Code Design for Fast Selective Retrieval of Fusion Stored Sensor Network/Time-Series Data

Sharadh Ramaswamy, Jayanth Nayak and Kenneth Rose
Monitoring with Sensor N/Ws

http://faculty.cua.edu/elsarkawy/images/cluster.jpg


http://groups.csail.mit.edu/drl/wiki/images/d/db/amour_with_sensors.jpg

http://gis.clemson.edu/cpost/images/sensor_network.jpg

04/19/07

Signal Compression Lab, UCSB
Motivation

- Dense sensor N/Ws in many apps. (monitoring/tracking/surveillance)
- Correlated measurements are collected and stored
- Distributed compression for efficient data collection
- Optimal storage strategy?
Typical User/Query Model

2D Sensor Field

Database

Fusion Center

Correlated Data

(Region) Selective Queries

Sensor Motes
Fusion Storage vs. Selective Retrieval

Query $Q \sim P(q)$ (subset of sources)

Fusion Storage

Storage Database

Selective Retrieval

Correlated Sources $X_1, X_2, \ldots, X_M$

$X_{(Q)} = \begin{bmatrix} X_j \\ X_m \\ X_l \\ X_k \end{bmatrix}$

Storage Rate $R_s$ vs. Retrieval Rate $R_r$
Fusion Storage vs. Selective Retrieval

- Fundamentally a *storage* problem
- Challenge: exploit correlations to
  - min. storage rate
  - min. retrieval rate
- Conflicting objectives ?!
- Trade-off characterized in prior work (Nayak et al. ISIT’05)
Naive Storage Strategy I

- Jointly compress all sources
  - optimal in exploiting inter-source corr.
  - minimizes storage rate $R_s$

Lossless Coding $\Rightarrow$ $R_{s,\text{min}} = H(X_1, \ldots, X_M)$

- retrieves all stored data even for a small subset
- high retrieval rate/time!! $R_r = R_s$
Naive Storage Strategy II

- Compress and store every subset of sources separately
  - retrieves only minimum info. reqd.
  - optimal in retrieval rate/time

\[
R_{r,\text{min}} = \sum_q P(q)H(X_q)
\]

- reqd. storage grows with size of query set
- (combinatorially) high storage rate!!

\[
R_s = \sum_q H(X_q) >> H(X_1, \ldots, X_M)
\]
Storage with Distortion

- Perfect reconstruction not always necessary
- Trade-off - Storage Rate vs. Retrieval Rate vs. Distortion

- In practice, storage devices are of fixed capacity
- Practical design by fixing maximum allowable $R_s$

- Query-dependent “bit-selection” (and relevant codebooks) for selective retrieval…
 Proposed Fusion Coding Approach

Correlated Sources \( \{X_1, \ldots, X_M\} \) → Encoder \( \mathcal{E} \) → Bit-Selector \( \mathcal{S} \) → Decoder \( \mathcal{D} \) → Retrieved \( \hat{X}(Q) \)

- Storage Rate \( R_s \)
- Distortion \( D \)
- Retrieval Rate \( R_r \)

Query \( Q \sim P(q) \)
Mathematically…

Encoder: $\mathcal{E} : \mathcal{R}^M \rightarrow \mathcal{I} = \{0, 1\}^{R_s}$

Decoder: $\mathcal{D} : \mathcal{I} \times \mathcal{B} \rightarrow \hat{x}$

Bit-Selector: $\mathcal{S} : \mathcal{Q} \rightarrow \mathcal{B} = 2^{\{1, \ldots, R_s\}}$

$$D = \sum_{q \in \mathcal{Q}} P(q) \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} d_q(x, \hat{x}) \quad R_r = \sum_q P(q) R_q = \sum_q P(q) |S(q)|$$

Objective: $$\min_{\mathcal{E}, \mathcal{S}, \mathcal{D}} J = D(R_s) + \lambda R_r(R_s), \lambda \geq 0$$
Optimality and Design

Optimal Encoder: \( \mathcal{E}(x) = \arg \min_{i \in \mathcal{I}} \sum_{q} P(q) d_q(x, D(i, S(q))), \forall x \)

Optimal Bit-Selector: \( S(q) = \arg \min_{e \in B} \left\{ \frac{1}{|\mathcal{X}|} \sum_{x} d_q(x, D(\mathcal{E}(x), e)) + \lambda |e| \right\}, \forall q \)

Optimal Decoder: \( D(i, e) = \frac{1}{|F|} \sum_{x \in F} x, \forall e, i \)

Design by Gradient Descent

where \( F = \{ x : (\mathcal{E}(x))_e = (i)_e \} \).
Experimental Set-up

- Sensor data modeled as corr. Gaussian

\[ C = \sigma^2 \begin{pmatrix}
  1 & \rho & \cdots & \rho^{M-2} & \rho^{M-1} \\
  \rho & 1 & \rho & \cdots & \rho^{M-2} \\
  \vdots & \vdots & \ddots & \vdots & \vdots \\
  \rho^{M-2} & \cdots & \rho & 1 & \rho \\
  \rho^{M-1} & \rho^{M-2} & \cdots & \rho & 1 
\end{pmatrix} \]

- Stock market data (UCR Data-mining archive) – 93 stocks
Exponential Query Distribution

- Query distribution modeled as exponential in query size
- Queries of same size equally likely
- Distribution approximated by a training set of 335 queries
Correlated Gaussian $\rho=0.3$
Correlated Gaussian $\rho=0.8$
Stock Market Data

![Graph showing Stock Market Data](image-url)
Conclusions

- Fusion storage vs. selective retrieval
  - optimization of conflicting objectives

- Fusion Coder proposed for practical design

- Fusion Coder provides large gains over joint compression (VQ) schemes