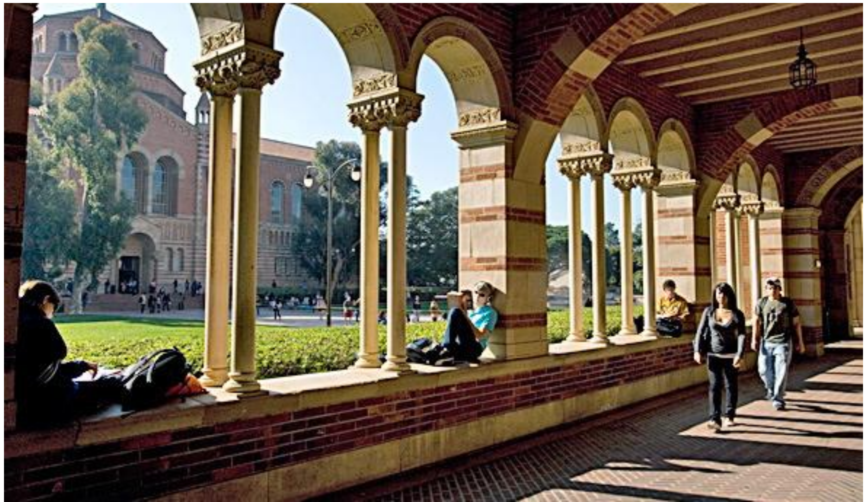
Finding it NOW: Stream Mining in Real Time

Prof. Mihaela van der Schaar

Electrical Engineering Department, UCLA

http://medianetlab.ee.ucla.edu/



Outline

- **What** is stream mining?
  Why should multimedia engineers care?
  - **Stream Mining: new processing paradigm**
  - Motivating applications
  - Distributed, real-time stream processing - **Challenges**

- A novel, systematic framework for real-time knowledge extraction
  - Models, problem formulation, solutions
  - Real-time topology construction
  - Decentralized framework for optimizing stream mining systems

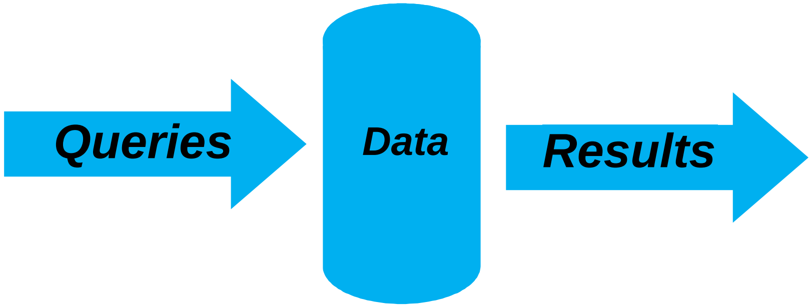
- **Research opportunities**

Knowledge Extraction Paradigm

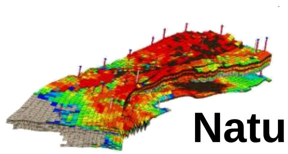
Traditional Search (Mining)

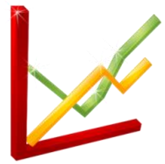
Historical fact finding with data-at-rest

- Batch paradigm, pull data model
- Query-driven: submits queries to static data
- Relies on Databases, Data Warehouses



Emergence of Large-Scale, Real-Time Multimedia Stream Mining Applications

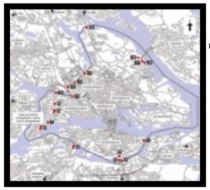




Stock market
- Impact of weather on securities prices
- Analyze market data at ultra-low latencies

Natural Systems
- Seismic monitoring
- Wildfire management
- Water management

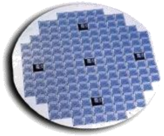
Law Enforcement
- Real-time multimodal surveillance



Transportation
- Intelligent traffic management

Fraud prevention
- Detecting multi-party fraud
- Real time fraud prevention

Manufacturing
- Process control for microchip fabrication



Radio Astronomy
- Detection of transient events

Health & Life Sciences
- Neonatal ICU monitoring
- Epidemic early warning system
- Remote healthcare monitoring

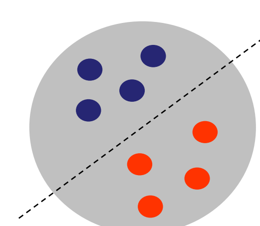
Telecom
- Processing of Call Detail records
- Real-time services, billing, advertizing
- Business intelligence
- Churn Analysis, Fraud Detection

Application 1:
Semantic Concept Detection in Multimedia

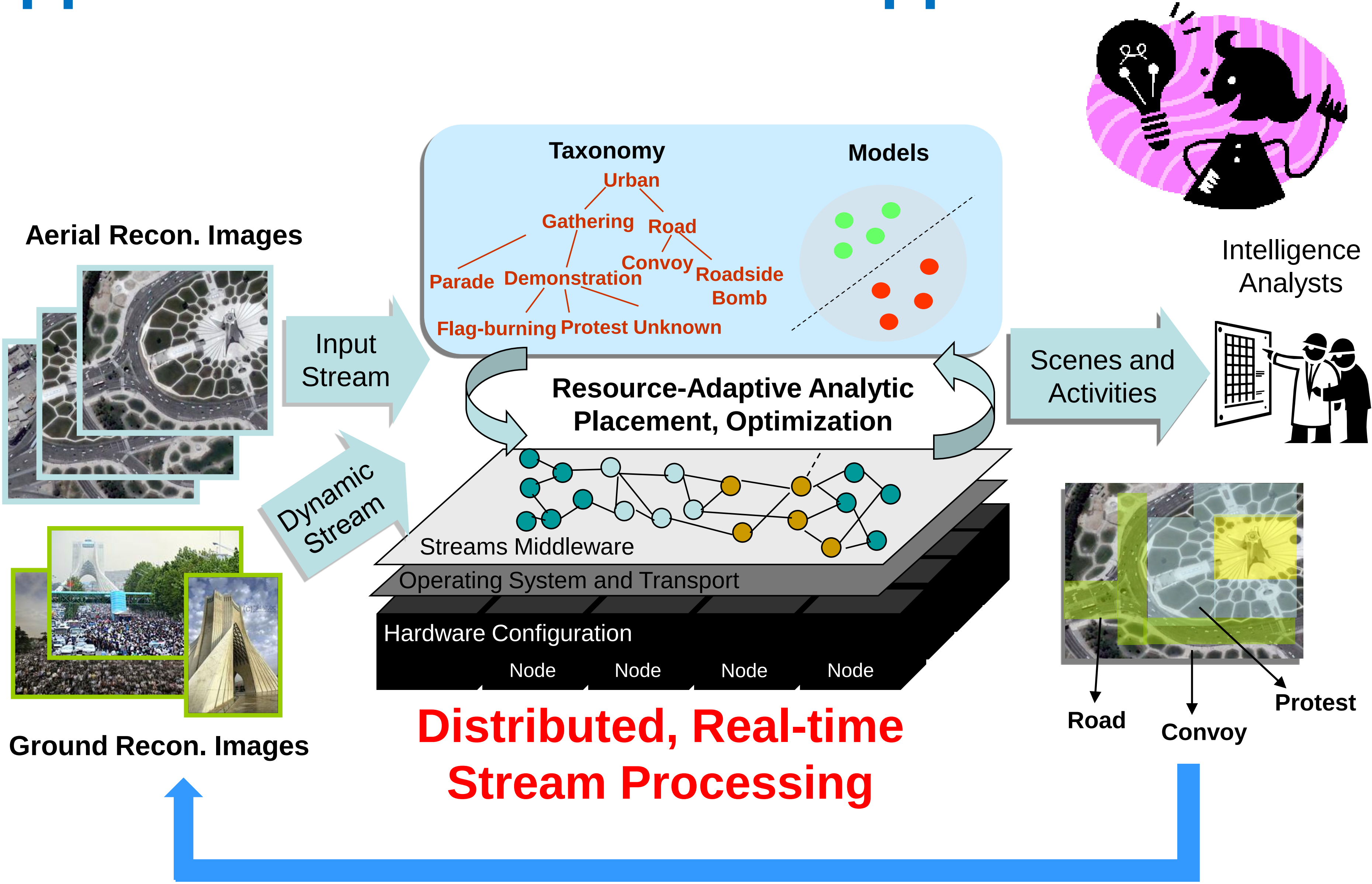
- Automatically categorize image and video into a list of concepts
  - Statistical learning methods and multimodal features

How to extract knowledge?

Preprocess, clean, filter, analyze,
learn models, predict, track, correlate, explore

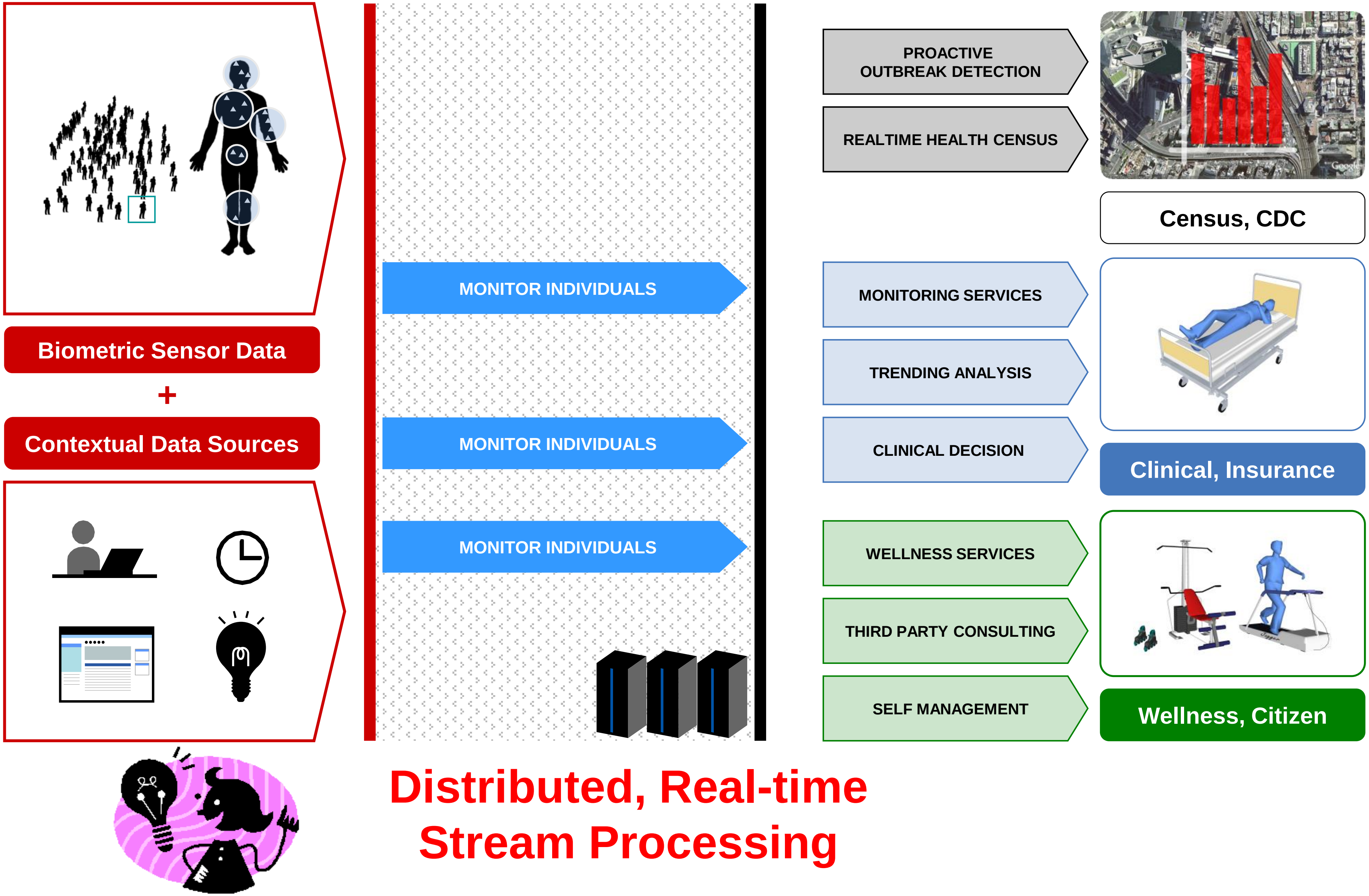


Application 2: Surveillance applications



Distributed, Real-time Stream Processing

Application 3: Online Healthcare Monitoring



Application 4: Analysis of Social Media

- Graph with nodes as people (e.g. Bloggers) and links represent interactions
  - Each node includes a temporal sequence of ‘documents’ (blog posts, tweets, …)

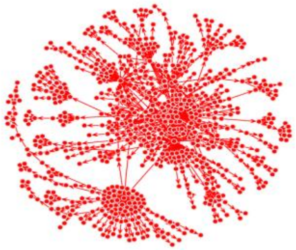
2a. Identify key influencers
Now: page rank, SNA measures, …

2b. Characterize viral potential
Now: use of follower statistics

INFLUENCE

1. Identify relevant content
Now: keyword search

RELEVANCE



3. Characterize objective/subjective content
Now: lexical and pattern-based models

SUBJECTIVITY

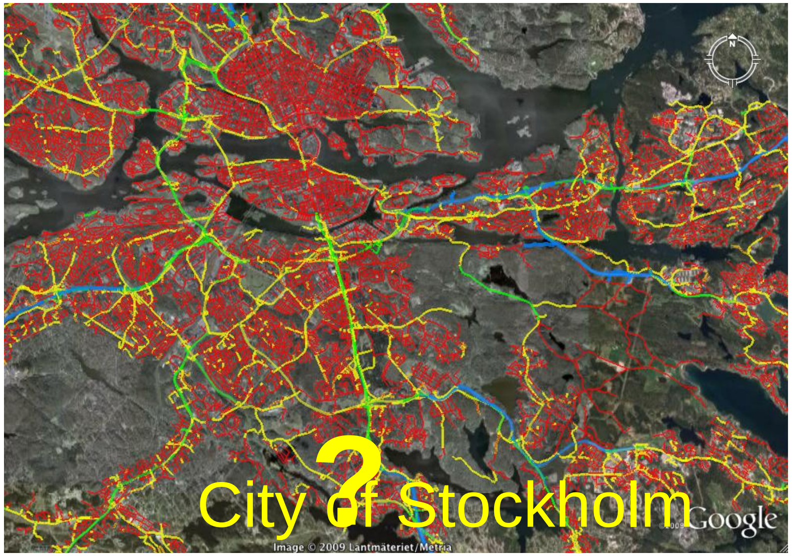
4a. Topic evolution & emergence
Now: word co-occurrence, clustering

4b. Classify new partially-observed documents
Now: unsupervised clustering

TOPIC IDENTIFICATION AND CLASSIFICATION

Distributed, Real-time Stream Processing

Application 5: Intelligent Traffic Monitoring



- **Applications**
  - Real Time Traffic Monitoring
  - Real Time Traffic Information
  - (Multimodal) Travel Planner
  - Anomaly Detection and Prediction
  - Infrastructure Planning
- **Available Multimodal Data Streams**
  - GPS
  - Cell-phones (location tracking)
  - Public Transport (bus, docking)
  - Pollution measurements
  - Weather Conditions (including road conditions)
  - Optical traffic flow detectors
  - Travel time data based on plate recognition
  - Accidents in network as they are being recorded
  - Road closures (road work, etc.)
  - Still pictures from road cameras

What “makes” a Streaming Application?

- **Complex and Heterogeneous Data Input**
  - Distributed data sources/sensors
  - Various data formats
  - Wide Range of Data Rates
    - Manufacturing: 5-10 Mbps, Astronomy: ~x00 Gbps, Healthcare: ~x00 Kbps per patient
  - Correlated primary sources
    - Signal-level correlation and Semantic-level correlation
  - Highly noisy and lossy data
    - Environmental noise and data loss
  - Structured or Unstructured
    - Documents, emails, transactions, digital video/audio data, RSS feeds, tickers, time series
  - Data decomposable into meaningful units
  - Data has timeliness **(streaming data)**
    - **Cannot store and process & Need to act NOW!**

What "makes" a Streaming Application?

- **Stream Data Analysis**
  - Real-time, open-loop or closed-loop, and long-running
  - Sub inquiries based on task portion, modality, progressive confidence, regional or temporal aspects
  - Exploration with potentially <u>unknown</u> or multiple versions of "ground truth"

- **High Performance**
  - Low-latency
  - High throughput
  - Scaling to massive data sources, concurrent processing
  - Distributed processing across non-collocated processing elements
  - Fault tolerant

The 5 Key Challenges of Stream-Mining

- *High Volume* of data: faster than a database can handle
- *Time Sensitivity*: responses required in real-time with low latency
- *Computational-efficiency*: massive data processing consumes enormous resources
- *Distributed information and knowledge extraction engines*
- *Complex Analytics*: correlation from multiple sources and/or signals; video, audio, graphics or other non-relational data types

Multi-Disciplinary Research Needed

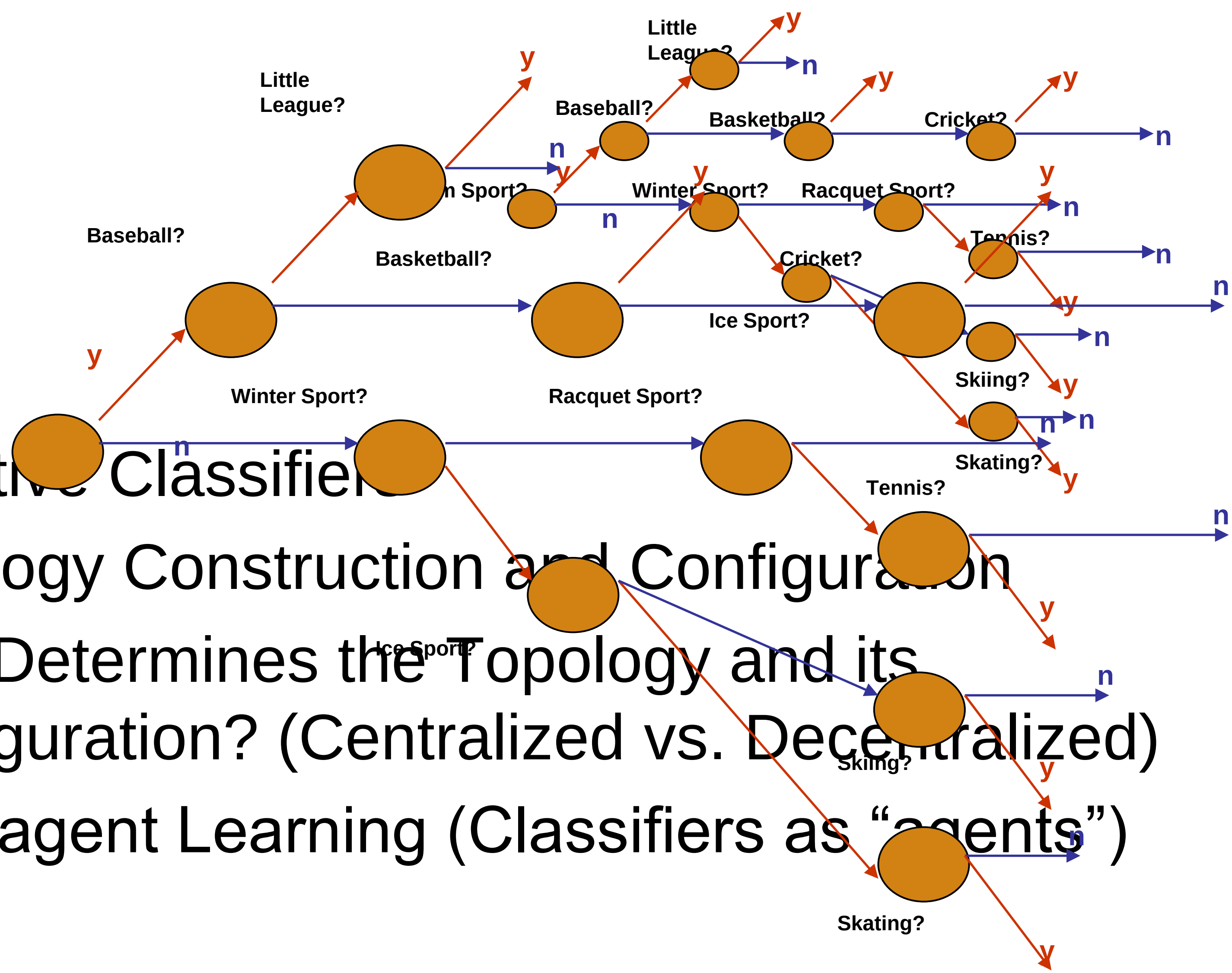
- Parallel and Grid Computing
  - High volume data stream processing
- Content-level Routing and Event Messaging
- Databases
- Application development
  - Simple methods to design and construct knowledge-extraction applications

- **Multimedia Signal Processing**
  - Real-time adaptive analytics
    - Stream data capturing, pre-processing, filtering, data reduction and data compression, processing, summarization
  - Real-time data mining
    - Stream mining, incremental learning, online learning, cooperative learning
  - **Cross layer design, networking and optimization**
    - **System and Analytics**

A novel, systematic framework for real-time knowledge extraction

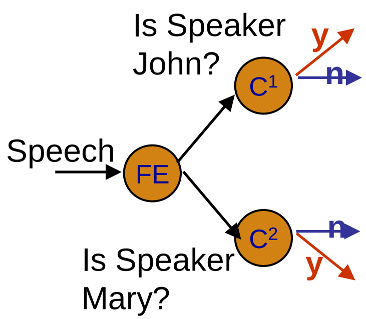
(partially implemented in IBM System S)

Knowledge extraction: stream mining done using an ensemble of classifiers

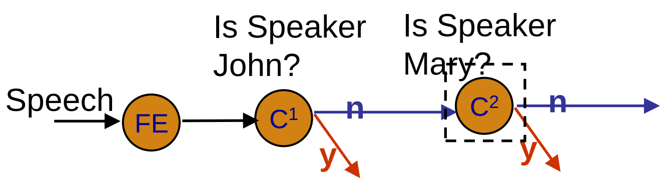


- Adaptive Classifiers
- Topology Construction and Configuration
- Who Determines the Topology and its Configuration? (Centralized vs. Decentralized)
- Multi-agent Learning (Classifiers as “agents”)

Multiple topologies possible



70% of speech from John, 20% from Mary, 10% unknown



“Mary” detector needs to process ~30% of data on average

Multi-concept (binary) detection

Hierarchical multi-concept (binary) detection

☺ Significant Advantage in Resource Constrained Scenarios

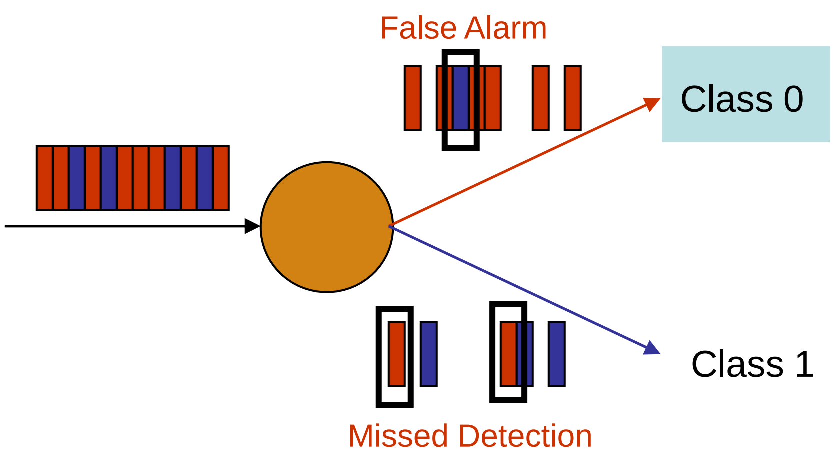
☺ Hierarchy can account for data density skew ⇒ limit unnecessary data processing by downstream classifiers

☺ Improve Classification Accuracy & Error Tolerance

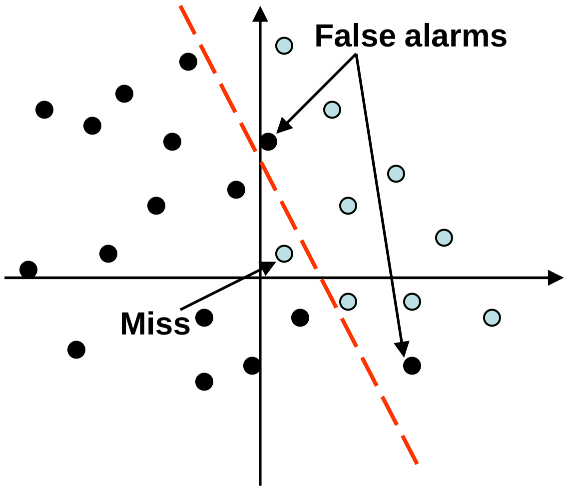
Classifiers make mistakes

Binary Classifiers:

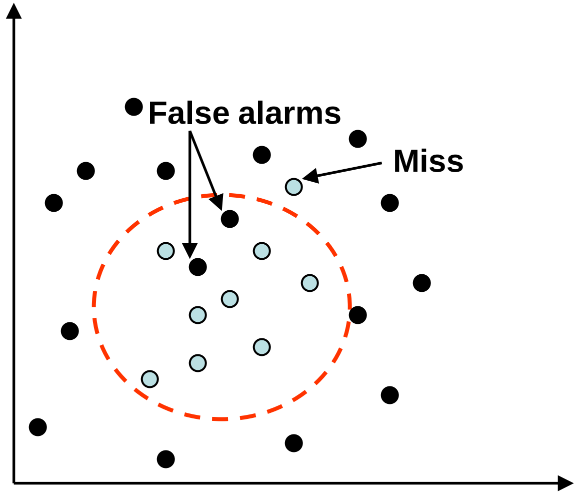
Filter Data into Two Classes



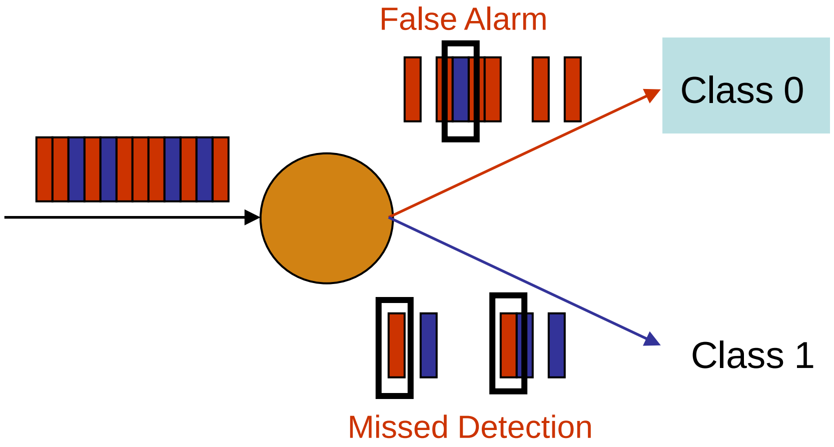
SVM: Linear Kernel Function



SVM: Radial Basis Kernel Function

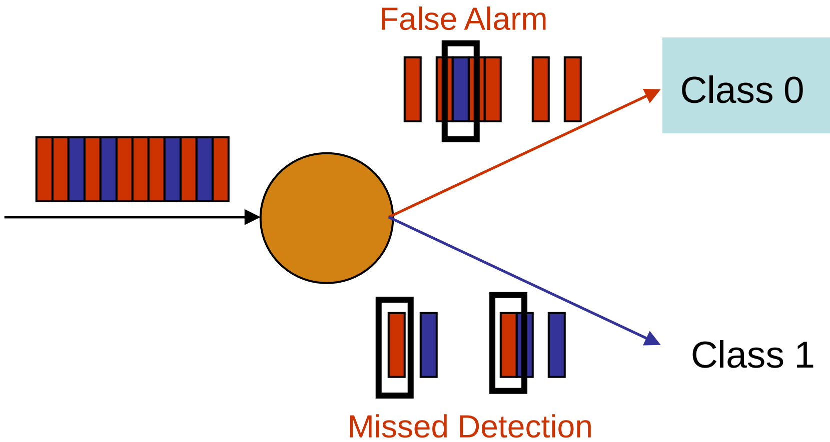


Binary Classifier Model



DET curve relates misses and false alarms.
Can parameterize operating point by *PF*.

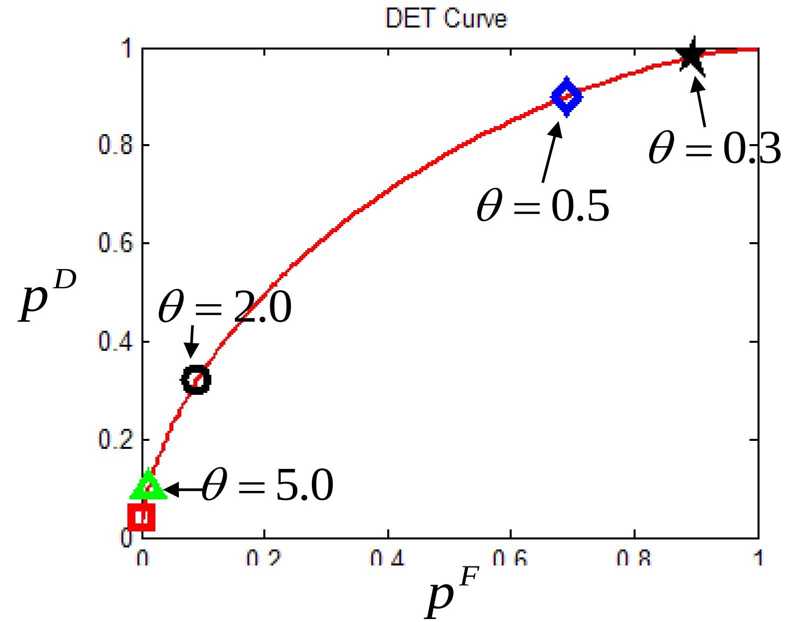
Binary Classifier Model



$$U = p^D - \theta p^F$$

Detection Error Tradeoff (DET)

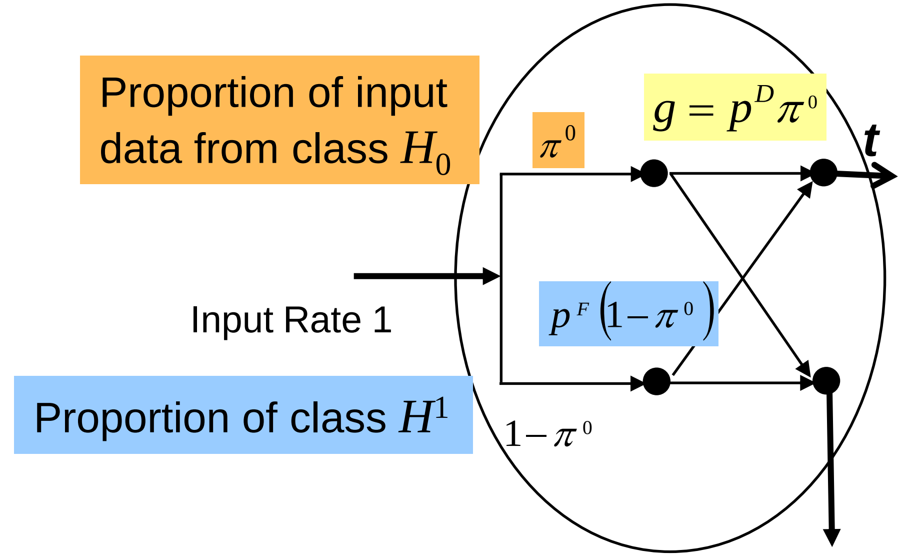
Classifier Accuracy Characterization
DET curve i.e. p^D versus p^F curve



Operating point on curve determined by desired tradeoff between p^D and p^F

$$\left(p^D, p^F\right) = \left(f\left(p^F\right), p^F\right)$$

Terminology



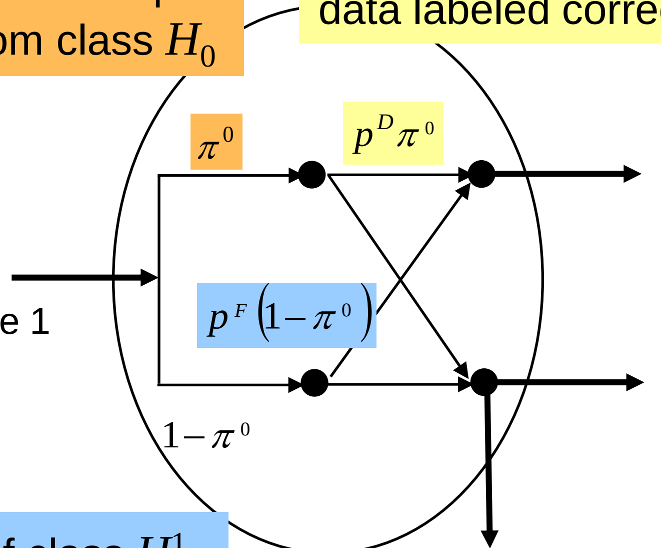
- ***A priori*** **Probabilities**
  - Unconditional probability (π) of data belonging to class H^0 or H^1 for a classifier
  - Note: these *a-priori* selectivities are inherent to the data features and to the relationships between concepts; thus, they do not depend on the operating points of individual classifiers
- **Throughput** (t) - Total rate output by classifier
- **Goodput** (g) - Correctly classified portion of throughput

Binary Classifier Model

Data from two classes: H^0 and $H^1 \rightarrow$ Consider data **labeled** as class H^0

Proportion of input data from class H_0

Proportion of class H_0 data labeled correctly



Proportion of class H^1 data labeled incorrectly

Throughput $t = \left[\pi^0 p^D + (1-\pi^0) p^F\right]$

Goodput $g = \pi^0 p^D$

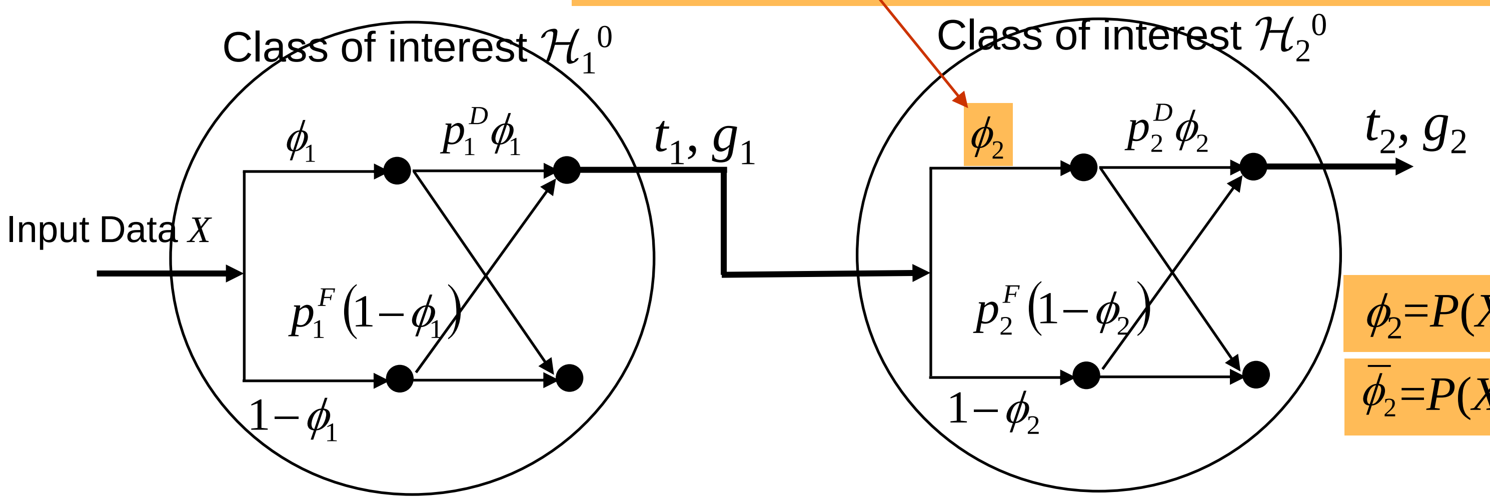
Can similarly define throughput $\bar{t}$ and goodput $\bar{g}$

Data with desired attribute forwarded
Other data discarded

Selected operating point/s control throughput & goodput

Cascade of Binary Classifiers

Conditional probability of data belonging to class $\mathcal{H}_2^0$ given it belongs to class $\mathcal{H}_1^0$



$\phi_2 = P(X \in \mathcal{H}_2^0 | X \in \mathcal{H}_1^0)$

$\bar{\phi}_2 = P(X \in \mathcal{H}_2^0 | X \notin \mathcal{H}_1^0)$

$$\begin{bmatrix} t_2 \\ g_2 \end{bmatrix} = \begin{bmatrix} p_2^F + \phi_2 (p_2^D - p_2^F) & \phi_2 (p_2^D - p_2^F) \\ 0 & \phi_2 p_2^D \end{bmatrix} \begin{bmatrix} t_1 \\ g_1 \end{bmatrix}$$

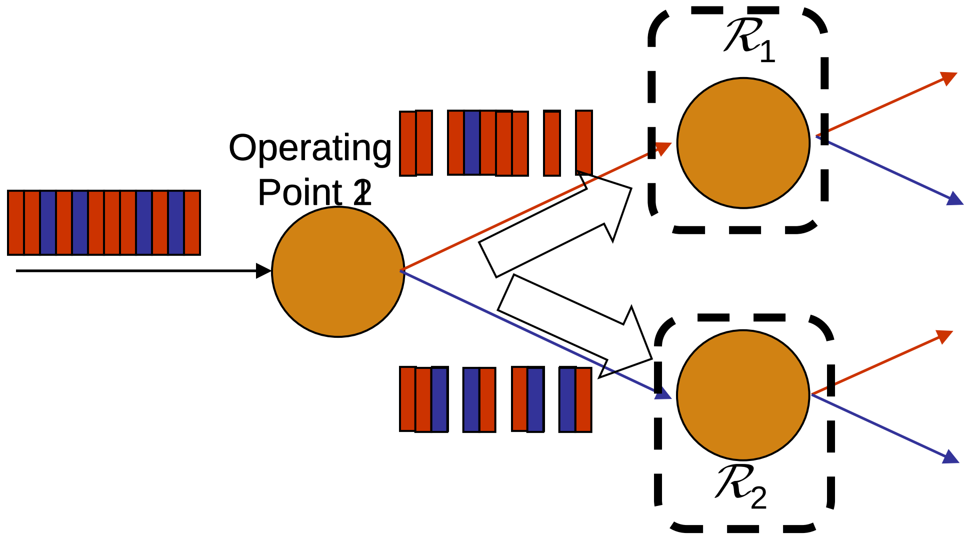
Exclusivity $\Rightarrow$ what is not of interest to parent, is not of interest to child classifier

Recursive relationship in throughput and goodput among classifiers

Classifier Resource Consumption Model

Classifier complexity → Underlying Model Complexity → Complexity measured in terms of processing (CPU and/or Memory) resource requirements

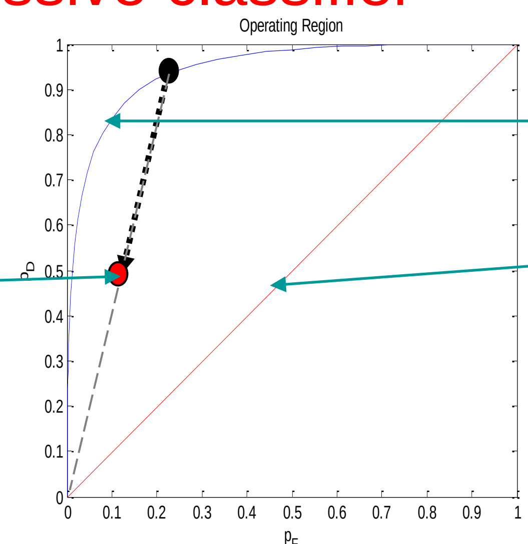
Topology of Classifiers



It may not be feasible to meet tight resource constraints – **Solutions?**

Operating point controls rate flowing through (resource consumption of) each successive classifier

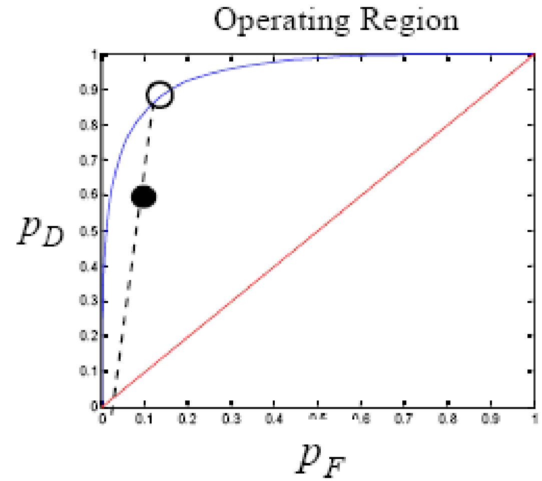
Arbitrary Load Shedding
(algorithms determine
a discard policy given the
observed data characteristics)



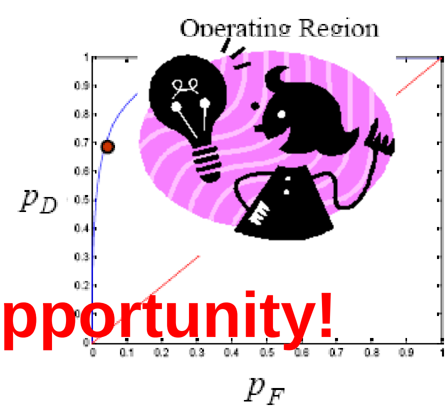
DET curve

Random data forwarding curve

Load-shedding vs. multiple operating point

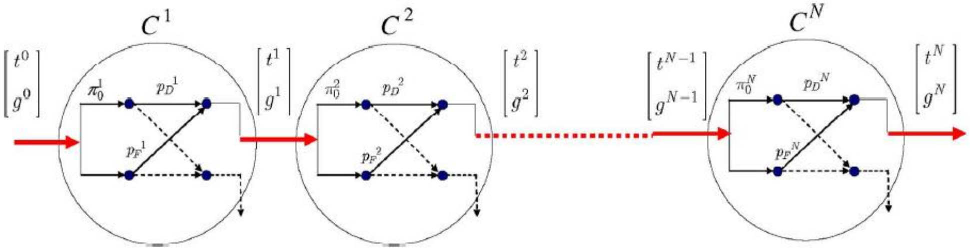


- Load-shedding based solution
  – formalize problem as network optimization problem and rely on load-shedding to decide what fraction of data to process
- Proposed – multiple operating points
  – determine how the available data should be *processed* (scalable processing) given the underlying resource allocation
  – allow individual classifiers in the ensemble to select different operating points
    - use a separate threshold for yes and no output edges
    - intelligent load shedding and replication of data
  – disadvantage: higher complexity

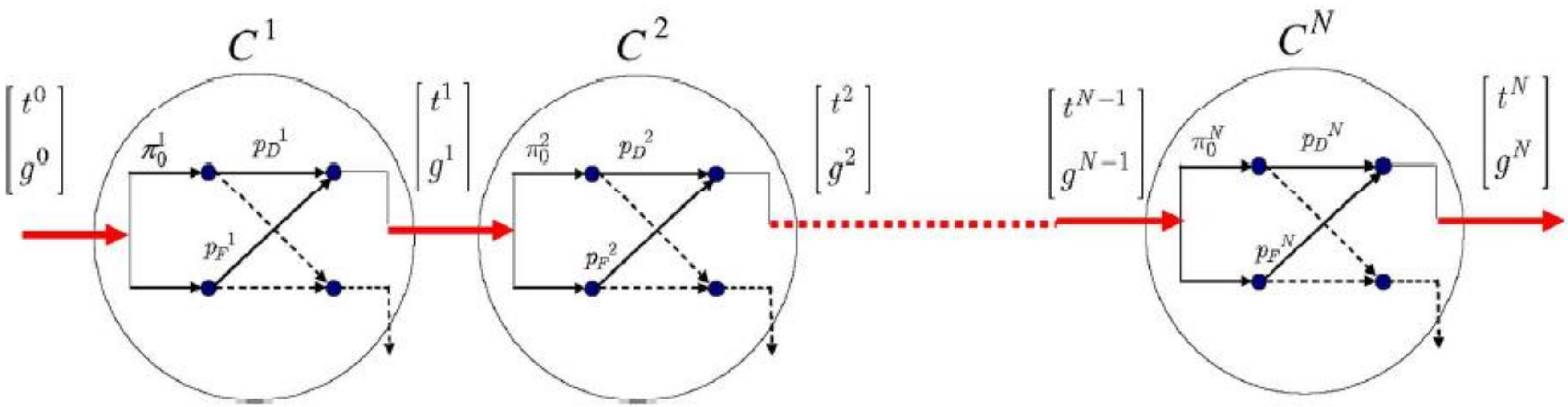


Adaptive classifiers needed - Research opportunity!

End-to-End Classification Accuracy: Chain



- Consider a terminal classifier C_N
  - False-Alarm Rate: t_N-g_N
  - Missed Detection rate: $t_0\pi_N$-g_N
    - t_0: Initial throughput, π_N:Apriori probability



$$c = c_k\,(t_N - g_N) + c_N\,(t_0\pi_N - g_N)$$

Falsely classified

Missed

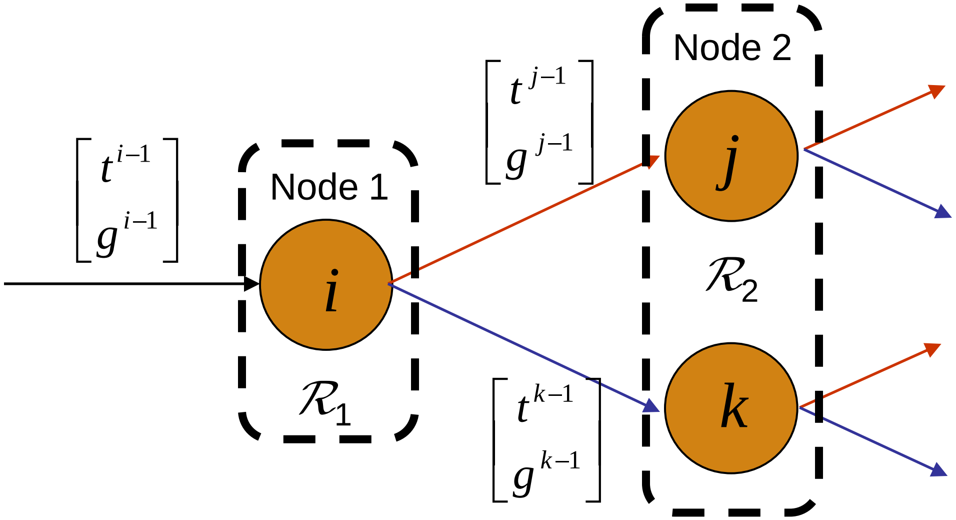
Tree

Delay and Resource Consumption

- Per-classifier delay for classifier C_k
  - $\beta_k t_{k\text{-}1}$
- End-to-end delay for a chain with N classifiers

$$\boxed{\sum_{k=1}^{N} \beta_k t_{k-1}}$$

Placement and Resource Constraints



$$r_i = \alpha_i t_{i-1} \le \mathcal{R}_1$$

$$r_j + r_k = \alpha_j t_{j-1} + \alpha_k t_{k-1} \le \mathcal{R}_2$$

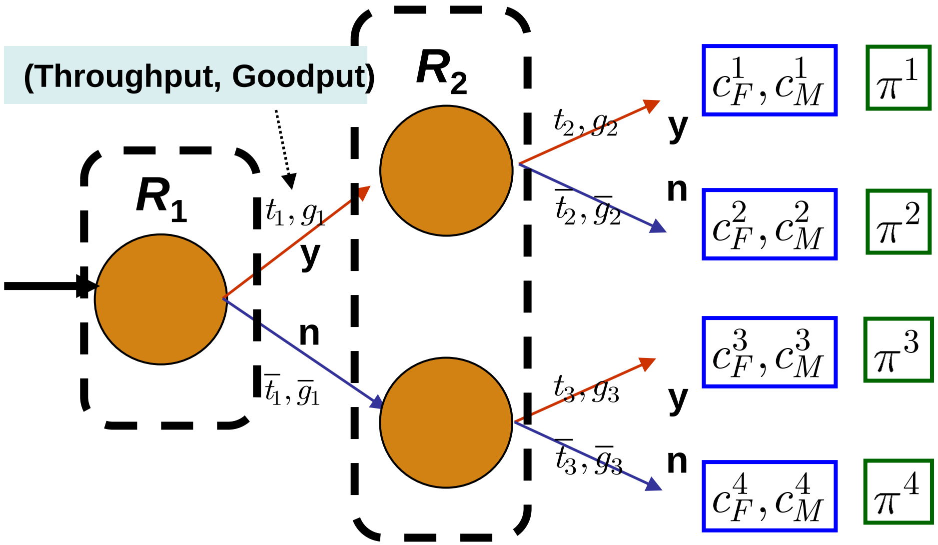
Combined resource constraints as:

Placement matrix **A**

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} \alpha_i t_{i-1} \\ \alpha_j t_{j-1} \\ \alpha_k t_{k-1} \end{bmatrix} \le \begin{bmatrix} \mathcal{R}_1 \\ \mathcal{R}_2 \end{bmatrix}$$

Problem: Resource-constrained real-time stream mining

- Given
  - Costs of misclassification (c_M, c_F) per data object per class
  - True fraction of data in each class π^k
  - Placement and Resource constraints
  - Throughput and Goodput
- **Objective**
  - **Minimize end-to-end misclassification cost**
  - **Satisfy resource constraints**



$$\text{minimize } c = \sum_{k=1}^{K} \left[c_F^k \cdot \text{false\_alarms} + c_M^k \cdot \text{misses} \right]$$

$$= \sum_{k=1}^{K} \left[c_F^k \quad t^k - g^k \quad + c_M^k \quad \pi^k - g^k \quad \right]$$

$$\text{s.t.} \quad \mathbf{Ah} \quad \mathbf{p}_F \quad \leq \mathbf{R}$$

$$0 \leq \mathbf{p}_F \leq 1$$

Problem: Resource-constrained real-time stream mining

- **Solutions**
  - Topology Construction
  - Topology Configuration
  - Joint Topology Construction and Classifier Configuration
  - Multiple concurrent queries
  - Optimization
    - Centralized vs. Distributed; Benevolent vs. Strategic
  - Multi-agent Learning

Topology Construction – Chains: Problem Formulation

For a permutation σ of N classifiers in chain, i.e. $\sigma \in Perm(N)$

For operating point for each classifier $\left(p_i^D, p_i^F\right) = \left(f\left(p_i^F\right), p_i^F\right)$

End to end misclassification penalty $c_{err}^{\sigma,\mathbf{p}^F} = c^M\left(\pi t_0 - g_N^{\sigma,\mathbf{p}^F}\right) + c^F\left(t_N^{\sigma,\mathbf{p}^F} - g_N^{\sigma,\mathbf{p}^F}\right)$

Total missed detection

Total false alarms

End to end processing delay $c_{delay}^{\sigma,\mathbf{p}^F} = \sum_{k=1}^{N} \boxed{\beta_{\sigma(k)}} t_{k-1}^{\sigma,\mathbf{p}^F}$

Time per unit rate of data for classifier $\sigma(k)$

Select operating point per classifier and order them to minimize misclassification penalty and delay

$$\min_{\sigma,\mathbf{p}^F} C\left(\sigma,\mathbf{p}^F\right) = c_{err}^{\sigma,\mathbf{p}^F} + \lambda c_{delay}^{\sigma,\mathbf{p}^F}$$

Ordering with ***Fixed*** Operating Points

- Maps onto pipeline ordering problem
  - Shown to be NP complete by mapping onto set-cover problem
- With fixed operating points
  - Goodput is independent of selected order => can be dropped from the optimization
- Greedy algorithms – proposed

Intuition: Select classifiers with high selectivity and small processing cost as early in chain as possible

Select order to minimize misclassification penalty and delay

$$\min_{\sigma} C(\sigma) = \sum_{h=0}^{N} \mu_{\sigma(h)} t_h^{\sigma}$$

$$\mu_{\sigma(h)} = \begin{cases} \dfrac{\lambda \beta_{\sigma(h+1)}}{c^F + c^M} & h \leq N-1 \\ \dfrac{c^F}{c^F + c^M} & h = N \end{cases}$$

relative price of classifier delaying the answer to the query, relative to missing deadline

Operating Point Selection for ***Fixed*** Order

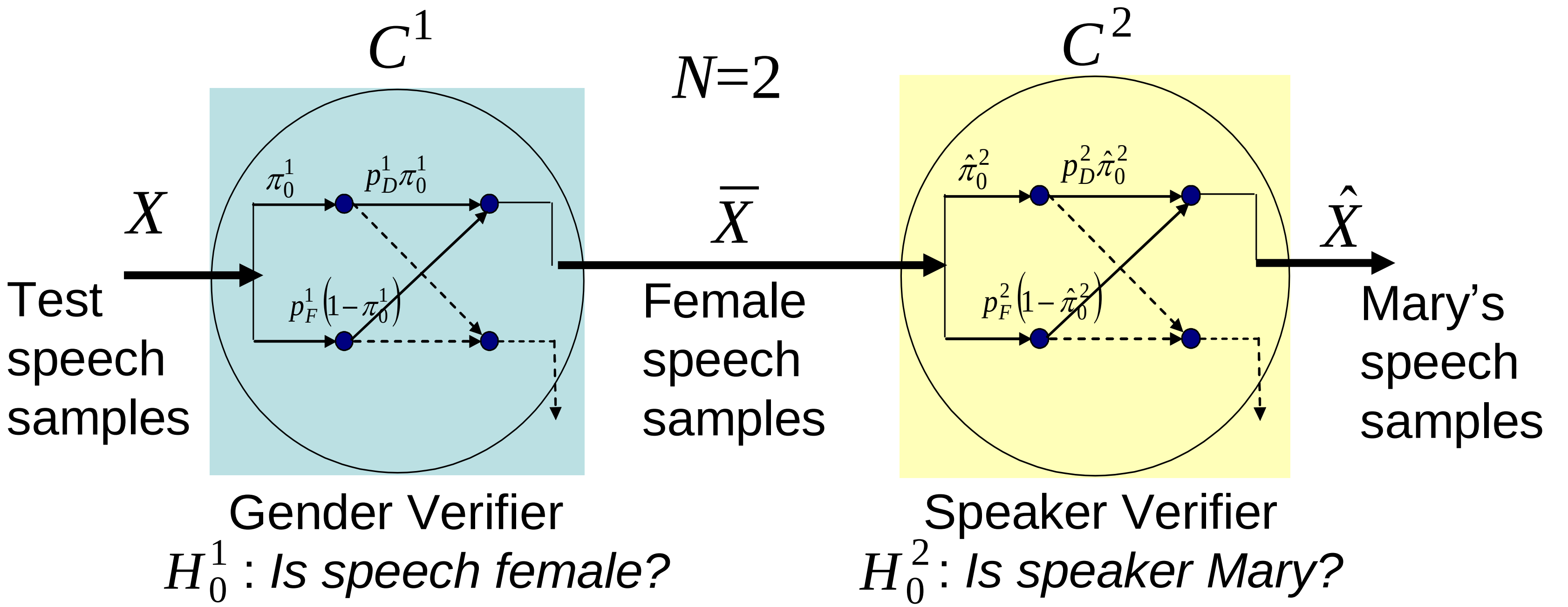
- No impact on delay of operating point selection – for fixed order
- Impacts resource consumption
- Resource constrained problem formulation

Select operating point to minimize misclassification penalty

$$\min_{\mathbf{p}^F} c_{err}^{\mathbf{p}^F} = c^M\left(\pi t_0 - g_N^{\mathbf{p}^F}\right) + c^F\left(t_N^{\mathbf{p}^F} - g_N^{\mathbf{p}^F}\right) \quad \text{s.t.} \quad \sum_{k=1}^{N} \alpha_k t_{k-1} \le \mathcal{R}$$

Solution: using gradient descent based optimization techniques

Illustrative Experimental Results



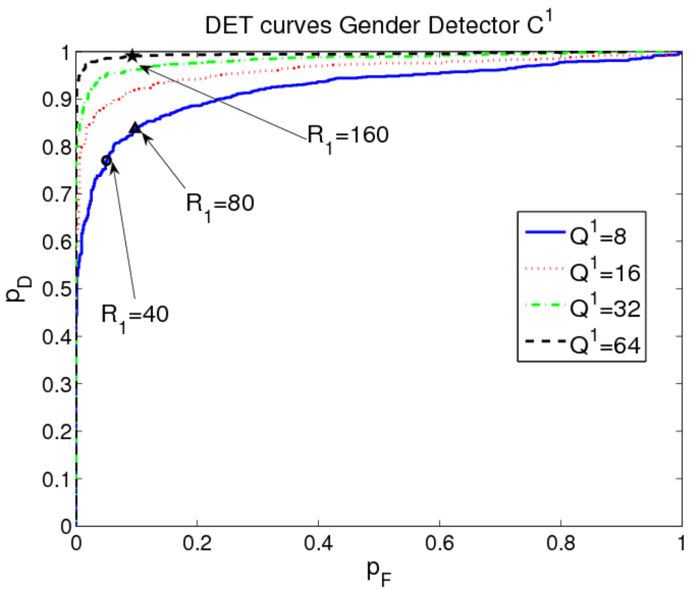
Dataset: **Real Telephony Speech Data Switchboard 2 Phase 2**

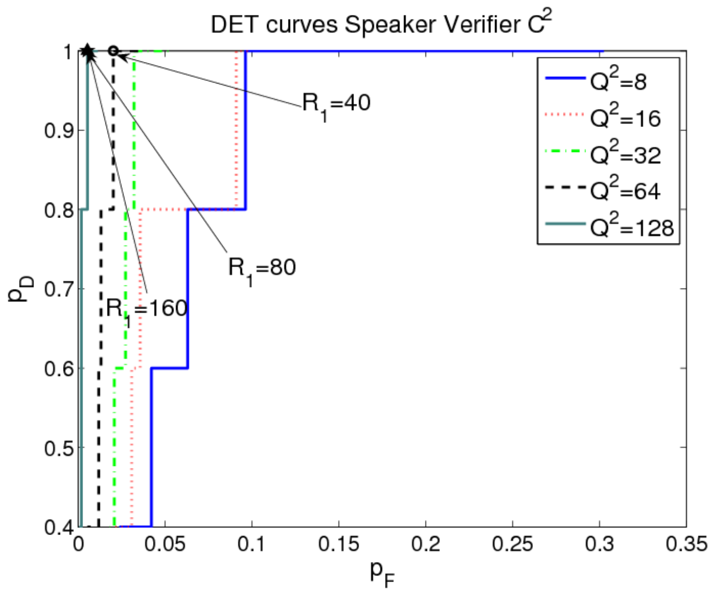
Train Set (41 male, 59 female), Test Set (106 male, 176 female)

$$P\left(X \in H_0^1\right) = \frac{176}{282} \quad P\left(X \in H_0^2\right) = \frac{1}{282} \quad P\left(X \in H_0^2 \mid X \in H_0^1\right) = \frac{1}{176}$$

Illustrative Experimental Results

Classifier Cascade versus Single Classifier under Resource Constraints





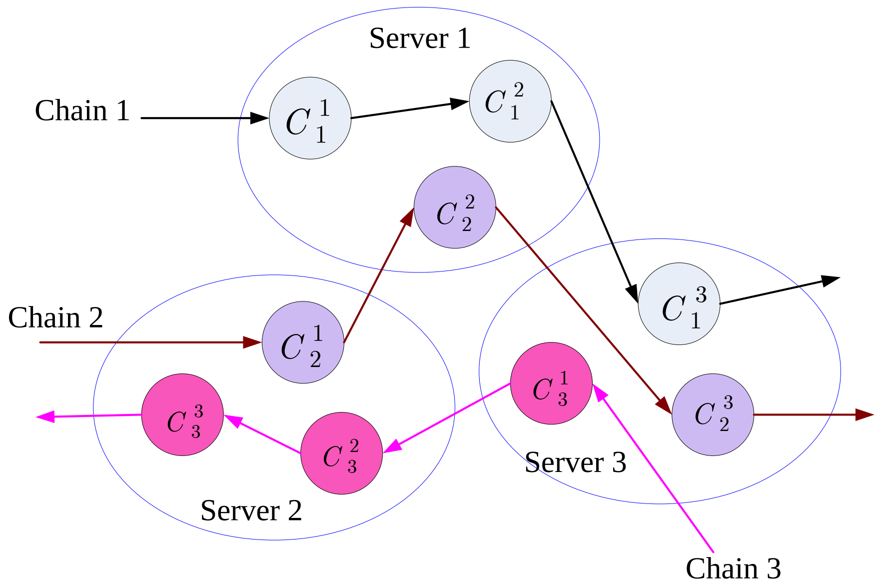
$U = p^D - \theta p^F$

Classifier Topology	$\mathcal{R}_1 = 40$	$\mathcal{R}_1 = 80$	$\mathcal{R}_1 = 160$
Cascade: $C^1 \rightarrow C^2$	0.17	0.27	0.32
No Cascade: C^2	0.03	0.15	0.30

Conclusions: Significant improvements with a cascade under resource constraints

Operating Point Selection: Multiple Chains

Collection of $\mathcal{I}$ chains deployed on shared nodes



Problem Formulation

$$\max_{\mathbf{p}^F} \sum_{i=1}^{\mathcal{I}} \ln\left(U_i\left(\mathbf{p}_i^F\right)\right)$$

$$s.t. \sum_{i=1}^{\mathcal{I}} \mathbf{A}_i \mathbf{h}_i\left(\mathbf{p}_i^F\right) \leq \mathcal{R}^{tot}$$

$$U_i\left(\mathbf{p}_i^F\right) > 0$$

$$0 \leq \left(\mathbf{p}_i^F\right)_k \leq 1 \quad \forall i, k$$

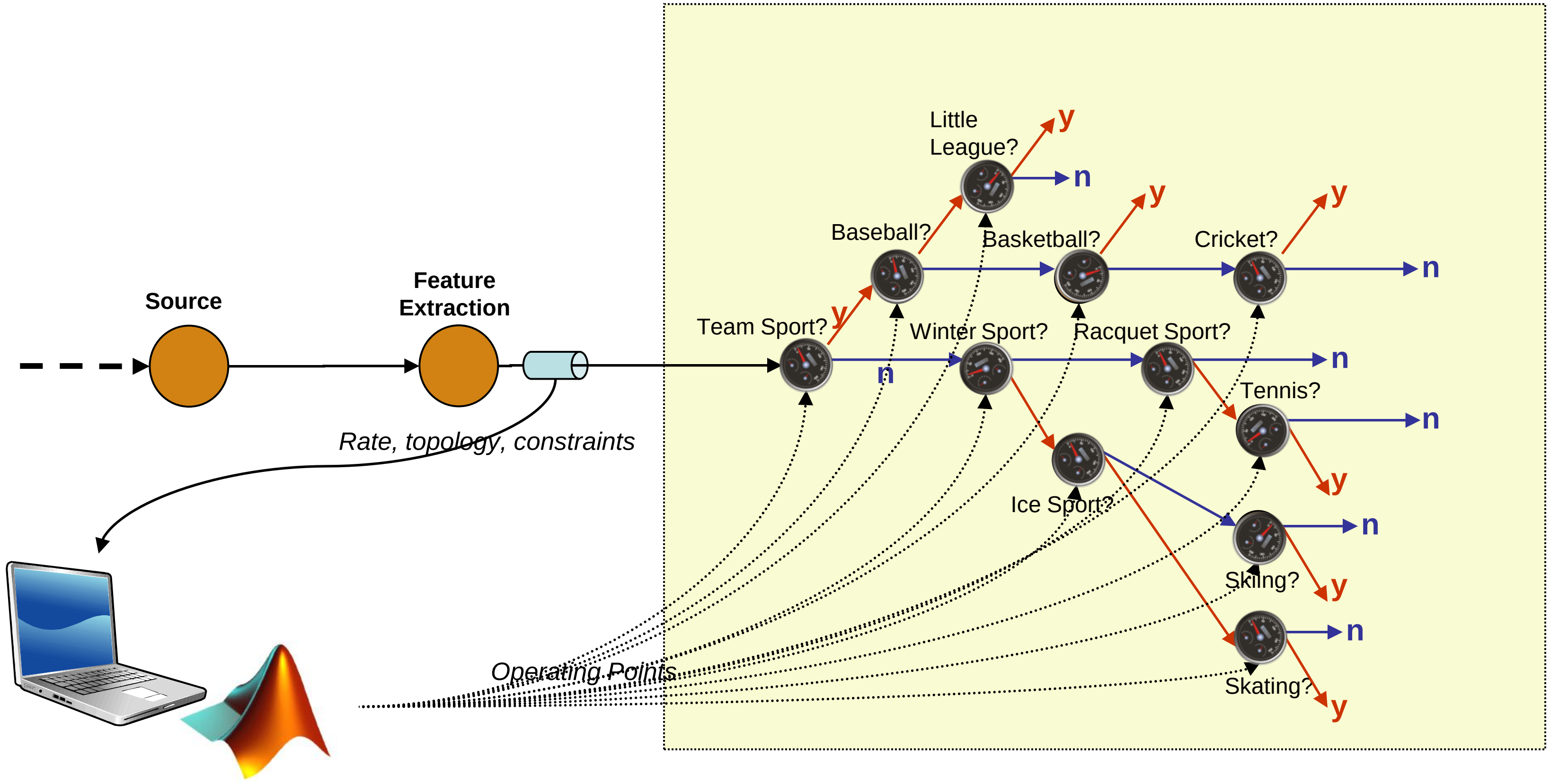
Solution: Sequential Quadratic Programming (SQP)

SQP - models the nonlinear optimization problem at each iteration by an approximate quadratic programming (QP) subproblem.

Solution to the approximate QP subproblem is then used to construct a better approximation at the next iteration. Procedure is repeated to create a sequence of approximations converging to optimal solution.

Note: SQP algorithm may converge to a *locally* optimal solution

Solution Outline



Iterative Algorithm using Sequential Quadratic Programming (SQP).

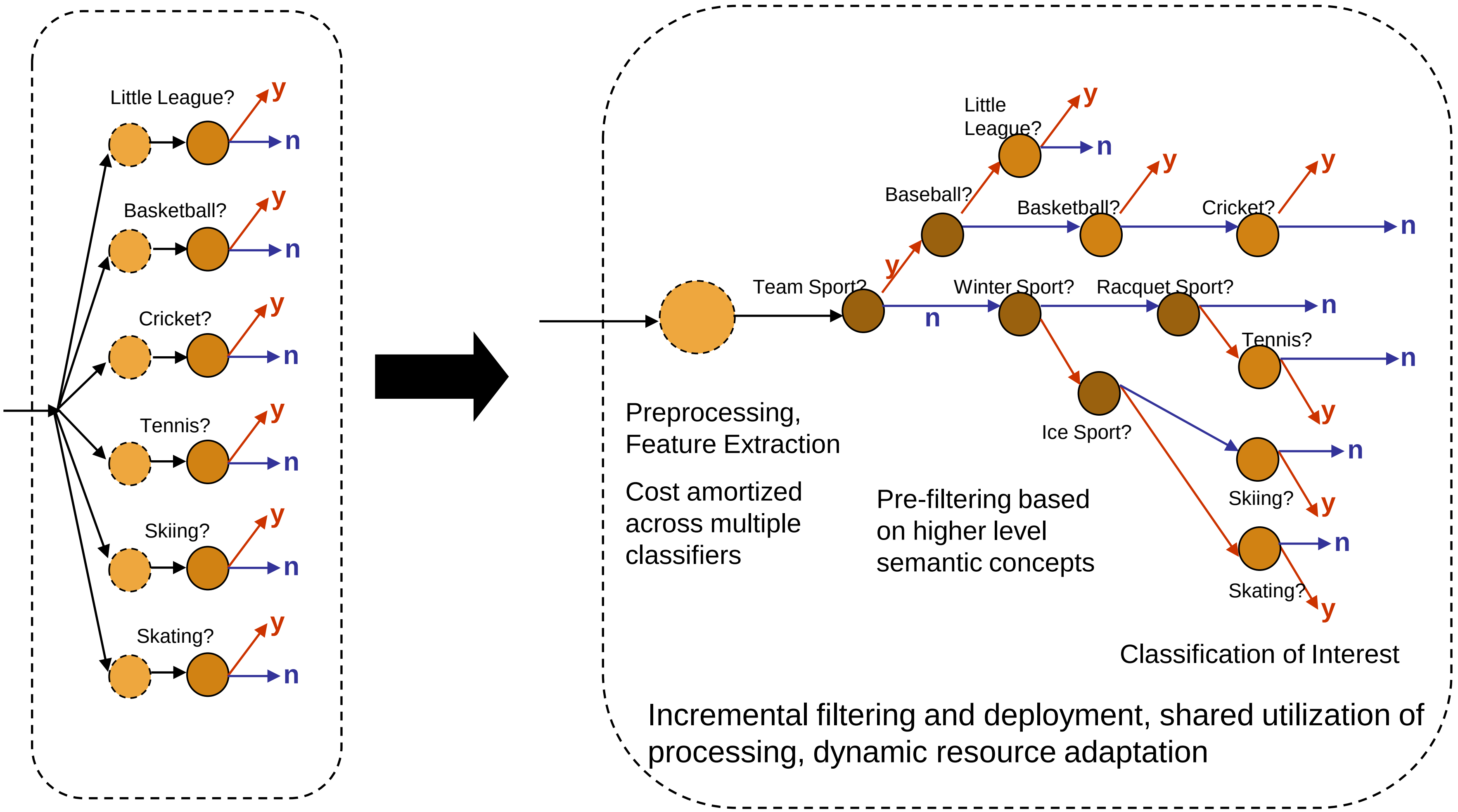
Guaranteed to converge to a local optimum

Run multiple times with different starting points for high probability of finding global optimum.

Application: Semantic Concept Detection

Streaming Sports Video → Key Frame Extraction →

Semantic Concept Detection Topology



Analytics/Classifiers

Classifiers use SVMs trained individually over 20000 sports key frames.

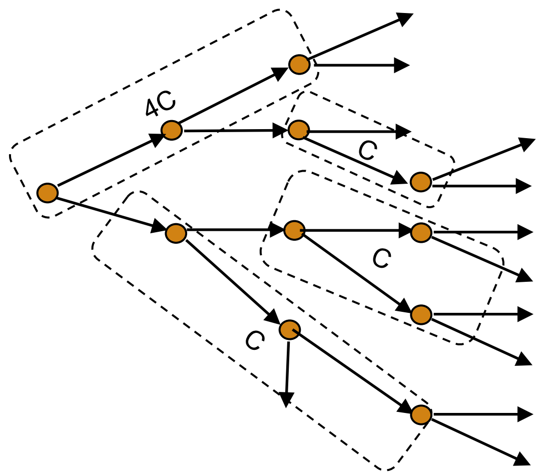
Experimental tree consists of 11 classifiers with a semantic hierarchy.

Experimental Results – set up

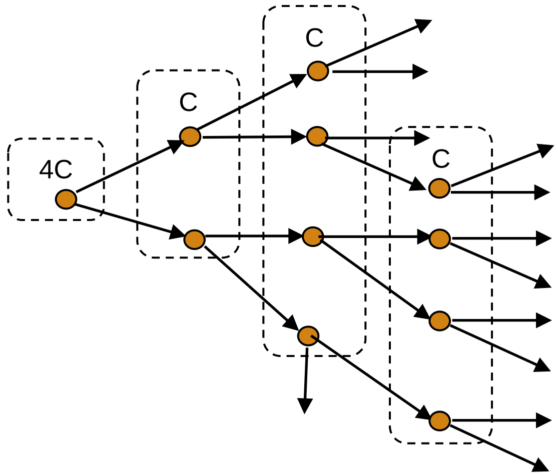
- Resource Adaptation Algorithms
  - A: Equal Error Rate configuration
  - B: Single operating point per classifier
  - C: Multiple operating points
  - D: Parallel Operation – Baseline

Table 1: Processing complexity per classifier.

Classifier	Complexity (cycles/image)	Apriori Probability
Team Sports (TS)	$3.884 \times C$	–
Baseball (BB)	$1.761 \times C$	–
Little League (LL)	$1.307 \times C$	0.0116
Basketball (BK)	$0.772 \times C$	0.0255
Cricket (CK)	$2.006 \times C$	0.0246
Winter Sports (WS)	$3.199 \times C$	–
Ice Sports (IS)	$2.223 \times C$	–
Skating (SA)	$2.403 \times C$	0.0415
Skiing (SI)	$2.608 \times C$	0.0655
Racquet Sports (RS)	$1.276 \times C$	–
Tennis (TN)	$1.720 \times C$	0.0095

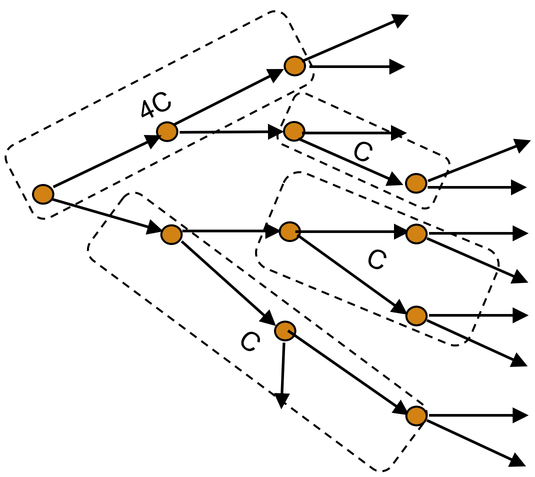


Placement 1-along each branch
(cross-talk minimizing)



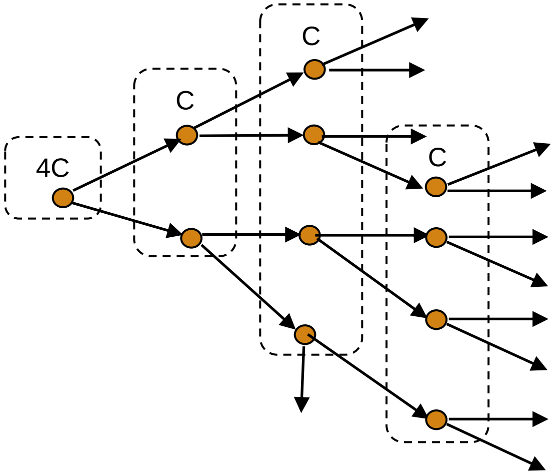
Placement 2-across different branches
(failure-resilient)

Experimental Results: Resource Constraints



Utility under Placement 1

	Equal Cost for FA and Miss	FA cost 10 times Miss cost	Miss cost 10 times FA cost
Algorithm A	-2.79	-2.19	-28.5
Algorithm B	1.74	27.93	-2.44
Algorithm C	2.53	34.54	7.99
Algorithm D	1.60	18.2	8.01



Utility under Placement 2

	Equal Cost for FA and Miss	FA cost 10 times Miss cost	Miss cost 10 times FA cost
Algorithm A	-3.14	-2.57	-31.9
Algorithm B	1.1	17.33	-3.96
Algorithm C	2.02	24.71	7.48
Algorithm D	0.79	8.15	6.86

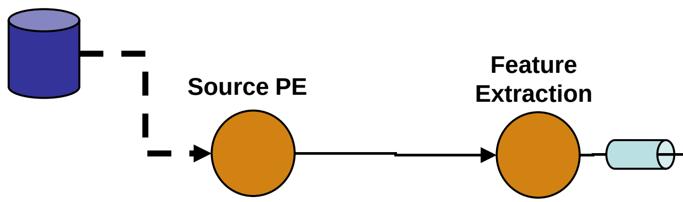
Centralized Algorithms: Limitations

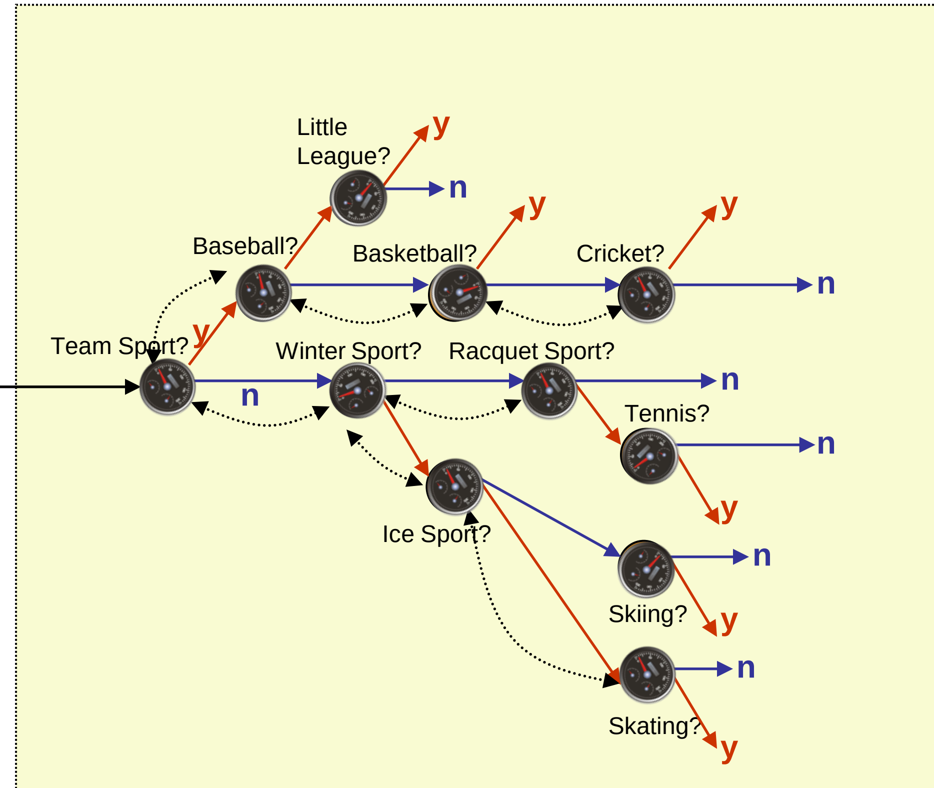
- *System and Information Bottlenecks* (single point of failure)
- *Topology specificity*.
- *Synchronization* requirements.
- *Time Delay* constraints are difficult to meet.
- Limited adaptability to *Dynamics*.
  - As an online process, stream mining optimization must involve algorithms that take into account the system’s dynamics, both in terms of the evolving stream characteristics and classifiers’ processing time variations
  - The dynamics are even more important in a multi-query context

Distributed Topology Construction

- **View individual classifiers as autonomous agents that optimize their own local utility**
  - Select appropriate operating points
  - Decide which classifier to forward current data tuple to
    - Convert topology construction into dynamic routing problem
    - Assume any classifier can forward data to any other classifier
  - Allows decentralized decision making
    - Requires minimal coordination between classifiers
    - Information on decisions directly encoded on data tuple
      - e.g. which classifiers have already processed the data sample, what operating points they used etc.
  - Allows dynamic optimization

Distributed Classifier Configuration





Individual classifiers reconfigure themselves

Classifiers exchange information

Goal: Organize set of classifiers into chain topology

- Minimize misclassification penalty
- Minimize processing delay

Problem Formulation - Distributed

For a permutation σ of N classifiers in chain, i.e. $\sigma \in Perm(N)$

For operating point for each classifier $\left(p_i^D, p_i^F\right) = \left(f(x_i), x_i\right)$

End to end misclassification penalty $c_{err}^{\sigma,\mathbf{x}} = c^M \left(\boxed{\Phi t_0 - g_N^{\sigma,\mathbf{x}}}\right) + c^F \left(\boxed{t_N^{\sigma,\mathbf{x}} - g_N^{\sigma,\mathbf{x}}}\right)$

Total missed detection

Total false alarms

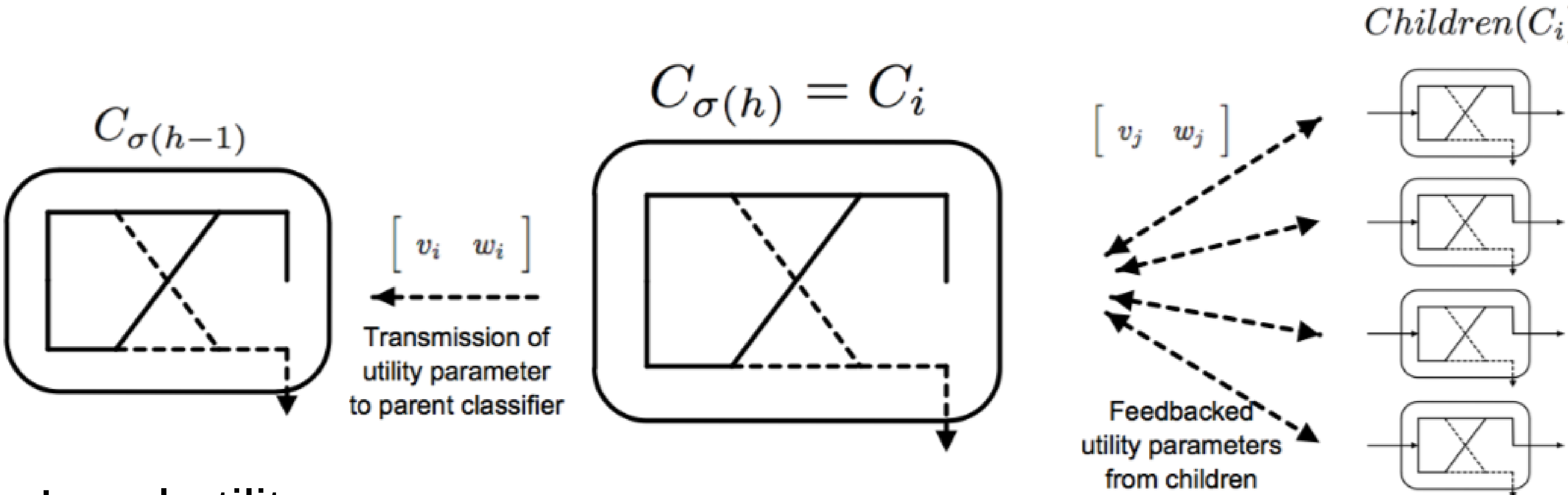
End to end processing delay $c_{delay}^{\sigma,\mathbf{x}} = \sum_{k=1}^{N} \boxed{\alpha_{\sigma(k)}} t_{k-1}^{\sigma,\mathbf{x}}$

Time per unit rate of data for classifier $\sigma(k)$

Select operating point per classifier and organize them to minimize **misclassification penalty and delay**

$$\min_{\sigma,\mathbf{x}} C(\sigma,\mathbf{x}) = c_{err}^{\sigma,\mathbf{x}} + \lambda c_{delay}^{\sigma,\mathbf{x}}$$

Distributed Order Selection: Exhaustive

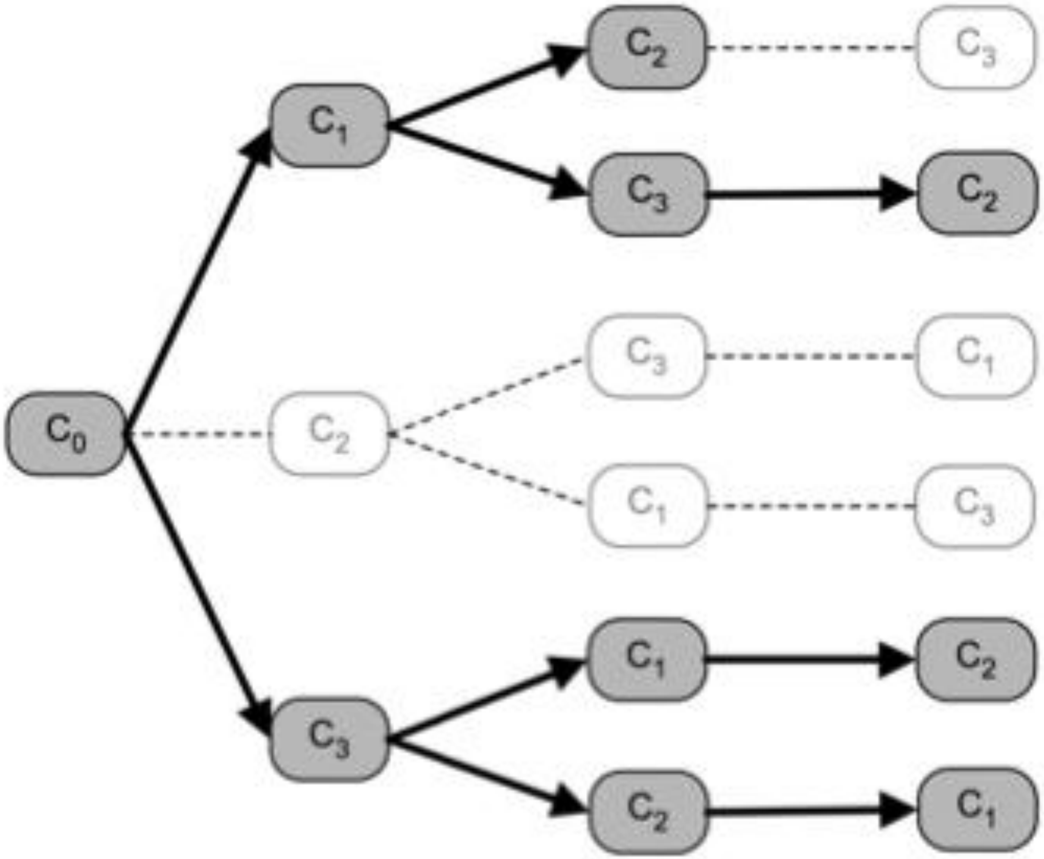


- Local utility
  - Computed by backward propagation from children to parent classifier
  - For any given order
    - Compute optimal operating point (utility coefficients) for last classifier
    - Propagate local utility coefficients backwards to previous classifier
    - Previous classifier can compute optimal utility coefficients
    - Continue back-propagation to first classifier to determine end to end utility
- Exhaustive Search based Ordering
  - All possible orders ($N!$) considered $\Rightarrow$ Computationally infeasible

Multi-agent Learning for Topology Construction

Which children to probe?

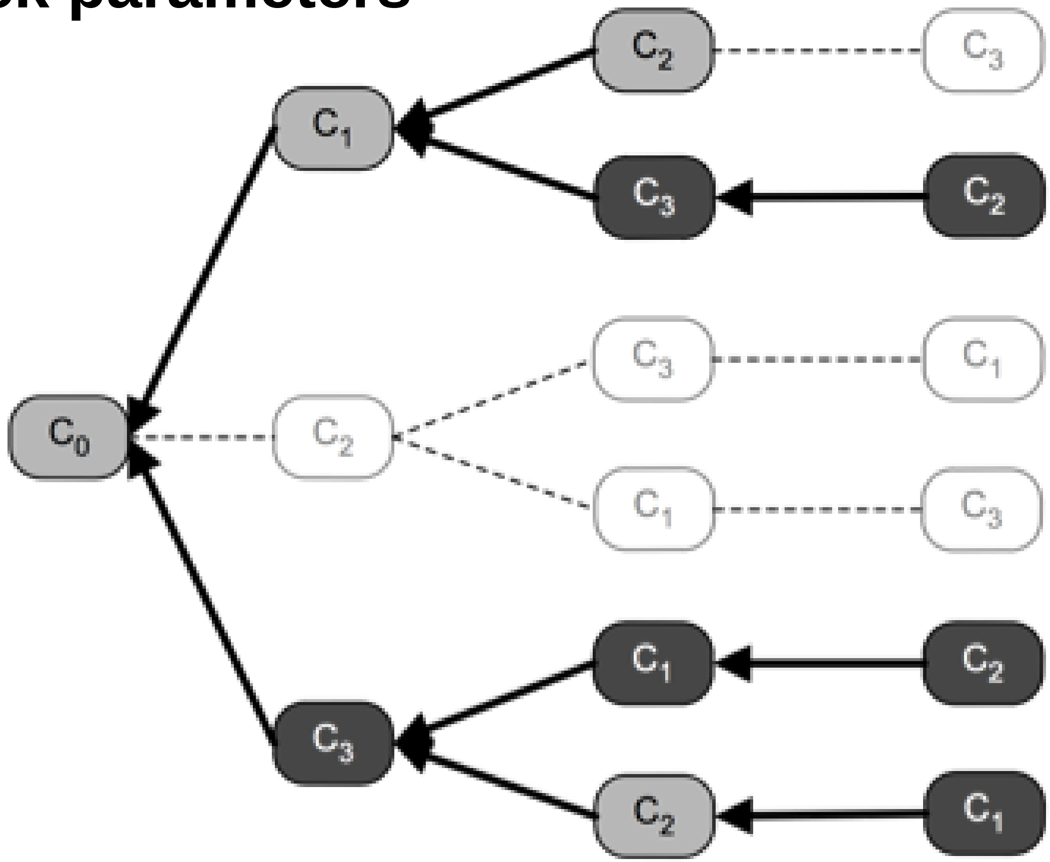
Each classifier requests the parameters of a subset of its children classifiers (online learning)



Multi-agent Learning for Topology Construction

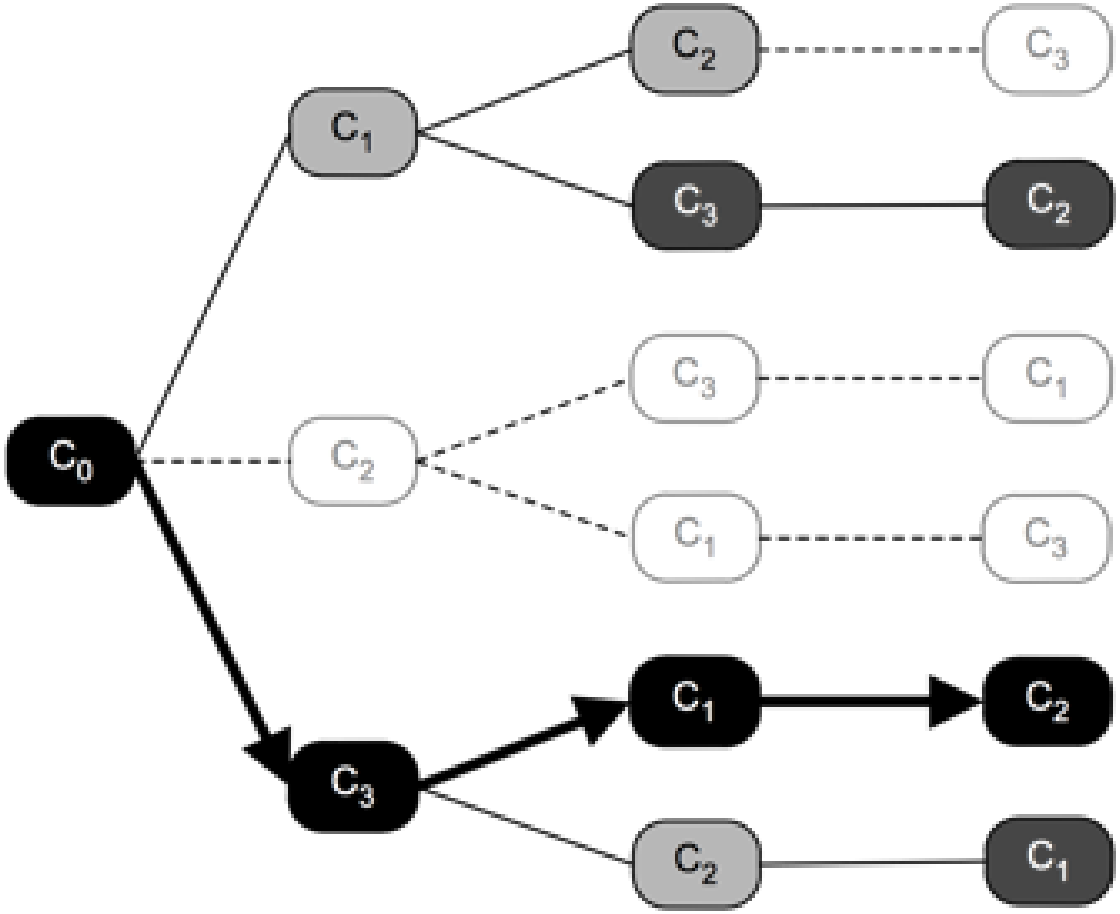
Who do you trust?

Computation of local utilities based on children utility feedback parameters



Multi-agent Learning for Topology Construction

Process the stream



Learning based Distributed Approach

- Determine random search probability to improve convergence speed of learning algorithm
- Given time constraint and time to search one order
  - Determine fixed optimal screening probability to ensure number of orders searched is within time constraint
- Exploration versus Exploitation
  - Modify the random probability to tradeoff exploitation versus exploration
  - Probability may be weighted after each iteration
  - Weight chosen proportional to utility achieved by prior probes
    - Different weighting schemes lead to different tradeoffs

- **Safe Experimentation**
- **Parametric Partial Search Ordering (new!)**

Selection criteria for distributed algorithms

Ordering Algorithm	System compliance	Utility achieved	Message exchange	Speed of convergence	Adaptability	Control
No algorithm	$\varnothing$	Suboptimal	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
Adaptive-Greedy	Low	Bounded	Heavy	Very rapid	Little	$\varnothing$
Safe Experimentation	High	Optimal	$\varnothing$	Medium	$\varnothing$	$\varnothing$
Partial Search (proposed)	Complete	Optimal	Light	Rapid	Total	Yes

Summary

- **New Knowledge Extraction Paradigm**
- Stream mining requires building topologies of classifiers (deployed on distributed systems) for multi-concept detection in data streams
- **Novel Framework for Stream Mining**
  – Decentralized decision framework
  – Joint topology construction and local classifier configuration
  – Real-time and distributed algorithms
  – Decentralized Solutions based on Interactive Multi-Agent Learning

Many Research Opportunities

- Design multimedia classifier with various operation points (multimedia processing, filtering, correlations, patter recognition etc.)
- Jointly training classifiers
- Building M-ary classifier trees
- Tree construction, design and training
- Joint optimization of performance with node placement
- Strategic classifiers (game theory, pricing etc.)
- Dynamic stream mining systems
- Decentralized stream mining systems – granularity of adaptation, message exchange, monitoring etc.
- Building new cyber-discovery multimedia applications – real-time knowledge extraction, decision making etc.
- Combine human and computer computation (crowdsourcing)

Multimedia Processing and Mining – Open research issues

- Sensing and Processing of Data from Diverse Media Sources
  - Task Driven, Adaptive Sensing
  - Robust and Error Resilient Data Gathering
  - Distributed compression, data reduction, processing
- Managing confidence, uncertainty and noise
  - Missing samples, delayed samples, non-time aligned
  - Algorithms and system services to support analytics/application
- Resource Constrained Data Analytics
  - Resource adaptive filtering, tracking, feature extraction, compression
  - Complexity scalable mining and classification algorithms
  - Distributed Analytics

Multimedia Processing and Mining – Open research issues

- Application-specific challenges
  - Heterogeneity of data types
  - Multiple time horizons – retrospective, predictive, real-time queries
  - Mixed model learning and use (dynamic update)
- Cross-Layer Design, Networking, Optimization and Control
  - Across analytics' applications, algorithms and systems
  - Interactions with resource management
    - Composing analytics into applications
    - Deploying applications on processing nodes
    - Managing resources dynamically
  - Use of application relevant utility-cost-complexity metrics for resource-performance tradeoffs
    - Relevance through Utility based metrics
    - Relevance through Information Bottleneck based metrics

Acknowledgements

- Collaborator:

  Dr. Deepak Turaga (IBM Research)
- My Graduate Students:

  Raphael Ducasse

  Dr. Brian Foo

  Dr. Fangwen Fu

References

- http://medianetlab.ee.ucla.edu/publications.html

1. H. Park. D. S. Turaga, O. Verscheure, and M. van der Schaar, Foresighted Tree Configuration Games in Resource Constrained Distributed Stream Mining Sensors," *Ad Hoc Networks (Invited Paper to Special Issue: Multimedia Ad Hoc and Sensor Network)*, 2010.

2. B. Foo and M. van der Schaar, "A Distributed Approach for Optimizing Cascaded Classifier Topologies in Real-Time Stream Mining Systems" *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 3035-3048, Nov. 2010.

3. R. Ducasse, D. Turaga, and M. van der Schaar, "Adaptive Topologic Optimization for Large-Scale Stream Mining" *IEEE Journal of Selected Topics in Signal Process. (JSTSP)*, vol. 4, no. 3, pp. 620-636, June 2010.

4. B. Foo, D. Turaga, O. Verscheure, L. Amini, and M. van der Schaar, "Configuring Trees of Classifiers in Distributed Multimedia Stream Mining Systems," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 21, no. 3, pp. 245-258, Mar. 2011.

5. B. Foo and M. van der Schaar, "A Rules-based Approach for Configuring Chains of Classifiers in Real-Time Stream Mining Systems," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, Article ID 975640, 17 pages, 2009. doi:10.1155/2009/975640.

6. B. Foo, D. Turaga, O. Verscheure, M. van der Schaar, and L. Amini, "Resource Constrained Stream Mining with Classifier Tree Topologies," *IEEE Signal Process. Lett.*, vol. 15, pp. 761-764, 2008.

7. F. Fu, D. Turaga, O. Verscheure, M. van der Schaar, and L. Amini, "Configuring Competing Classifier Chains in Distributed Stream Mining Systems" *IEEE Journal of Selected Topics in Signal Process. (JSTSP)* , vol. 1, no. 4, pp. 548-563, Dec. 2007.

http://medianetlab.ee.ucla.edu/publications.html