
Experiments protocol We conduct the experiments on a laptop with 2.8 GHz Intel i-5 CPU and 12GB memory by a single thread, and implement the algorithms using MATLAB R2014b 64bit edition. In the experimental study, we use a popular regression prediction metric, Mean Absolute Error (MAE), to measure the prediction quality.

3.2 Results on the synthetic dataset

We first conduct the experiments on a challenging synthetic dataset. Here we use parallel discrete event simulation (PDEs) (Fujimoto 1990) to generate synthetic sensor data[2]. To be more specific, we simulate an open environment and the sensors detecting temperature. Our environment is a square region with Dirichlet boundaries temperature set at constant zero. There are 200 sensors randomly placed to form the sensor network. The overall simulation time period is 1000 time units. Heat comes from 20 diffusion sources, whose locations randomly change 50 times, i.e., once per 20 time units. The sensors' data sampling rate is one per time unit.

Table 1 lists the MAE of the six aforementioned algorithms under different prediction interval L, the amount of time each prediction phase lasts. By building better connection with multi-variate interpolation, and leveraging spatial information, SRSSS achieves the best performance among all the methods. Figure 2 (a) displays the locations of all the sensors. The active sensors selected by PMRL and SRSSS at

[2] The toolbox is downloaded from http://www.ulb.ac.be/di/labo/projects.html.



(a) Sensors (b) PMRL (c) SRSSS

Figure 2: Sensor selection by PMRL (b) and SRSS (c) at $t = 100$. The selected sensors are marked as solid red dots. The sensors selected by SRSSS represent the original data set (a) better.

Table 1: MAEs of different methods on the synthetic dataset.

L	LS	JELSR	TED	RRSS	PMRL	SRSSS
5	0.754	0.378	0.400	0.305	0.506	**0.263**
7	0.825	0.512	0.372	0.320	0.547	**0.311**
10	1.733	0.825	0.969	0.730	1.182	**0.599**

$t = 100$ are marked as solid red dots in Figure 2 (b) and (c), respectively. As can be seen, the sensors selected by SRSSS are more spread-out in space. As is well-known, temperature is a typical physical measurement that closer observations in space are more correlated than distant ones. By incorporating the spatial regularizations, as can be verified by the results in Table 1, SRSSS selects a better sensor subset for predicting those inactive sensors.

3.3 Results on the real dataset

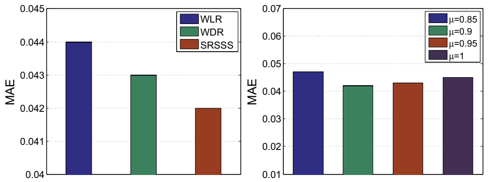
We further perform the proposed algorithm on a real-world dataset derived from the Intel research laboratory at Berkeley. This dataset is popular for testing sensor selection algorithms (Aggarwal, Xie, and Yu 2011)[3]. It has 54 sensors and 5 days of temperature readings. The sensors 5 and 15 were removed as they do not provide any measurement. The readings are sampled every 30 seconds, and so the dataset contains in total 14400 readings from the rest 52 sensors.

The results are reported in Table 2. SRSSS performs better than the other methods under different L's. For example, SRSSS achieves 23.6%, 27.8%, 21.5% relative error deduction over PMRL, respectively.

We also verify the effectiveness of the different components of the objective function (1) in SRSSS on this dataset. We set α and β to zeros respectively, and fix the other variables. We call them WLR (without local regularization) and WDR (without distance regularization), respectively. The results are shown in Figure 3 (a). SRSSS is superior to WLR and WDR, which demonstrates that both the two spatial regularizations (local information and distance) are beneficial for selecting informative sensors. In addition, we also test the effect of the time-forgetting factor μ on the prediction, whose result is shown in Figure 3 (b). When $\mu = 1$, all the samples have the same weight for model update. As can be seen, when $\mu = 0.9$, SRSSS obtains the best performance. This indicates that the forgetting factor enables SRSSS to adapt to sensor streams' evolvement.

Table 2: MAEs of different methods on the real dataset.

L	LS	JELSR	TED	RRSS	PMRL	SRSSS
5	0.061	0.048	0.061	0.052	0.055	**0.042**
7	0.066	0.053	0.074	0.061	0.072	**0.052**
10	0.085	0.067	0.088	0.077	0.079	**0.062**



(a) Regularization terms (b) The forgetting factor μ

Figure 3: Effectiveness verification of the components of SRSSS on the real dataset.

3.4 Efficiency

We test the efficiency of our algorithm on both the synthetic and the real datasets. We compare SRSSS with the sensor selection algorithm PMRL, the feature selection approach JELSR, and the active learning method RRSS. Figure 4 shows the runtime of these approaches. SRSSS has the least runtime time on both datasets. Compared to PMRL, SRSSS is over 30 and 10 times faster on the synthetic and the real datasets, respectively.

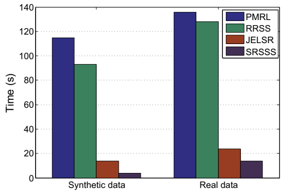


Figure 4: The runtime of different methods on both the synthetic and the real datasets.

4. Conclusion

In this paper, we propose a novel streaming sensor selection for redundant sensor network. By performing sensor selection in a multi-variate interpolation framework and incorporating sensors' spatial information, the information loss of the inactive sensors can be minimized. By defining a statistical variable to compress the data without information

[3] The dataset is downloaded from http://www.ulb.ac.be/di/labo/datasets.html.

loss and introducing a time forgetting factor to set different weights on samples, the memory usage can be optimized. Experiments on both synthetic and real datasets demonstrate the effectiveness and efficiency of our method.

Acknowledgments

The work was supported in part by the National Natural Science Foundation of China (Grant No. 61532009, 61402023, 11501210), and the Shanghai YangFan Funding (Grant No. 15YF1403400).

References

Abrams, Z.; Goel, A.; and Plotkin, S. 2004. Set k-cover algorithms for energy efficient monitoring in wireless sensor networks. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, 424–432. ACM.

Aggarwal, C. C.; Bar-Noy, A.; and Shamoun, S. 2011. On sensor selection in linked information networks. In *International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS)*, 1–8. IEEE.

Aggarwal, C. C.; Xie, Y.; and Yu, P. S. 2011. On dynamic data-driven selection of sensor streams. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 1226–1234. ACM.

Boyd, S.; Parikh, N.; Chu, E.; Peleato, B.; and Eckstein, J. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning* 3(1):1–122.

Chan, K. Y.; Khadem, S.; Dillon, T. S.; Palade, V.; Singh, J.; and Chang, E. 2012. Selection of significant on-road sensor data for short-term traffic flow forecasting using the taguchi method. *IEEE Transactions on Industrial Informatics* 8(2):255–266.

Fujimoto, R. M. 1990. Parallel discrete event simulation. *Communications of the ACM* 33(10):30–53.

Golovin, D.; Faulkner, M.; and Krause, A. 2010. Online distributed sensor selection. In *Proceedings of the 9th International Conference on Information Processing in Sensor Networks*, 220–231. ACM.

He, X.; Cai, D.; and Niyogi, P. 2005. Laplacian score for feature selection. In *Advances in Neural Information Processing Systems (NIPS)*, 507–514.

Hou, C.; Nie, F.; Yi, D.; and Wu, Y. 2011. Feature selection via joint embedding learning and sparse regression. In *International Joint Conference on Artificial Intelligence (IJCAI)*, volume 22, 1324–1329. Citeseer.

Hovland, G. E., and McCarragher, B. J. 1997. Dynamic sensor selection for robotic systems. In *IEEE International Conference on Robotics and Automation*, volume 1, 272–277. IEEE.

Kollios, G.; Byers, J. W.; Considine, J.; Hadjieleftheriou, M.; and Li, F. 2005. Robust aggregation in sensor networks. *IEEE Data Engineering Bulletin* 28(1):26–32.

Li, C.; Wei, F.; Dong, W.; Liu, Q.; Wang, X.; and Zhang, X. 2015. Dynamic structure embedded online multiple-output regression for stream data. *CoRR* abs/1412.5732.

Lin, Z.; Chen, M.; Wu, L.; and Ma, Y. 2009. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *Technical report, UIUC Technical Report UILU-ENG-09-2215*.

Lin, Z.; Liu, R.; and Su, Z. 2011. Linearized alternating direction method with adaptive penalty for low-rank representation. In *Advances in Neural Information Processing Systems (NIPS)*, 612–620.

Nie, F.; Wang, H.; Huang, H.; and Ding, C. 2013. Early active learning via robust representation and structured sparsity. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI)*, 1572–1578. AAAI Press.

Rowaihy, H.; Eswaran, S.; Johnson, M.; Verma, D.; Bar-Noy, A.; Brown, T.; and La Porta, T. 2007. A survey of sensor selection schemes in wireless sensor networks. In *Defense and Security Symposium*, 65621A–65621A.

Wang, H.; Yao, K.; Pottie, G.; and Estrin, D. 2004. Entropy-based sensor selection heuristic for target localization. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, 36–45. ACM.

Yu, K.; Bi, J.; and Tresp, V. 2006. Active learning via transductive experimental design. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, 1081–1088. ACM.

Zhang, Y., and Ji, Q. 2005. Sensor selection for active information fusion. In *Proceedings of the 20th national conference on Artificial intelligence (AAAI)*, 1229–1234.

Appendix

Theorem 2.1: *The objective function (1) is equivalent to*

$$
\begin{aligned}
&(\mathbf{W}_{k+1}, \mathbf{z}_{k+1}) \\
&= \arg\min_{\mathbf{W},\mathbf{z}} \sum_{i=1}^{k} \mu^{k-i} \|\mathbf{X}_k^i \mathbf{D}_\mathbf{z} \mathbf{W} - \mathbf{X}_k^i (\mathbf{I} - \mathbf{D}_\mathbf{z})\|_2^2 \\
&+ \alpha \sum_{i,j=1}^{n} \|\mathbf{y}_i - \mathbf{y}_j\|_2 |W_{ij}| - \beta \sum_{i,j=1}^{n} \|\mathbf{y}_i - \mathbf{y}_j\|_2 z_i z_j + \lambda \|\mathbf{W}\|_F^2 \\
&s.t.\ \mathbf{z} = [z_1, \dots, z_n] \in \{0,1\}^n, \mathbf{c}^T \mathbf{z} \le P. \qquad (13)
\end{aligned}
$$

Proof. Let

$$
\begin{aligned}
J(\mathbf{W}, \mathbf{z}) = &\sum_{i=1}^{k} \mu^{k-i} \|\mathbf{X}_k^i \mathbf{D}_\mathbf{z} \mathbf{W} (\mathbf{I} - \mathbf{D}_\mathbf{z}) - \mathbf{X}_k^i (\mathbf{I} - \mathbf{D}_\mathbf{z})\|_2^2 \\
&+ \alpha \sum_{i,j=1}^{n} \|\mathbf{y}_i - \mathbf{y}_j\|_2 |W_{ij}| - \beta \sum_{i,j=1}^{n} \|\mathbf{y}_i - \mathbf{y}_j\|_2 z_i z_j + \lambda \|\mathbf{W}\|_F^2.
\end{aligned}
$$

$$
\begin{aligned}
\tilde{J}(\mathbf{W}, \mathbf{z}) = &\sum_{i=1}^{k} \mu^{k-i} \|\mathbf{X}_k^i \mathbf{D}_\mathbf{z} \mathbf{W} - \mathbf{X}_k^i (\mathbf{I} - \mathbf{D}_\mathbf{z})\|_2^2 \\
&+ \alpha \sum_{i,j=1}^{n} \|\mathbf{y}_i - \mathbf{y}_j\|_2 |W_{ij}| - \beta \sum_{i,j=1}^{n} \|\mathbf{y}_i - \mathbf{y}_j\|_2 z_i z_j + \lambda \|\mathbf{W}\|_F^2.
\end{aligned}
$$

We will prove that $\min J = \min \tilde{J}$ and $\arg\min J = \arg\min \tilde{J}$.

Given k, we first notice that in order to minimize J, W an should be bounded. More specifically, the $\|W\|_\infty$ should be bounded. We will prove this fact now. Notice that

$$
\begin{aligned}
&\min_{\mathbf{W},\mathbf{z}} J(\mathbf{W}, \mathbf{z}) \le \min_{\mathbf{z}} J(\mathbf{0}, \mathbf{z}) \\
&\le \min_{\mathbf{z}} \sum_{i=1}^{k} \mu^{k-i} \|\mathbf{X}_k^i (\mathbf{I} - \mathbf{D}_\mathbf{z})\|_2^2 - \beta \sum_{i,j=1}^{n} \|\mathbf{y}_i - \mathbf{y}_j\|_2 z_i z_j \\
&\le \sum_{i=1}^{k} \mu^{k-i} \|\mathbf{X}_k^i\|_2^2.
\end{aligned}
$$

On the other hand,

$$
J(\mathbf{W}, \mathbf{z}) \ge -\beta \sum_{i,j=1}^{n} \|\mathbf{y}_i - \mathbf{y}_j\|_2 + \lambda \|W\|_\infty^2.
$$

Therefore in order to minimize J, we must have

$$
-\beta \sum_{i,j=1}^{n} \|\mathbf{y}_i - \mathbf{y}_j\|_2 + \lambda \|W\|_\infty^2 \le \sum_{i=1}^{k} \mu^{k-i} \|\mathbf{X}_k^i\|_2^2,
$$

and thus we obtain an upper bound for $\|W\|_\infty$.

Thus to minimize $J(\mathbf{W}, \mathbf{z})$, we can simply restrict $\mathbf{W}, \mathbf{z}$ to a compact space. Therefore the global minimizer exist, and suppose the minimizer is given by $\mathbf{W}^*, \mathbf{z}^*$.

Similarly, we can show that for $\tilde{J}(\mathbf{W}, \mathbf{z})$, a global minimizer exist. Suppose the global minimizer is given by $\tilde{\mathbf{W}}^*, \tilde{\mathbf{z}}^*$.

We now show that $J(\mathbf{W}, \mathbf{z}^*) \ge \tilde{J}(\mathbf{W}^*, \mathbf{z}^*)$. Let $\tilde{\mathbf{W}} = \mathbf{W}^*(\mathbf{I} - \mathbf{D}_{\mathbf{z}^*})$. Notice that in this case,

$$
\tilde{\mathbf{W}}(\mathbf{I} - \mathbf{D}_{\mathbf{z}^*}) = \mathbf{W}^*(\mathbf{I} - \mathbf{D}_{\mathbf{z}^*})(\mathbf{I} - \mathbf{D}_{\mathbf{z}^*}) = \mathbf{W}^*(\mathbf{I} - \mathbf{D}_{\mathbf{z}^*}),
$$

as $\mathbf{z}^*$ is a diagonal matrix with diagonal entries being 0 or 1. And thus we also have that

$$
\mathbf{D}_{\mathbf{z}^*} \tilde{\mathbf{W}}(\mathbf{I} - \mathbf{D}_{\mathbf{z}^*}) = \mathbf{D}_{\mathbf{z}^*} \mathbf{W}^*(\mathbf{I} - \mathbf{D}_{\mathbf{z}^*}).
$$

Therefore we have

$$
\begin{aligned}
&\min \tilde{J}(\mathbf{W}, \mathbf{z}) \le \tilde{J}(\tilde{\mathbf{W}}, \mathbf{z}^*) \\
&= \sum_{i=1}^{k} \mu^{k-i} \|\mathbf{X}_k^i \mathbf{D}_{\mathbf{z}^*} \tilde{\mathbf{W}} - \mathbf{X}_k^i (\mathbf{I} - \mathbf{D}_{\mathbf{z}^*})\|_2^2 \\
&\quad + \alpha \sum_{i,j=1}^{n} \|\mathbf{y}_i - \mathbf{y}_j\|_2 |\tilde{W}_{ij}| - \beta \sum_{i,j=1}^{n} \|\mathbf{y}_i - \mathbf{y}_j\|_2 z_i z_j + \lambda \|\tilde{\mathbf{W}}\|_F^2 \\
&= \sum_{i=1}^{k} \mu^{k-i} \|\mathbf{X}_k^i \mathbf{D}_{\mathbf{z}^*} \mathbf{W}^*(\mathbf{I} - \mathbf{D}_{\mathbf{z}^*}) - \mathbf{X}_k^i (\mathbf{I} - \mathbf{D}_{\mathbf{z}^*})\|_2^2 \\
&\quad + \alpha \sum_{i,j=1}^{n} \|\mathbf{y}_i - \mathbf{y}_j\|_2 |\tilde{W}_{ij}| - \beta \sum_{i,j=1}^{n} \|\mathbf{y}_i - \mathbf{y}_j\|_2 z_i^* z_j^* + \lambda \|\tilde{\mathbf{W}}\|_F^2 \\
&= \sum_{i=1}^{k} \mu^{k-i} \|\mathbf{X}_k^i \mathbf{D}_{\mathbf{z}^*} \mathbf{W}^*(\mathbf{I} - \mathbf{D}_{\mathbf{z}^*})^2 - \mathbf{X}_k^i (\mathbf{I} - \mathbf{D}_{\mathbf{z}^*})\|_2^2 \\
&\quad + \alpha \sum_{i,j=1}^{n} \|\mathbf{y}_i - \mathbf{y}_j\|_2 |\tilde{W}_{ij}| - \beta \sum_{i,j=1}^{n} \|\mathbf{y}_i - \mathbf{y}_j\|_2 z_i^* z_j^* + \lambda \|\tilde{\mathbf{W}}\|_F^2 \\
&= \sum_{i=1}^{k} \mu^{k-i} \|\mathbf{X}_k^i \mathbf{D}_{\mathbf{z}^*} \tilde{\mathbf{W}}(\mathbf{I} - \mathbf{D}_{\mathbf{z}^*}) - \mathbf{X}_k^i (\mathbf{I} - \mathbf{D}_{\mathbf{z}^*})\|_2^2 \\
&\quad + \alpha \sum_{i,j=1}^{n} \|\mathbf{y}_i - \mathbf{y}_j\|_2 |\tilde{W}_{ij}| - \beta \sum_{i,j=1}^{n} \|\mathbf{y}_i - \mathbf{y}_j\|_2 z_i^* z_j^* + \lambda \|\tilde{\mathbf{W}}\|_F^2 \\
&= \sum_{i=1}^{k} \mu^{k-i} \|\mathbf{X}_k^i \mathbf{D}_{\mathbf{z}^*} \mathbf{W}^*(\mathbf{I} - \mathbf{D}_{\mathbf{z}^*}) - \mathbf{X}_k^i (\mathbf{I} - \mathbf{D}_{\mathbf{z}^*})\|_2^2 \\
&\quad + \alpha \sum_{i,j=1}^{n} \|\mathbf{y}_i - \mathbf{y}_j\|_2 |\tilde{W}_{ij}| - \beta \sum_{i,j=1}^{n} \|\mathbf{y}_i - \mathbf{y}_j\|_2 z_i^* z_j^* + \lambda \|\tilde{\mathbf{W}}\|_F^2 \\
&\le \sum_{i=1}^{k} \mu^{k-i} \|\mathbf{X}_k^i \mathbf{D}_{\mathbf{z}^*} \mathbf{W}^*(\mathbf{I} - \mathbf{D}_{\mathbf{z}^*}) - \mathbf{X}_k^i (\mathbf{I} - \mathbf{D}_{\mathbf{z}^*})\|_2^2 \\
&+ \alpha \sum_{i,j=1}^{n} \|\mathbf{y}_i - \mathbf{y}_j\|_2 |W^*| - \beta \sum_{i,j=1}^{n} \|\mathbf{y}_i - \mathbf{y}_j\|_2 z_i^* z_j^* + \lambda \|\mathbf{W}^*\|_F^2 \\
&= J(\mathbf{W}^*, \mathbf{z}^*) = \min J,
\end{aligned}
$$

where the equality holds only when $\|\mathbf{W}^*\|_F^2 = \|\tilde{\mathbf{W}}\|_F^2$, and $\mathbf{A}^2$ denotes $\mathbf{A} * \mathbf{A}$. Thus we must have $\mathbf{W}^* = \tilde{\mathbf{W}}$, i.e., the columns of $\mathbf{W}$ not selected by $\mathbf{z}^*$ has to be 0.

Therefore so far we have proved that $\mathbf{W}^* = \tilde{\mathbf{W}}$, and that $\min_{\mathbf{W},\mathbf{z}} J(\mathbf{W}, \mathbf{z}) \ge \min_{\mathbf{W},\mathbf{z}} \tilde{J}(\mathbf{W}, \mathbf{z})$.

We are left to prove that $\tilde{J}(\tilde{\mathbf{W}}^*, \tilde{\mathbf{z}}^*) \ge J(\mathbf{W}^*, \mathbf{z}^*)$. We want to prove the claim

$$
\tilde{J}(\tilde{\mathbf{W}}^*, \tilde{\mathbf{z}}^*) \ge \tilde{J}(\tilde{\mathbf{W}}^*(\mathbf{I} - \mathbf{D}_{\tilde{\mathbf{z}}^*}), \tilde{\mathbf{z}}^*), \qquad (14)
$$

with the equality holds if and only if $\tilde{\mathbf{W}}^* = \mathbf{W}' := \tilde{\mathbf{W}}^*(\mathbf{I} - \mathbf{D}_{\tilde{\mathbf{z}}^*})$. If this is the case, similar as in proving the other direction, we will have

$$
\begin{aligned}
&\tilde{J}(\tilde{\mathbf{W}}^*, \tilde{\mathbf{z}}^*) = \tilde{J}(\tilde{\mathbf{W}}^*(\mathbf{I} - \mathbf{D}_{\tilde{\mathbf{z}}^*}), \tilde{\mathbf{z}}^*) \\
&= J(\tilde{\mathbf{W}}^*(\mathbf{I} - \mathbf{D}_{\tilde{\mathbf{z}}^*}), \tilde{\mathbf{z}}^*) = J(\mathbf{W}', \tilde{\mathbf{z}}^*) \ge J(\mathbf{W}^*, \mathbf{z}^*).
\end{aligned}
$$

To prove the claim (14),

$$\begin{aligned}
&\tilde{J}(\tilde{\mathbf{W}}^*, \tilde{\mathbf{z}}^*) = \\
&\textstyle\sum_{i=1}^{k} \mu^{k-i} \|\mathbf{X}_k^i \mathbf{D}_{\tilde{\mathbf{z}}^*} \tilde{\mathbf{W}}^* - \mathbf{X}_k^i (\mathbf{I} - \mathbf{D}_{\tilde{\mathbf{z}}^*})\|_2^2 \\
&+ \alpha \sum_{i,j=1}^{n} \|\mathbf{y}_i - \mathbf{y}_j\|_2 |\tilde{W}_{ij}^*| - \beta \sum_{i,j=1}^{n} \|\mathbf{y}_i - \mathbf{y}_j\|_2 z_i^* z_j^* \\
&+ \lambda \|\tilde{\mathbf{W}}^*\|_F^2 \\
\geq &\textstyle\sum_{i=1}^{k} \mu^{k-i} \|\mathbf{X}_k^i \mathbf{D}_{\tilde{\mathbf{z}}^*} \tilde{\mathbf{W}}^* - \mathbf{X}_k^i (\mathbf{I} - \mathbf{D}_{\tilde{\mathbf{z}}^*})\|_2^2 \\
&+ \alpha \sum_{i,j=1}^{n} \|\mathbf{y}_i - \mathbf{y}_j\|_2 |{W'}_{ij}^*| - \beta \sum_{i,j=1}^{n} \|\mathbf{y}_i - \mathbf{y}_j\|_2 z_i^* z_j^* \\
&+ \lambda \|\mathbf{W'}^*\|_F^2.
\end{aligned}$$

The equality holds only when $\|\mathbf{W'}^*\|_F^2 = \|\tilde{\mathbf{W}}^*\|_F^2$.

Now we are only left to show that

$$\begin{aligned}
&\textstyle\sum_{i=1}^{k} \mu^{k-i} \|\mathbf{X}_k^i \mathbf{D}_{\tilde{\mathbf{z}}^*} \tilde{\mathbf{W}}^* - \mathbf{X}_k^i (\mathbf{I} - \mathbf{D}_{\tilde{\mathbf{z}}^*})\|_2^2 \\
\geq &\textstyle\sum_{i=1}^{k} \mu^{k-i} \|\mathbf{X}_k^i \mathbf{D}_{\tilde{\mathbf{z}}^*} \mathbf{W}' - \mathbf{X}_k^i (\mathbf{I} - \mathbf{D}_{\tilde{\mathbf{z}}^*})\|_2^2. \qquad (15)
\end{aligned}$$

Let $\mathbf{A} = \mathbf{X}_k^i \mathbf{D}_{\tilde{\mathbf{z}}^*} \tilde{\mathbf{W}}^* - \mathbf{X}_k^i (\mathbf{I} - \mathbf{D}_{\tilde{\mathbf{z}}^*})$ for writing conveniently. Notice that

$$\begin{aligned}
&\textstyle\sum_{i=1}^{k} \mu^{k-i} \|\mathbf{A}\|_2^2 \\
= &\textstyle\sum_{i=1}^{k} \mu^{k-i} \|\mathbf{A}(\mathbf{D}_{\tilde{\mathbf{z}}^*} + (\mathbf{I} - \mathbf{D}_{\tilde{\mathbf{z}}^*})\|_2^2 \\
= &\textstyle\sum_{i=1}^{k} \mu^{k-i} \|\mathbf{A}\mathbf{D}_{\tilde{\mathbf{z}}^*}\|_2^2 + \sum_{i=1}^{k} \mu^{k-i} \|\mathbf{A}(\mathbf{I} - \mathbf{D}_{\tilde{\mathbf{z}}^*})\|_2^2 \\
= &\textstyle\sum_{i=1}^{k} \mu^{k-i} \|\mathbf{X}_k^i \mathbf{D}_{\tilde{\mathbf{z}}^*} \tilde{\mathbf{W}}^* \mathbf{D}_{\tilde{\mathbf{z}}^*}\|_2^2 \\
&+ \textstyle\sum_{i=1}^{k} \mu^{k-i} \|\mathbf{X}_k^i \mathbf{D}_{\tilde{\mathbf{z}}^*} \tilde{\mathbf{W}}^* (\mathbf{I} - \mathbf{D}_{\tilde{\mathbf{z}}^*}) - \mathbf{X}_k^i (\mathbf{I} - \mathbf{D}_{\tilde{\mathbf{z}}^*})\|_2^2 \\
\geq &\textstyle\sum_{i=1}^{k} \mu^{k-i} \|\mathbf{X}_k^i \mathbf{D}_{\tilde{\mathbf{z}}^*} \mathbf{W}' - \mathbf{X}_k^i (\mathbf{I} - \mathbf{D}_{\tilde{\mathbf{z}}^*})\|_2^2,
\end{aligned}$$

with equality holds only when $\|\mathbf{X}_k^i \mathbf{D}_{\tilde{\mathbf{z}}^*} \tilde{\mathbf{W}}^* \mathbf{D}_{\tilde{\mathbf{z}}^*})\|_2^2 = 0$. Combining with the previous argument, we have that

$$\tilde{J}(\tilde{\mathbf{W}}^*, \tilde{\mathbf{z}}^*) \geq \tilde{J}(\tilde{\mathbf{W}}^*(\mathbf{I} - \mathbf{D}_{\tilde{\mathbf{z}}^*}), \tilde{\mathbf{z}}^*), \qquad (16)$$

with the equality holds if and only if $\tilde{\mathbf{W}}^* = \mathbf{W}' := \tilde{\mathbf{W}}^*(\mathbf{I} - \mathbf{D}_{\tilde{\mathbf{z}}^*})$.

Collecting all what we have obtained so far, we have

$$\begin{aligned}
\min J &= J(\mathbf{W}^*, \mathbf{z}^*) \\
&\geq \tilde{J}(\mathbf{W}^*, \mathbf{z}^*) \geq \tilde{J}(\tilde{\mathbf{W}}^*, \tilde{\mathbf{z}}^*) \geq J(\mathbf{W}', \tilde{\mathbf{z}}^*).
\end{aligned}$$

Therefore we have $\min J = \min \tilde{J}$, and the minimizer can be obatined to be the same, such that the columns of $\mathbf{W}$ selected by $\mathbf{z}$ has to be 0. □