Sensing in Social Spaces

Project Ideas (Continued)

Announcements/Reminders

- The class is set up on Piazza (search for CS 598 TAR)
  - You have been enrolled using your email in the official UIUC class roster.
- Project title, abstract, and member list is due September 17th.
A Deluge of Information!!!

The Age of Social Broadcast

The Past

O(n)  O(n)
The Age of Social Broadcast

O(n) \rightarrow O(n^2)

The Present

Real-time data grows much faster than our cognitive ability to consume it

Fast data growth!

Slow Evolution!
Real-time data grows much faster than our cognitive ability to consume it.

Fast data growth!

Slow Evolution!

Challenge: Extractive Summarization

Build a data service that allows applications to retrieve (extractive) data summaries at arbitrary levels of granularity in accordance with an application-specific redundancy metric.
Customizability: The Distance Metric

Data Object → Application Callback → Feature Vector → Difference Function → Distance Metric

Data Object → Application Callback → Feature Vector → Difference Function → Distance Metric

Opaque type (not interpreted by service)

Application specific functions (customization API)

Distance Metric (Must obey triangle inequality)
Customizability: The Distance Metric

![Diagram showing the relationship between data objects, feature vectors, and distance metrics.]

- Data Object
- Application Callback
- Feature Vector
- Difference Function
- Distance Metric

**Examples:** Scalars, vectors, pictures, text, etc..

Customizability: The Distance Metric

![Diagram showing the relationship between data objects, feature vectors, and distance metrics.]

- Data Object
- Application Callback
- Feature Vector
- Difference Function
- Distance Metric

(Must obey triangle inequality)
Customizability: The Distance Metric

Data Object

 Application Callback

 Feature Vector

 Difference Function

 Distance Metric (Must obey triangle inequality)

Data Object

 Application Callback

 Feature Vector

 Hierarchical Clustering

Summarization

Data Object

 Application Callback

 Feature Vector

 Difference Function

 Distance Metric (Must obey triangle inequality)

Data Object

 Application Callback

 Feature Vector

 Hierarchical Clustering
Summarization

Hierarchical Clustering

Data Object → Application Callback → Feature Vector → Difference Function → Distance Metric (Must obey triangle inequality)

Hierarchical Clustering
Summarization

- Data Object
  - Application Callback
  - Feature Vector
  - Difference Function
  - Hierarchical Clustering

Representative sampling versus noise reduction?

A Network Paradigm Shift
Communication → Information Distillation

Present Networks

**Goal:** Communication
- Maximizes bit throughput between end-points
- Most data is “logical”
- Protocols geared primarily for point-to-point communication
- Data loss may be a problem

Future Distillation Networks

**Goal:** Information Distillation
- Maximizes information flow
- Much data is “physical”
- Protocols geared for data filtering, and aggregation
- Data loss may be a feature intended to reduce less informative bits

The data fire-hose effect

The data fire-hose effect
A Primary Network Design Challenge

- How to build networks that maximize useful information flow from the physical world?

Information-maximizing Prioritization

- Determine transmission order?
Information-maximizing Prioritization

- Determine transmission order?

1

Information-maximizing Prioritization

- Determine transmission order?

1 2
Information-maximizing Prioritization

- Determine transmission order?

1 3 2

1 3 4 2
Information-maximizing Prioritization

- Determine transmission order?

Coverage-monotonic scheduling
Information-maximizing Prioritization

Note: Coverage can be defined in an abstract feature space

Coverage-monotonic scheduling

Example: Data Forwarding in Disruption-tolerant Networks

A big disaster strikes a city...

- Volunteers are recruited
- They scout the area, capture pictures and send them to a rescue center
- Network constraints prevent sending all pictures

Images are collected from the Internet

Hurricane Katrina 2005
Nepal earthquake 2015
Thailand flood 2011
Challenge: Data Selection to Maximize Coverage

Example of Bad Coverage

An Example of Poor Data Selection (Low Coverage)
Example of Good Coverage

An Example of Good Data Selection (High Coverage)

Fire on 6th and Main.
Collapse on Park Ave.
Flooding on State St.
Structural damage on Pier Square

A Scheduling Approach: Coverage-maximizing Priorities

- Implement coverage-maximizing in-network prioritization for forwarding and storage
  - Objects are forwarded/dropped in a priority order aimed to maximize coverage of delivered content
    - Objects similar to previously forwarded ones get lower priority
  - Challenge: Forwarding and dropping must be made aware of the degree of semantic redundancy (i.e., similarity) between objects
Failures in Complex Systems

When systems fail, a common goal is:
Localize and fix the root cause!

Another Thought

Individual software components are easy to "debug"
- Therefore, they are typically built reliably

Systems do not fail because of "bugs" localized to single components
- Systems fail because of unexpected interactions between many individually well-behaved components
- No single component is to blame
- No predicate over current state explains failure
- Unexpected sequences of events lead to problems
A “Kitchen” Analogy

Prepare the chicken, onions...

Add salt

Add seasoning

Marinate the chicken

Delicious chicken

It's time to stop the oven (4.00 pm)

Put in the oven (3.00 pm)

Who Burnt the chicken?

It's daylight saving, set the clock one hour behind

Put in the oven (3.00 pm)
A “Kitchen” Analogy

Prepare the chicken, onions...
Marinate the chicken
Put in the oven (3.00 pm)
Its time to stop the oven (4.00 pm)
Its daylight saving, set the clock one hour behind
Put in the oven (3.00 pm)

Moral of the story: Need to analyze sequences of events to identify the cause of anomaly

Problem: Diagnostics

Application

High level “Bad” behavior metric
Delay>5ms => Bad

Log collection front end

Execute n times

N Log files

Data preprocessing middleware

Frequent sequence generation algorithm

Good log K
Good log 1
Good log 2
Good log K

Delay<5ms

Frequent sequences in set of good logs

Frequent sequences only in set of good logs

Remove common sequences

Bad log L
Bad log 1
Bad log 2
Bad log L

Delay>5ms

Frequent sequences in set of bad logs

Frequent sequences only in set of bad logs
Tool Design Overview

Event Stream

Reduce Space of Multi-parameter Events

Filter Masked Events

Normal Count

Anomalous Count

Discriminative Event Calculation

Discriminative Sequence Mining

Diagnostic Feedback

Event mask file

Event clutter to remove

OK? Yes

No

Reliability Challenge: Avoidance

Fix it before it happens
Complexity Reduction

- Verification: Argue that all component interaction sequences are safe!
- In order to simplify analysis of interactions:
  - Must reduce overall dependencies
  - Dependencies must be “well-founded”: A critical component may use but should not depend on a less critical one

Complexity Reduction: Simplifying Dependencies

Reduce interactions and coupling
- Reduces propagation of local failures globally

Tightly coupled  Robustness solutions  Less coupled
The Performance/Robustness Trade-off

**Performance:** Exploring the edge of stability with global knowledge (global → more dependencies)

**Robustness:** Guaranteeing delivery in the face of adverse conditions and limited knowledge

Interactive Complexity in Cyber-Physical Systems

- Performance optimizations lead to:
  - Complex interactions (e.g., global versus local)
  - More dependencies
    - Deeper cascading failures
  - Lower robustness

- Cascading failure on "high-performance" road
- Non-cascading failure on side-street
Achieve both Performance and Robustness together?

The Simplex Architecture (by Lui Sha)

A simple verifiable core; diversity in the form of 2 alternatives; feedback control of the software execution.

```
Simple high assurance core

Data Flow Block Diagram

Complex high performance subsystem

System state

Decision
```

Example

- Joe is a new student who partied a bit too much. He masters bubble sort but only have 50% chance of writing a correct quick sort program.
- He must submit a program that will be evaluated as follows:
  - Correct and fast $O(n \log n)$: A
  - Correct but slow: B
  - Incorrect: F

- What is Joe’s optimal strategy?
Example

- Joe is a new student who partied a bit too much. He masters bubble sort but only have 50% chance of writing a correct quick sort program.
- He must submit a program that will be evaluated as follows:
  - Correct and fast \(O(n \log n)\): A
  - Correct but slow: B
  - Incorrect: F

![Quick Sort and Bubble Sort Diagram](image)

Prerequisite: the set of numbers to be sorted cannot be altered.

Project Ideas:
Robustness in Human-centric CPS

- **Ensure Robustness of:**
  - **Underlying physical resources:** A set of inter-dependent resource networks (e.g., for data transport, power, and physical mobility)
  - **Data communication and storage resources:** A digital plane that offers routing, storage, and capacity to access raw data
  - **Information resources:** Information filters for assessing quality of information and for filtering higher-quality information from raw data
  - **Inference processes:** Tools for modeling, estimation, and prediction of latent variables relevant to decision support