Reducing The Data Transmission in Wireless Sensor Networks Using The Principal Component Analysis

A. Rooshenas, H. R. Rabiee, A. Movaghar, M. Y. Naderi

The Sixth International Conference on Intelligent Sensors, Sensor Networks and Information Processing
ISSNIP 2010

8 Dec 2010

Problem Definition

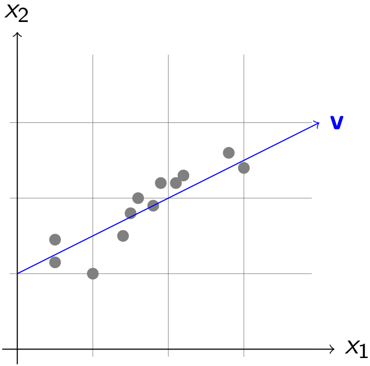
Different Power Consumption in Routing Tree

- Leaf Nodes
- Intermediate Node

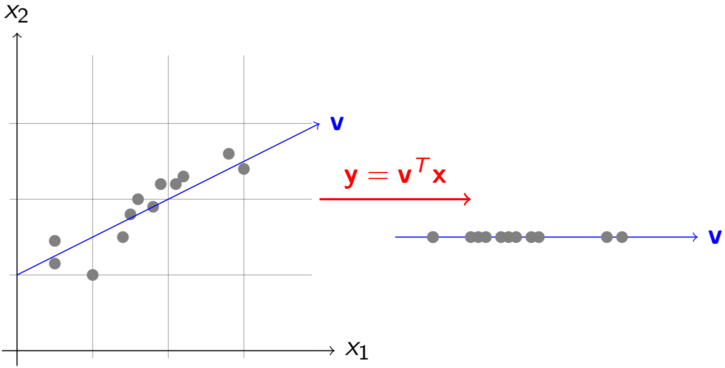
Problems

- Intermediate nodes should relay other packets to the base station.
- Lifetime of intermediate nodes affect the longevity of Wireless Sensor Networks

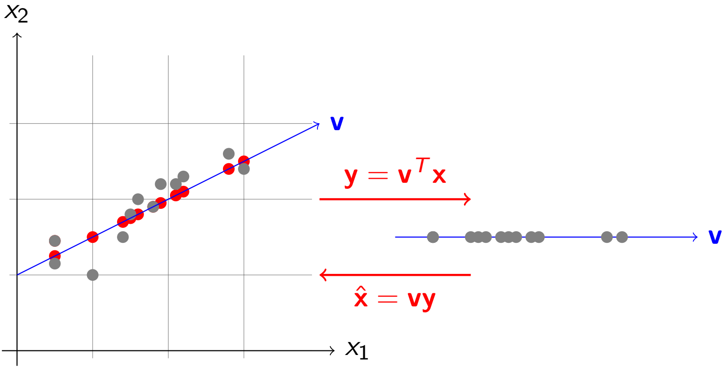
Dimension Reduction



Dimension Reduction



Dimension Reduction

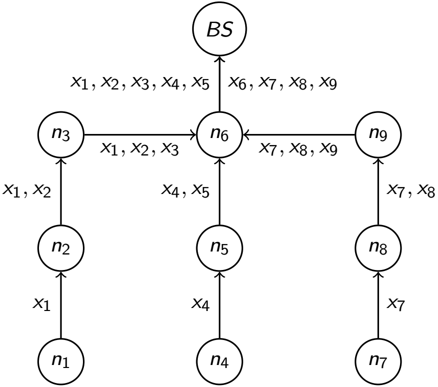


Related Works

- Principal Component Aggregation (PCAg).
  - Unsupervised
  - Supervised
- Distributed Principal Component Analysis (DPCA).

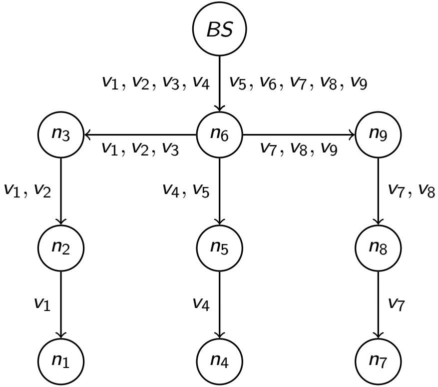
Principal Component Aggregation

- Base station gathers observations from sensor networks



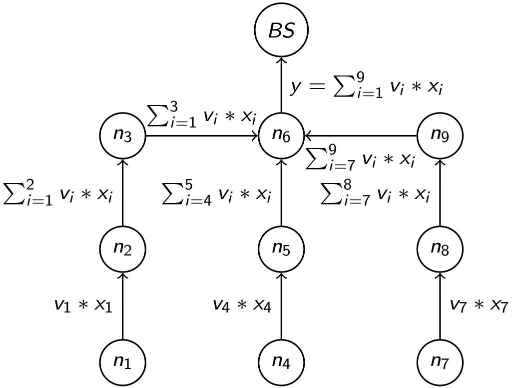
Principal Component Aggregation

- Base station calculates eigenvectors and sends them back to network



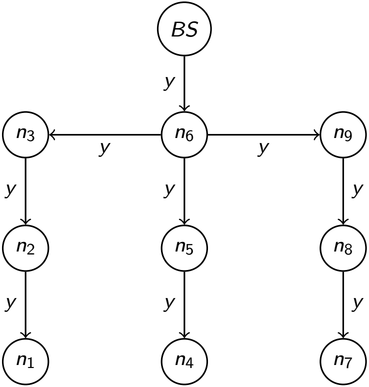
Principal Component Aggregation

- Sensor nodes send partial PCs in each epoch
- The base station reconstructs observations
- Reconstruction error is not available at the base station
- The base station calculates eigenvectors periodically



Principal Component Aggregation

- The base station sends PCs to the network
- Each sensor can calculate its reconstruction error
- $e_i = \|x_i - y * v_i\|$
- Updating one observation may change all other elements of an eigenvector
- Update decision is ambiguous



Distributed Principal Component Analysis

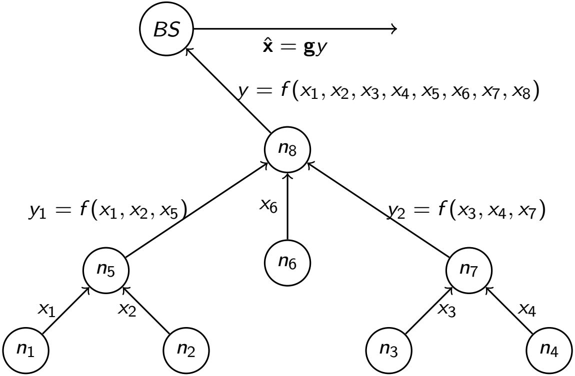
- Assumptions:
  - Each sensor receives sensed value from its neighbors
  - Only the data of sensors which are in the radio range of each other are correlated
- For the computation of eigenvectors, DPCA uses: $v_i^{t+1} = \sum_{j=1}^{N} C_{ij} * v_j^t$
- Each sensor broadcasts its eigenvector's entry v_i and its sensed value.
- Assumptions of DCPA are not supported by weather data

Using PCA for Data Reduction

Retained Variance for 50 sensors

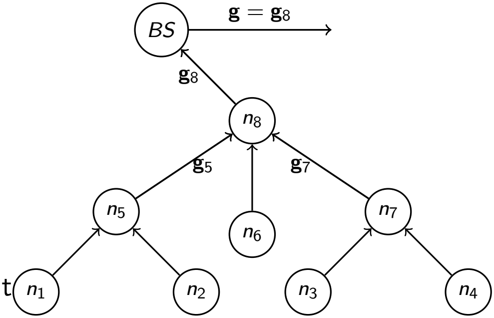
	Retained Variance	λ_1/λ_2
Temperature	1.0	3.6909e+015
Humidity	1.0	3.4407e+015
Light	1.0	2.4402e+015
Voltage	1.0	3.1737e+015
Combined	1.0	1.9305e+015

- Leaf nodes send their observations
- Intermediate nodes compute PCs
- Intermediate nodes recalculate PCs according to reconstruction error
- The next level intermediate nodes reconstruct observation and compute PCs from all downstream observations
- The base station reconstructs observations



Updating Eigenvectors

- Eigenvectors are common between each pair of parent-child intermediate nodes
- Nodes which are in the vicinity of the base station must send large eigenvectors.
- For these nodes, the computation of the covariance matrix and eigenvectors is energy consuming
- Updated eigenvectors must be sent to the base station
- The base station calculates final eigenvectors



Features of Using PCA

Features of Using PCA

- Each intermediate node has error threshold that can be configured by the base station
- High threshold value causes low accuracy and high efficiency
- An intermediate node only sends PCs in each epoch

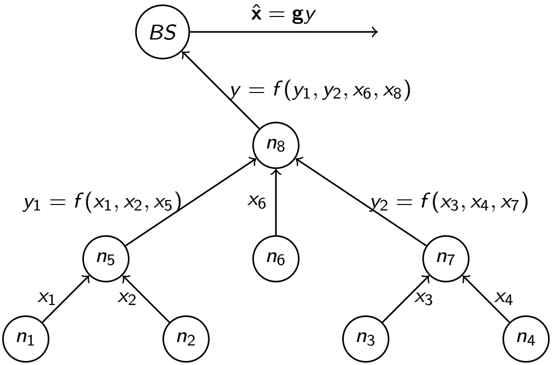
Collective PCA

- Reconstruction of data in each intermediate node and combination again is not scalable
- Collective PCA: PCA is invariant to an orthonormal linear transformation
- We want to find the principal component of $\mathbf{x} = [\mathbf{x}_1^T, \mathbf{x}_2^T, s]^T$ where s is sensed value.
- y_1 and y_2 are principal compenents of $\mathbf{x}_1$ and $\mathbf{x}_2$
- $\mathbf{z} = \begin{bmatrix} y_1 \\ y_2 \\ s \end{bmatrix}$ and $\mathbf{g}$ is the eigenvector of $\mathbf{z}\mathbf{z}^T$ then

  the principal component of $\mathbf{x}$ is $\mathbf{y} = \mathbf{g}^T\mathbf{z}$

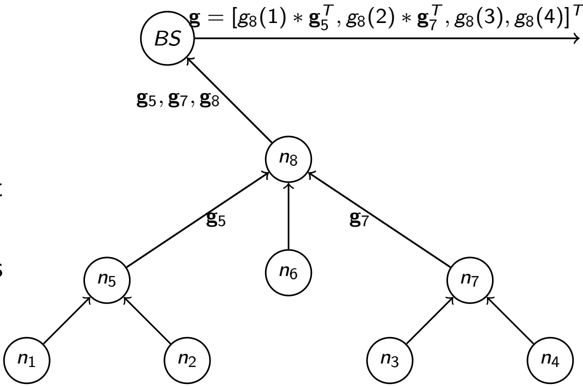
Aggregation of PCs using Collective PCA

- Leaf nodes send their observations
- Intermediate nodes compute PCs
- The next level intermediate nodes compute PCs from observations and received PCs
- The base btation reconstructs observations
- Intermediate nodes only send PCs in each epoch



Updating Eigenvectors

- Each intermediate node has its eigenvectors
- Each intermediate node updates its eigenvectors independently
- Updated eigenvectors must be sent to the base station
- The base station calculates final eigenvectors
- Update rate depends on the number of changes in environment



Proposed method: An aggregation service for calculation of reconstruction Error

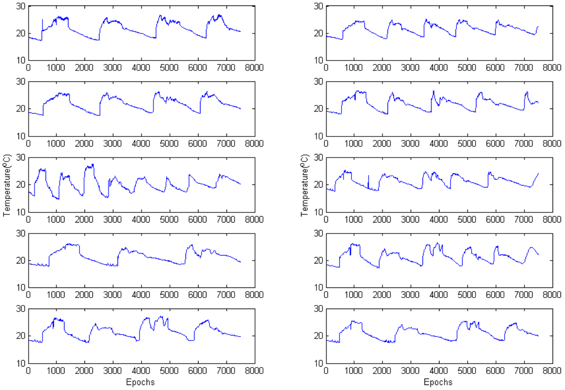
- In PCAg, the base station needs to update eigenvectors in order to compute reconstruction error
- In LocalPCA, the base station needs to tune thresholds
- In PCAg and LocalPCA, the base station do not access the real observation to calculate reconstruction error
- We proposed an aggregation service facilitating the calculation of reconstruction error at the base station.
- EAPCAg : We enhance PCAg using this aggregation service

Proposed method: An aggregation Service for Calculation of Reconstruction Error

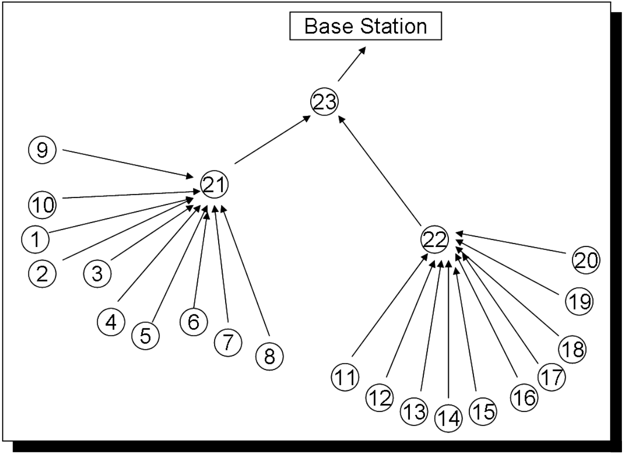
$$
\begin{aligned}
\left\|\hat{\mathbf{x}} - \mathbf{x}\right\|^2 &= (\hat{\mathbf{x}} - \mathbf{x})^T(\hat{\mathbf{x}} - \mathbf{x}) \\
&= \hat{\mathbf{x}}^T\hat{\mathbf{x}} - \hat{\mathbf{x}}^T\mathbf{x} - \mathbf{x}^T\hat{\mathbf{x}} + \mathbf{x}^T\mathbf{x} \\
&= \mathbf{x}^T\mathbf{A}\underbrace{\mathbf{A}^T\mathbf{A}}_{I}\mathbf{A}^T\mathbf{x} - 2\mathbf{x}^T\mathbf{A}\mathbf{A}^T\mathbf{x} + \mathbf{x}^T\mathbf{x} \\
&= \mathbf{x}^T\mathbf{x} - \mathbf{x}^T\mathbf{A}\mathbf{A}^T\mathbf{x} \\
&= \mathbf{x}^T\mathbf{x} - \mathbf{y}^T\mathbf{y} \\
&= \sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{p} y_i^2 \qquad p \ll n
\end{aligned}
$$

Data Set

7500 epoch of 10 sensors from Intel-lab



Simulation Senario



Packet

Packet Size

Field	Size
Header	10
Observation	2
Eigenvector element	4
Principal Component	4
Squared observation	4
Request for observation	1

Numerical Results

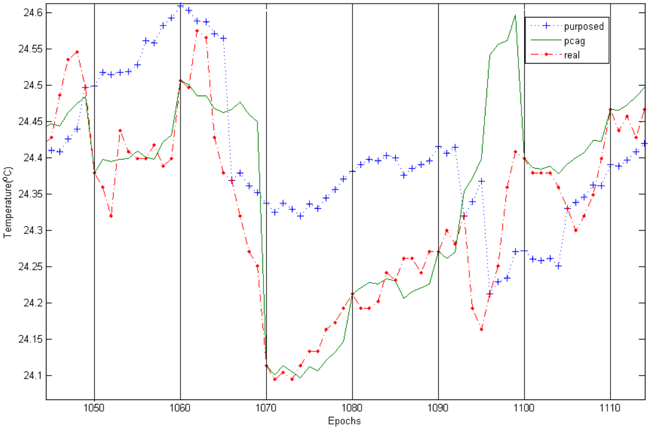
Numerical Results

	Transmitted Bytes	Error Mean	Error Variance	Error Max	Parameters
PCAg	200340	0.82	0.96	8.34	update: 11 epoch
LocalPCA	168114	0.82	0.07	1.43	threshold at 23:1.0 threshold at 22,21: 0.35
EAPCAg	215216	0.82	0.23	6.16	threshold:1.42

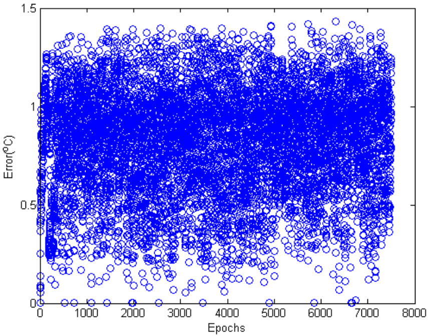
Discussion

- For the sensor node 23rd, EAPCAg method has sent about 15K bytes more than PCAg algorithm
- PCAg method has sent about 32K bytes more than LocalPCA.
- The average updating interval in EAPCAg is about 13.84 epochs which is around 3 epochs more than PCAg.
- EAPCAg has less efficiency than PCAg because the size of each PC packet is increased from 14 bytes to 18 bytes
- In PCAg and EAPCAg methods, when the number of sensor nodes increases, the eigenvector packets should be divided to smaller packets, so the amount of transmitted data will increase

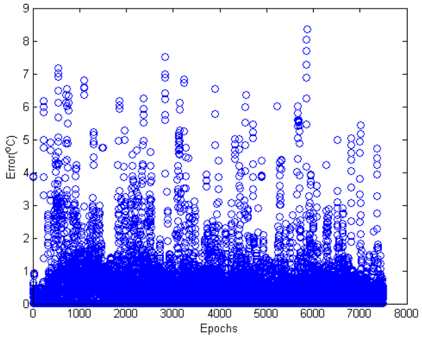
Comparision of LocalPCA with PCAg at the Base Station



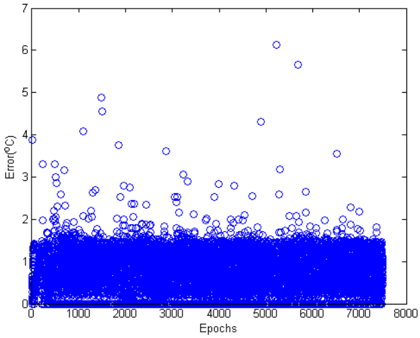
Error of LocalPCA



Error of PCAg



Error of EAPCAg



Future Works

PCA

- Efficiency of LocalPCA depends on thresholds
- Thresholds should be adjusted adaptively
- Weather data has low dimensions. LocalPCA will show better performance with high dimensional data (e.g. image)
- Kernel PCA is an non-linear version of PCA and can handle data which changes rapidly

- Borgne, Y. L., Raybaud, S., and Bontempi, G. Distributed principal component analysis for wireless sensor networks. Sensors 8 (2008)
- Hillol, K., Huang, W., Krishnamoorthy, S., Byung-Hoon, P., and Wang, S. Collective principal component analysis from distributed, heterogeneous data. Lecture Notes in Computer Science. 2000.
- Jollife, I. Principal Component Analysis, second ed. Springer, 2002.
- Y. L. Borgne and G. Bontempi. Unsupervised and supervised compression with principal component analysis in wireless sensor networks. in Proc. First International Workshop on Knowledge Discovery from Sensor Data, 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2007

Questions?

How to find eigenvectors

- Computation of Covariance Matrix
  $\boldsymbol{\Sigma} = E[(\mathbf{x} - E[\mathbf{x}])(\mathbf{x} - E[\mathbf{x}])^T]$
- Solve eigenvector, eigenvalue equation $\boldsymbol{\Sigma}\boldsymbol{\alpha} = \lambda\boldsymbol{\alpha}$
- if λ_i is largest eigenvalue and $\boldsymbol{\Sigma}\boldsymbol{\alpha}_i = \lambda_i\boldsymbol{\alpha}_i$
  then $\boldsymbol{\alpha}_i$ is dominant eigenvector.
- When the dimension reduction from n dimension to q dimension is efficient?
- Retained Variance: $H(q) = \frac{\sum_{k=1}^{q} \lambda_k}{\sum_{k=1}^{n} \lambda_k}$

Power Iteration Method

- Calculation of eigenvectors by eigenvector, eigenvalue equations is time consuming $O(n^3)$.
- Power Iteration Method approaches to dominant eigenvector very fast $O(n^2)$.
- PIM iteration: $\mathbf{g}^{k+1} = \boldsymbol{\Sigma}\mathbf{g}^k$
- FastPCA : Extension of PIM for calculation of other eigenvectors.