University of Colombo School of Computing, April 23rd 2009

Machine Learning in Support of Biomedical Security

Selected Examples

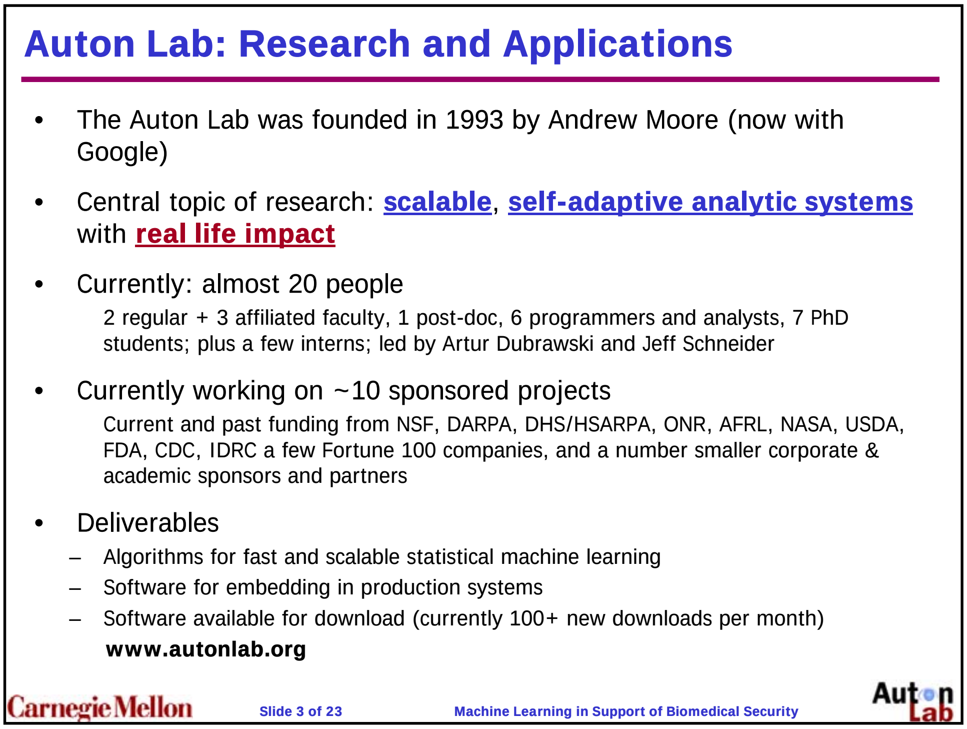
Artur Dubrawski

Director, The Auton Lab
Carnegie Mellon University
www.autonlab.org
awd@cs.cmu.edu

Agenda

1. Quick introduction of the CMU Auton Lab
2. Learning detectors of events manifested in multiple streams of data
3. Maintaining scalability of machine learning systems using cached sufficient statistics (example: T-Cube)
4. Demo: Real-Time Biosurveillance Project – pilot deployment of event detection system in Sri Lanka

Note: This agenda is meant to be flexible and can be adjusted on the fly…



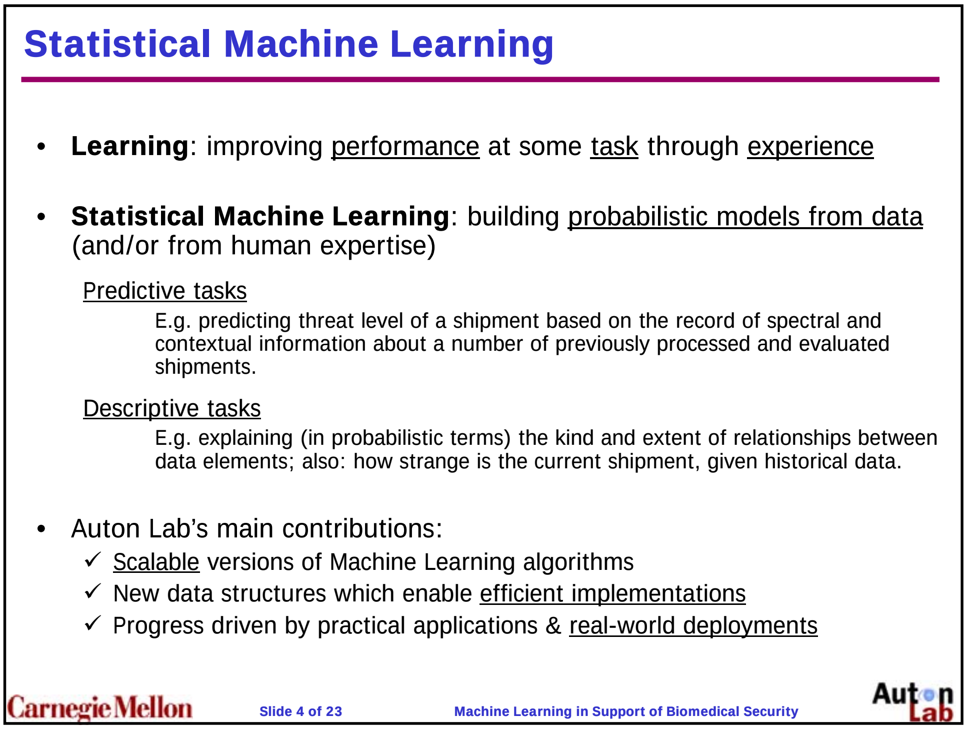
Auton Lab: Research and Applications

- The Auton Lab was founded in 1993 by Andrew Moore (now with Google)
- Central topic of research: **scalable**, **self-adaptive analytic systems** with **real life impact**
- Currently: almost 20 people

  2 regular + 3 affiliated faculty, 1 post-doc, 6 programmers and analysts, 7 PhD students; plus a few interns; led by Artur Dubrawski and Jeff Schneider
- Currently working on ~10 sponsored projects

  Current and past funding from NSF, DARPA, DHS/HSARPA, ONR, AFRL, NASA, USDA, FDA, CDC, IDRC a few Fortune 100 companies, and a number smaller corporate & academic sponsors and partners
- Deliverables
  - Algorithms for fast and scalable statistical machine learning
  - Software for embedding in production systems
  - Software available for download (currently 100+ new downloads per month)

  www.autonlab.org



Statistical Machine Learning

- **Learning**: improving performance at some task through experience
- **Statistical Machine Learning**: building probabilistic models from data (and/or from human expertise)

  Predictive tasks

  E.g. predicting threat level of a shipment based on the record of spectral and contextual information about a number of previously processed and evaluated shipments.

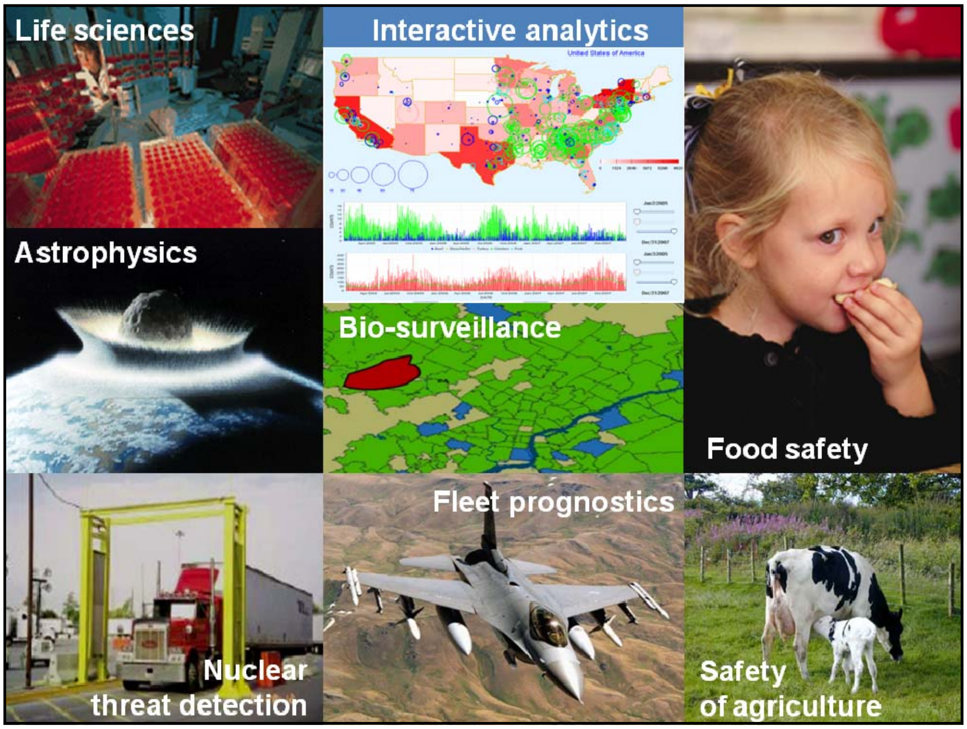
  Descriptive tasks

  E.g. explaining (in probabilistic terms) the kind and extent of relationships between data elements; also: how strange is the current shipment, given historical data.
- Auton Lab's main contributions:

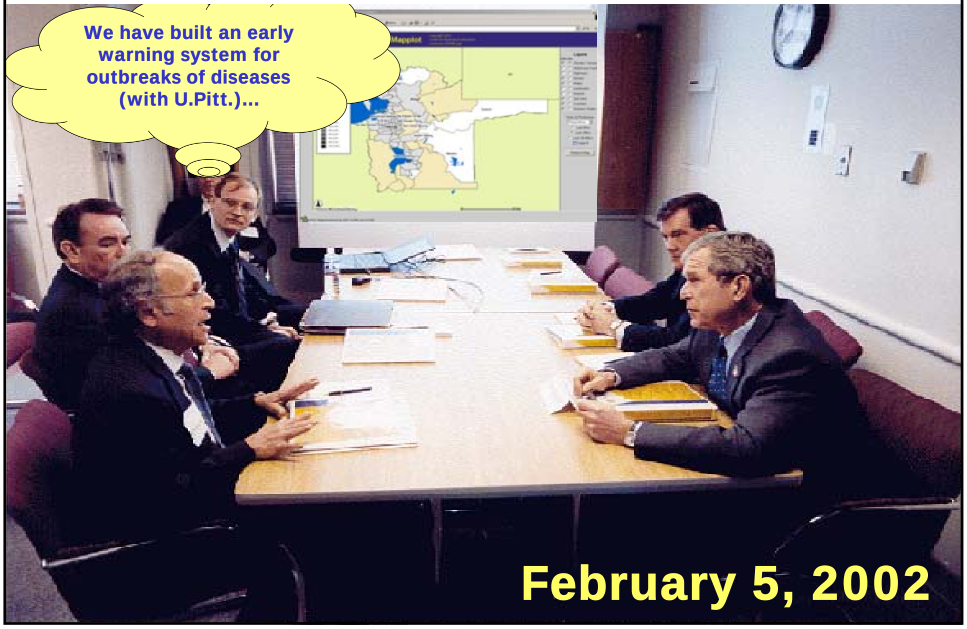
  ✓ Scalable versions of Machine Learning algorithms

  ✓ New data structures which enable efficient implementations

  ✓ Progress driven by practical applications & real-world deployments



Real-time Monitoring of Emergency Department Chief Complaints during 2002 Winter Olympics



February 5, 2002

Why Do We Care?

Highly motivational example: Adverse bio-events

- Terrorist acts (Anthrax, Smallpox, …)
  - 100 kg of anthrax released in DC may kill 1—3 million people (WHO)
  - “For the life of me, I cannot understand why terrorists have not attacked our food supply because it is so easy to do” (Resignation speech by departing U.S. Health Secretary Tommy Thompson, December 3, 2004)
- Naturally occurring events, including emerging threats (SARS, Avian Influenza, West Nile Virus, Mad Cow Disease, Foot-and-Mouth Disease, E.coli, Salmonella, …)
  - Lower-bound estimate of death toll of Avian Influenza pandemic: 2—7 million lives
- Unintentionally introduced events (such as accidental contamination of food at processing plant)
  - Each year in the US: 76 million cases of food borne diseases (5,000+ of them fatal)

Why Do We Care?

- A 2-day gain in detection time during an incident of inhalational Anthrax release could reduce fatalities by a factor of six (DARPA)
- Improvements of even an hour over current detection capabilities could reduce economic impact of a bioterrorist anthrax attack by hundreds of millions of dollars (Wagner et al. 2004)
- US agricultural industry: $1T of economic activity, $60B in food exports, root cause identification is apparently very difficult: 82% of food-borne disease cases are of unknown origin.

How Machine Learning community can help:

- Exploit available early signals
- Build algorithms and supporting data structures for rapid detection

Auton Lab’s algorithms and data structures for rapid detection of emerging patterns

- What’s Strange About Recent Events (WSARE)

  Monitoring Emergency Rooms chief complaints for unusually high counts of patients from specific sub-populations who report with similar symptoms

- Fast Spatial Scan
  - Expectation-based
  - Multivariate Bayesian

  Monitoring volumes of non-prescription drug sales and/or ER data for spatio-temporal over-densities

- Multi-Stream Temporal Monitor

  Combining evidence from multiple streams of time series for detection of anomalies and specific patterns of interest

- Tip Monitor

  Monitoring food consumer complaints for small sets of similar reports which may be attributed to a common underlying cause

- T-Cube

  In-memory data structure enabling huge speedups in time series extraction and aggregation

What’s Strange About Recent Events (Wong’s Ph.D. thesis, 2003)

Representative Surveillance Data

Date	Time	Hospital	ICD9	Prodrome	Gender	Age	Home Location	Many more...
6/1/03	9:12	1	781	Fever	M	20s	NE	...
6/1/03	9:45	1	787	Diarrhea	F	40s	SE	...
:	:	:	:	:	:	:	:	:

Standard Approach

Select in advance which subpopulations to monitor (e.g., each county, zip)

Do not pay close attention to effect of multiple testing

WSARE Approach

Monitor hundreds of thousands of subpopulations

Pay close attention to effect of multiple testing

Evaluation

Detailed comparison on 2,000 simulated scenarios and Western PA ER Data

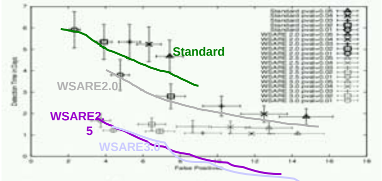
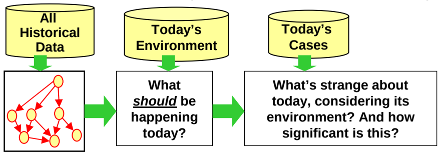


Figure 5: Scatterplot of Detection Time versus False Positives with Error Bars for Simulated Data

Method

- Search over 100,000s of subpopulations
- For each subpopulation, use as good of a model as can be created to predict expected counts
- The model will have a form of a multi-component rule, e.g.*”There is a surprisingly large number of children with respiratory problems today”*
- Compute p value, taking into account multiple testing


All Historical Data

Today’s Environment

Today’s Cases

What *should* be happening today?

What’s strange about today, considering its environment? And how significant is this?

Significance

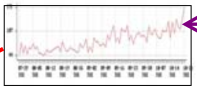
- WSARE is designed to detect small clusters of illness in healthcare workers, age groups, workplaces...
- Israeli Center for Disease Control evaluated WSARE retrospectively using an unusual outbreak of influenza type B that occurred in an elementary school in central Israel. It detected the outbreak on the second day from its onset.
- Retrospective analysis of Walkerton case: alerted one day ahead of the issuance of boil water advisory (at the expected rate of 2 false positives per year).

Fast Spatial Scan

(Neill's Ph.D. thesis, 2006)

Method (Expectation-Based Variant of FSS)

OTC drug sales



Time series of counts for each zip code (at least 3 months of historical data).

- This increase could be due to an outbreak, or due to chance.
- Which regions of increase are significant?
- Search all rectangular regions on the grid.

Multivariate Bayesian variant of this method analyzes multiple streams of data to maximize detection power while enabling disambiguation among possible causes of outbreaks

- Solution: We learn the expected count for each area from historical data.
- Then we find regions where the recent **counts are significantly higher than expected**, accounting for anticipated spatial and temporal variations

Evaluation

- Current version typically turns multi-day analysis into <**20 minutes** for daily counts from over 20,000 drugstores nationwide
- Searches *all* rectangular regions.
- Results are *exact* (not approximations).

Significance

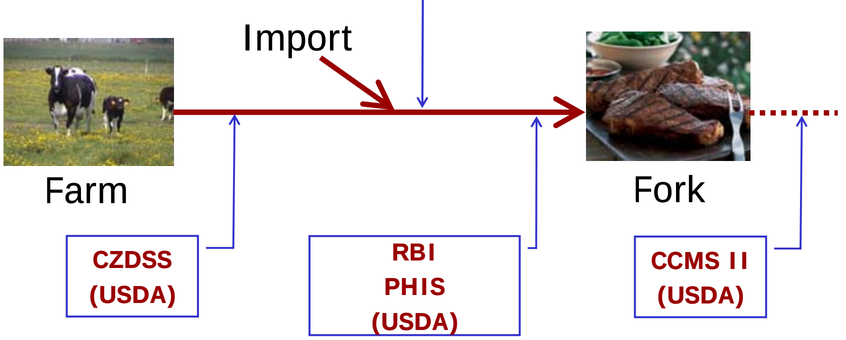
- Traditional spatial scan is very expensive, especially with randomization tests of significance
- A few hours difference may actually matter.
- Retrospective analysis of Walkerton case: alerted two days ahead of the issuance of boil water advisory.

Algorithm	Search space	# of regions	Search time	Time / region	Likelihood ratio
SaTScan	Circles centered at datapoints	150 billion	16 hours	400 ns	413.56
exhaustive	Axis-aligned rectangles	1.1 trillion	45 days	3600 ns	429.85
FSS (2006)	Axis-aligned rectangles	1.1 trillion	81 minutes	4.4 ns	429.85

ER dataset (600,000 records), 1000 replicas; for SaTScan: M=17,000 distinct spatial locations; for exhaustive/fast: 256 x 256 grid.

Safety of Food Supply and Agriculture

Common denominator of these multiple projects: To quickly detect emerging patterns and rank them according to the expected risk, so that investigative resources can be allocated efficiently



Slaughterhouses
(condemnations & results of lab tests for pathogens, residues and microbes causing zoonotic diseases)

Multiple Streams
(condemnations, lab tests, inspection results, recalls, …)

Consumer Complaints

The Need for Multivariate / Multi-stream Detectors

Different streams of data or different dimensions of the same data stream often carry corroborating evidence about the events of interest

It may be beneficial to analyze them jointly in order to:

- Increase accuracy of detection
- Decrease latency of detection
- Improve specificity of detection

Examples:

Syndromic bio-surveillance:

- Multi-stream: Over-The-Counter medicine sales vs. Records of patients reporting to hospitals vs. School absenteeism vs. Lab test requests vs. Many other measurable factors
- Multi-variate: Different categories of OTC drugs; Different symptoms of ER patients

Attribution and predictive analytics in public health:

- Multi-stream: Microbial isolates from humans vs. Results of microbial testing of food samples taken at food factories

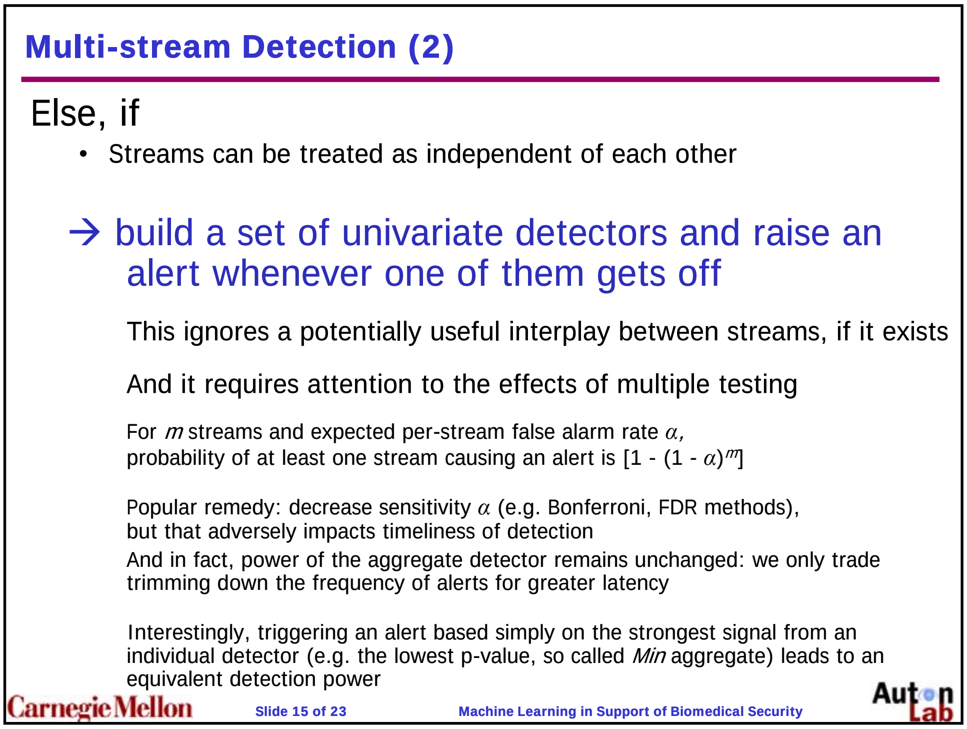
Multi-stream Detection (1)

If

- A strong model of informative relationships between the multiple variables (multiple data streams) is available, and
- Streams have coherent statistical properties

→ build joint multivariate models

This approach is a very appealing standard, but it requires understanding of the structure of relationships and a sufficient amount of evidence in data for reliable estimation of the joint model



Multi-stream Detection (2)

Else, if

- Streams can be treated as independent of each other

→ build a set of univariate detectors and raise an alert whenever one of them gets off

This ignores a potentially useful interplay between streams, if it exists

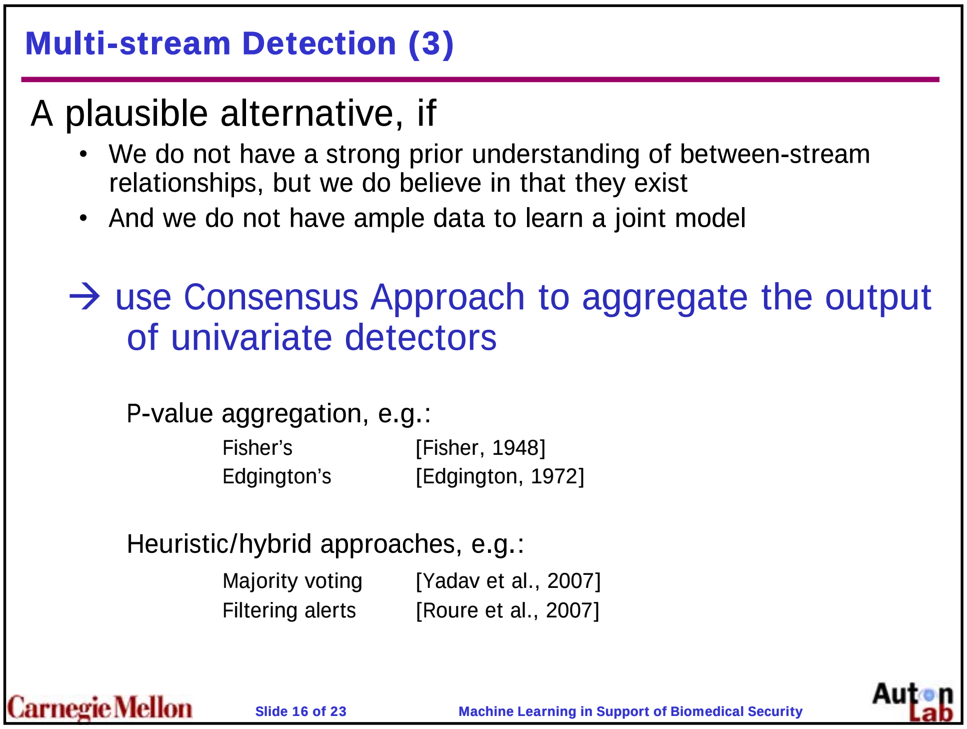
And it requires attention to the effects of multiple testing

For m streams and expected per-stream false alarm rate α,
probability of at least one stream causing an alert is $[1 - (1 - \alpha)^m]$

Popular remedy: decrease sensitivity α (e.g. Bonferroni, FDR methods), but that adversely impacts timeliness of detection

And in fact, power of the aggregate detector remains unchanged: we only trade trimming down the frequency of alerts for greater latency

Interestingly, triggering an alert based simply on the strongest signal from an individual detector (e.g. the lowest p-value, so called *Min* aggregate) leads to an equivalent detection power



Multi-stream Detection (3)

A plausible alternative, if

- We do not have a strong prior understanding of between-stream relationships, but we do believe in that they exist
- And we do not have ample data to learn a joint model

→ use Consensus Approach to aggregate the output of univariate detectors

P-value aggregation, e.g.:

Fisher's	[Fisher, 1948]
Edgington's	[Edgington, 1972]

Heuristic/hybrid approaches, e.g.:

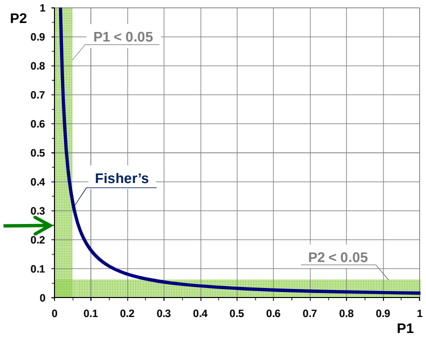
Majority voting	[Yadav et al., 2007]
Filtering alerts	[Roure et al., 2007]

Fisher's Method of P-value Aggregation

Under null hypothesis, *p*-values are distributed uniformly, and Fisher's statistic for *m* independent *p*-values ($2\Sigma\ln p_i$) has a chi-square distribution with 2*m* d.o.f. In fact, there exists a closed-form solution for the combined *p*-value [Jost]:

$$p_F = k\sum_{i=0}^{m-1}\frac{(-\ln k)^i}{i!}\text{ , where } \; k=\prod_{i=0}^{m-1} p_i \; .$$

Example:
2 streams with independent univariate detectors



Null hypothesis rejection region for *Min* criterion with α=0.05

Fisher's aggregation leads to rejecting null hypothesis also if both component p-values are just slightly greater than critical

It is more conservative than *Min* approach when either of the components is much greater than critical

Example Application:
Detection of Events in Multi-stream Food / Agriculture Safety Data

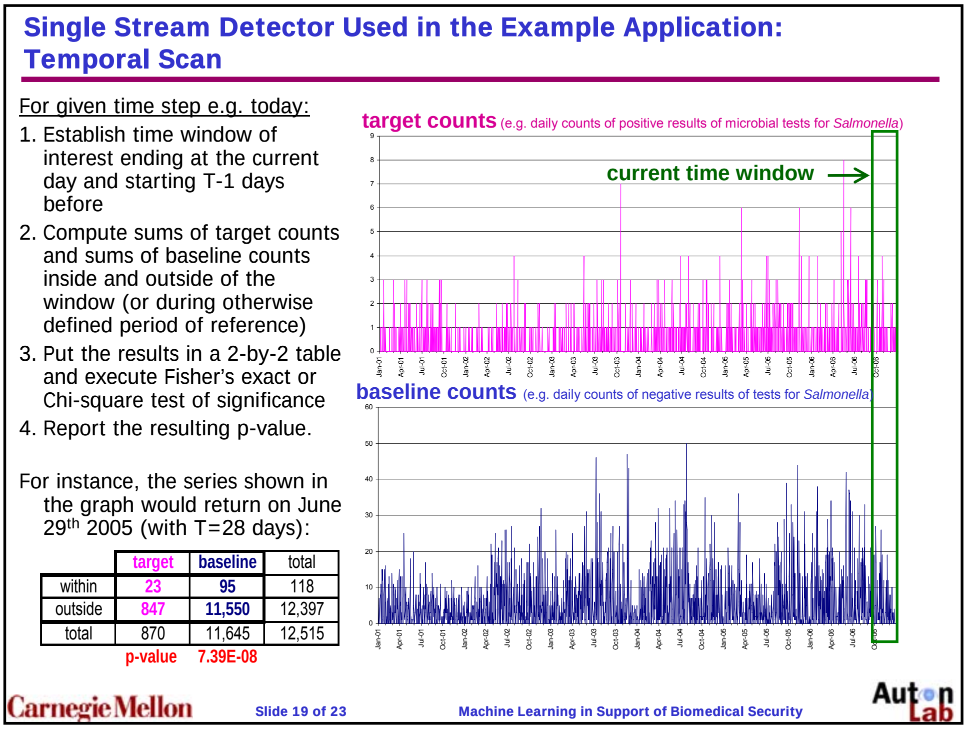
The U.S. Department of Agriculture (USDA) collects data from different sources, including:

Stream A: Daily counts of healthy and condemned cattle arriving at slaughterhouses

Stream B: Daily counts of positive and negative results of microbial tests of meat products

Stream C: Daily counts of passed and failed sanitary inspections of slaughterhouses

USDA analysts are interested in monitoring these streams of data for unexpected, temporally coinciding increases in positive counts in all data or in specific, e.g. geographically co-located, subsets of it



Single Stream Detector Used in the Example Application: Temporal Scan

For given time step e.g. today:

1. Establish time window of interest ending at the current day and starting T-1 days before
2. Compute sums of target counts and sums of baseline counts inside and outside of the window (or during otherwise defined period of reference)
3. Put the results in a 2-by-2 table and execute Fisher's exact or Chi-square test of significance
4. Report the resulting p-value.

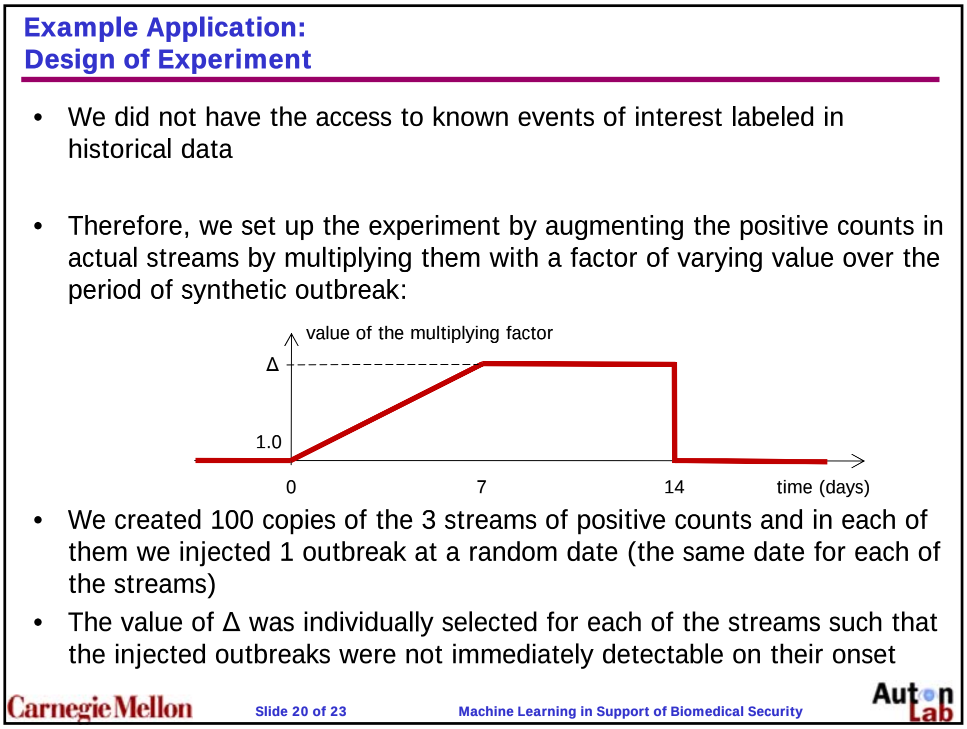
For instance, the series shown in the graph would return on June 29th 2005 (with T=28 days):

	target	baseline	total
within	23	95	118
outside	847	11,550	12,397
total	870	11,645	12,515
	p-value	7.39E-08	

target counts (e.g. daily counts of positive results of microbial tests for *Salmonella*)

current time window

baseline counts (e.g. daily counts of negative results of tests for *Salmonella*)



Example Application: Design of Experiment

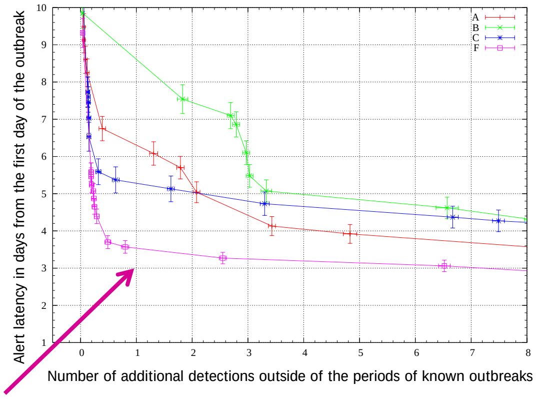
- We did not have the access to known events of interest labeled in historical data
- Therefore, we set up the experiment by augmenting the positive counts in actual streams by multiplying them with a factor of varying value over the period of synthetic outbreak:

- We created 100 copies of the 3 streams of positive counts and in each of them we injected 1 outbreak at a random date (the same date for each of the streams)
- The value of Δ was individually selected for each of the streams such that the injected outbreaks were not immediately detectable on their onset

Applying Fisher's Aggregation

Power of detectors can be conveniently evaluated using Activity Monitoring Operating Characteristic (AMOC) graphs:

This figure presents characteristics obtained for uni-variate detectors applied to the individual streams A, B and C, as well as AMOC of their Fisher's aggregate (labeled F)



Each point in the graph corresponds to one setting of sensitivity of a detector (i.e. the significance threshold α)

It corresponds to the mean latency and the mean number of additional detections computed from 100 independent tests

The error bars depict standard errors in these means

The aggregate detector is substantially more powerful than any of the individual components

Non-specific vs. Specific Detectors & Learning Them

- That is very nice

  But, Fisher's method produces a non-specific detector

  It treats all the components equally in targeting departures from the joint NULL distribution

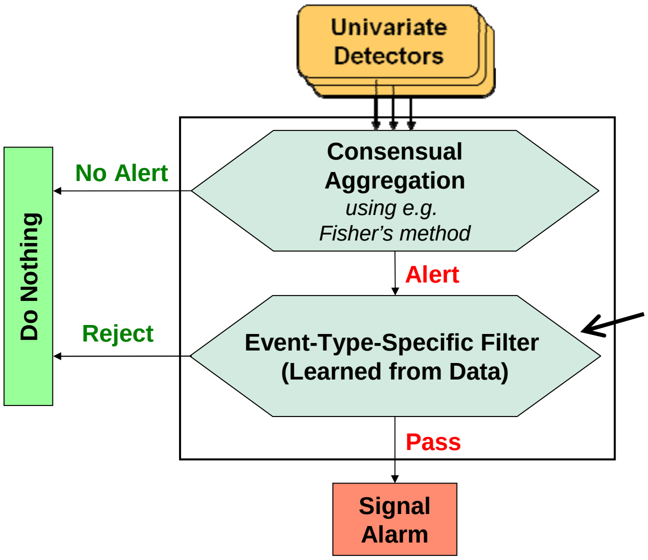
- That is quite useful if we are after general anomaly detection and therefore we do not care about specific events with particular characteristics

- If we do, it may be possible to tweak Fisher's method to produce a more powerful, specific detector

- Manual design of specific detectors can be subjective and tedious though

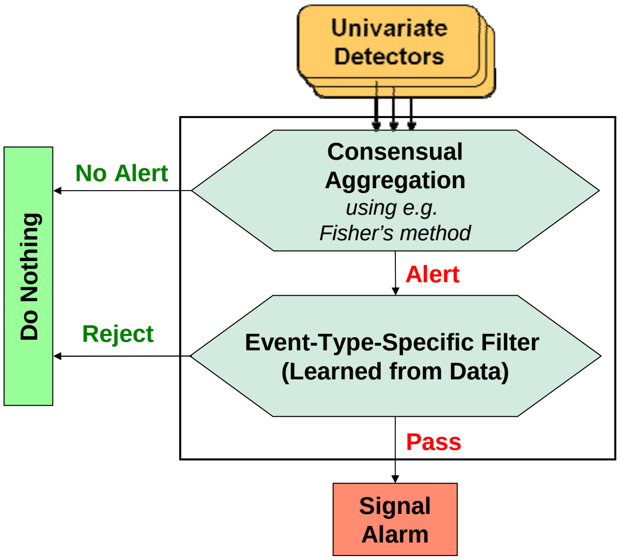
- If we have data with labeled events of interest and labeled periods without them, we could use it to automatically train classifiers of events

How Would That Work?


Univariate Detectors
Consensual Aggregation *using e.g. Fisher's method*
No Alert
Alert
Do Nothing
Reject
Event-Type-Specific Filter (Learned from Data)
Pass
Signal Alarm

A learned from data filter of irrelevant alerts

How Would That Work?


Univariate Detectors
Consensual Aggregation *using e.g. Fisher's method*
No Alert
Alert
Do Nothing
Reject
Event-Type-Specific Filter (Learned from Data)
Pass
Signal Alarm

We can learn from data filters of irrelevant alerts:

→ Using a set of positive (outbreaks) and negative examples (normal data)

→ Using only negative (no-event) examples

→ Or using negative examples with whatever small number of positive examples we may have

→ We can also train filters that tell apart different types of events

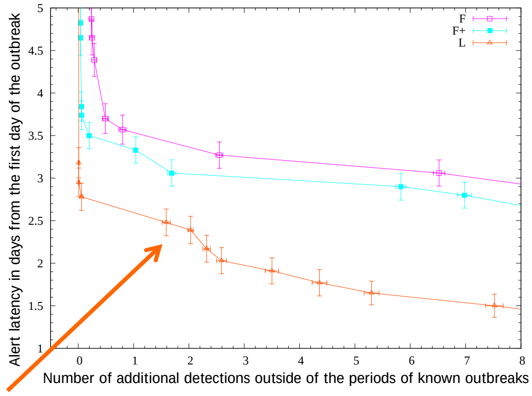
Experiment:
Learning Classifiers as Multivariate Event Detectors

- We independently created 100 synthetic data sets with injected outbreaks for training, and another 100 for testing
- Output space: Data belonging to the period of outbreak were labeled as positive and data for the 14 days after the outbreak as negative examples
- Input feature space: a Cartesian product of the following sets:
  - 4 *p*-values: 3 computed using temporal scan independently for each component stream; and 1 Fisher's aggregate
  - 3 widths of temporal scan window: 1, 2, and 3 days
  - 3 days of data: the day of analysis and the two preceding days

  That makes 36 numeric features per data point (one day)
- We used a Random Forest classifier [Breiman 2001]
  - Each forest made of 40 decision trees, each tree trained on a different bootstrap sample of the training data
  - The percentage of classifiers predicting positive used as the output value
  - 10 forests, each built with a different random seed of the bootstrap sample
  - Prediction deemed positive if the average prediction of 10 forests exceeds a pre-set threshold

Learning Specific Detectors from Labeled Data

This graph compares AMOC curves for Fisher's aggregate (**F**), Fisher's based hand-crafted detector (**F+**) and the classifier-based detector (**L**)



Specific multivariate detector learned from data outperforms the manually designed one as well as the non-specific detector based on Fisher's aggregate

Its characteristic was obtained by varying the value of the prediction threshold

Challenges of Learning Detectors from Data

- The news from the previous slide is VERY GOOD!
- BUT, how often do we get to see hundreds of available and documented adverse health-related events of the same kind recorded in data?
- In many practical scenarios experts cannot easily identify many examples of outbreaks…

  (that is NOT very good news for our learning filters)
- However, often experts can easily provide examples of days when no special events have occurred

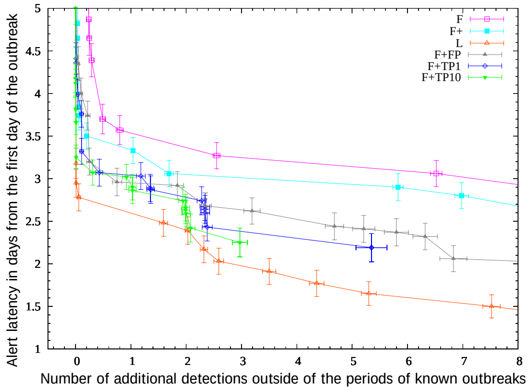
  And, we can learn useful detectors using primarily such negative examples ☺

Learning Detectors Using Limited Number of Labeled Outbreaks

This graph compares performance of detectors learned from only negative examples (F+FP), negative and just one labeled positive case (F+TP1) and negatives plus 10 positives (F+TP10), against that of previously discussed models (F, F+, L)

Filter trained on negative-only examples used Gaussian Kernel Density Estimator

These examples were taken from data labeled as negative by the Fisher's aggregation model



Filters trained on negative and few positive examples use separate Gaussian Kernel Density models for the two classes of examples, linked via Bayes' rule

Detectors trained to filter out false-positives do better than either plain or specific Fisher's models, but they are outperformed by the detector trained using a large sample of positive examples

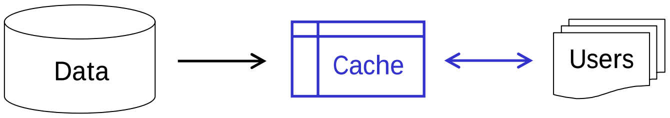
The more the positive examples at hand, the better the attained performance

Multi-stream Event Detection: Lessons Learned

1. Aggregation of complementary evidence from multiple streams of data allows for increased detection power
2. Event-type-specific detectors can further reduce the number of false detections, which allows for increased sensitivity of detection
3. Hand-crafting them may be tedious and subjective
4. Learning specific detectors from data is a good alternative
5. They can be automatically learned even if the amount of training data is very limited:
    - Only from negative examples
    - From negative and just a few available positive examples
    - The more labeled examples the better the attainable performance of trained detectors
6. Learning makes an appealing method of design; It has a potential to reduce costs of development and maintenance of future detection systems

Scalability: Data Access via Cached Sufficient Statistics

- Old School:
    - Static database reports processed in a batch mode
- Modern:
    - Data Warehousing, interactive access to data from the analyst's desktop
    - Business Intelligence portals with pre-formatted, automatically updated reports
    - Efficiency relies on Data Cubes
        - Data Cubes store pre-computed answers to most likely queries
        - Not that useful when dealing with arbitrary queries



- Crazy idea behind Cached Sufficient Statistics:
    - Let's store answers to **all** conceivable queries!
        - That should minimize the user-perceived response time
        - And, it would enable unconstrained, massive scale data mining

Example: Mining Categorical Data

Mining categorical data is quite often all about counting (co-)occurrences:

- Association rule learning, Decision trees, Bayesian networks, …
- E[P(ArrackDrinker|SriLankan)] =

  = NumberOf(SriLankan ArrackDrinker)/NumberOf(SriLankan)

It needs to be done in multi-variate data spaces

Standard approach: a Contingency Table (a Data Cube of counts)

Example: Mining Categorical Data

Mining categorical data is quite often all about counting (co-)occurrences:

- Association rule learning, Decision trees, Bayesian networks, …
- E[P(ArrackDrinker|SriLankan)] =

  = NumberOf(SriLankan ArrackDrinker)/NumberOf(SriLankan)

It needs to be done in multi-variate data spaces

Standard approach: a Contingency Table (a Data Cube of counts)

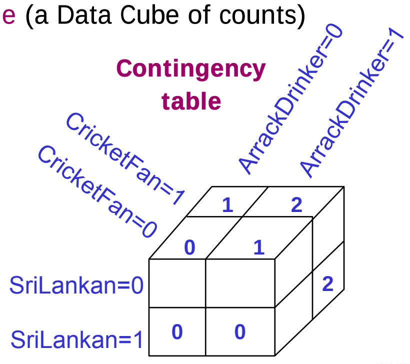
Raw data

N=7 Records

SriLankan?	ArrackDrinker?	CricketFan?
1	0	1
0	1	0
0	1	1
0	0	1
1	1	1
0	1	1
1	1	1

M=3 Attributes

Contingency table



Example: Mining Categorical Data

Mining categorical data is quite often all about counting (co-)occurrences:

- Association rule learning, Decision trees, Bayesian networks, …
- E[P(ArrackDrinker|SriLankan)] =
  = NumberOf(SriLankan ArrackDrinker)/NumberOf(SriLankan)

It needs to be done in multi-variate data spaces

Standard approach: a Contingency Table (a form of Data Cube)

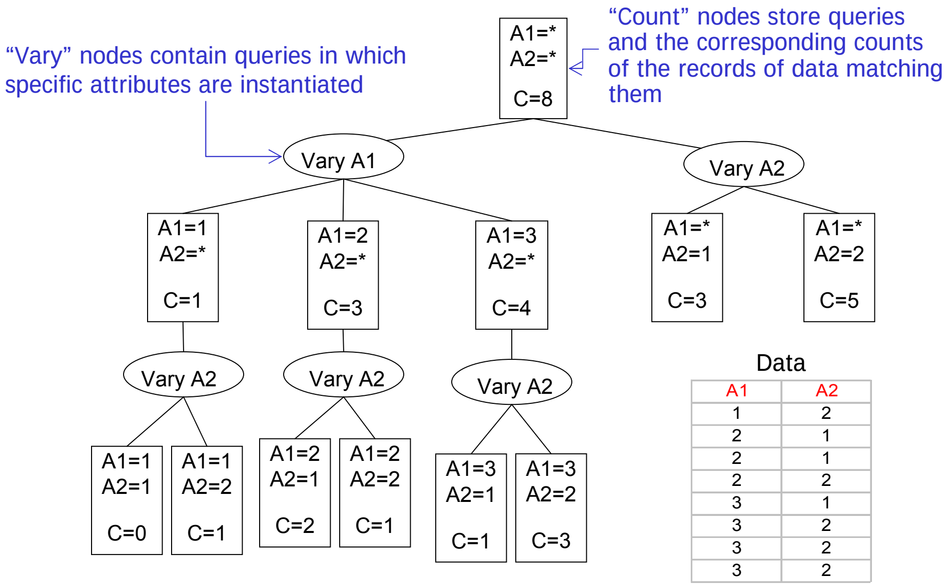
Complaint:

- Contingency Tables can reach enormous sizes (numbers of cells) if the underlying data is highly dimensional and if the involved variables have high arities (i.e. they can assume many different values)
  That may (and it often does) kill the purpose

Is there a better way?

Yes! Use **AD-Trees (All-Dimensional Trees)** [Moore & Lee 1998]

Example: Fully Developed AD-Tree for a 2-D Dataset

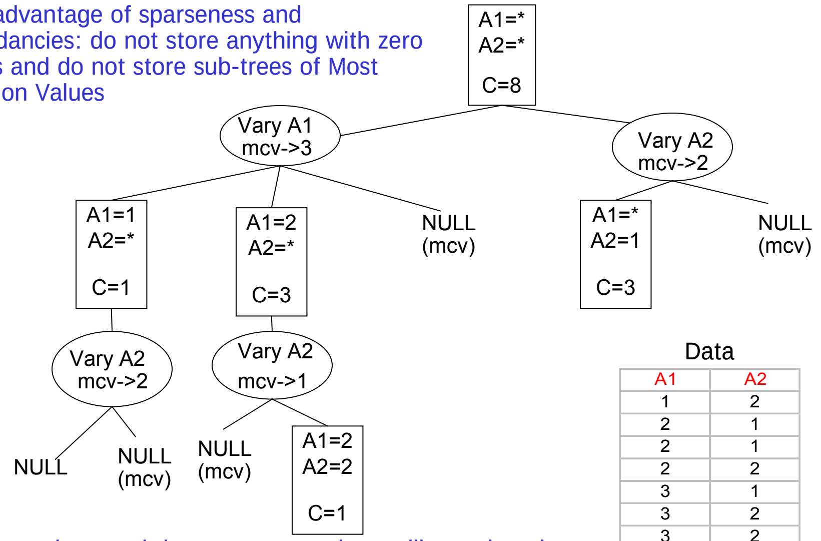


Data

A1	A2
1	2
2	1
2	1
2	2
3	1
3	2
3	2
3	2

AD-Tree: A Smarter Version

Take advantage of sparseness and redundancies: do not store anything with zero counts and do not store sub-trees of Most Common Values



Data

A1	A2
1	2
2	1
2	1
2	2
3	1
3	2
3	2
3	2

This tree takes much less memory and we still can cheaply compute all the removed pre-computed counts!

Practical Benefits of AD-Trees and Other Cached Sufficient Statistics Structures

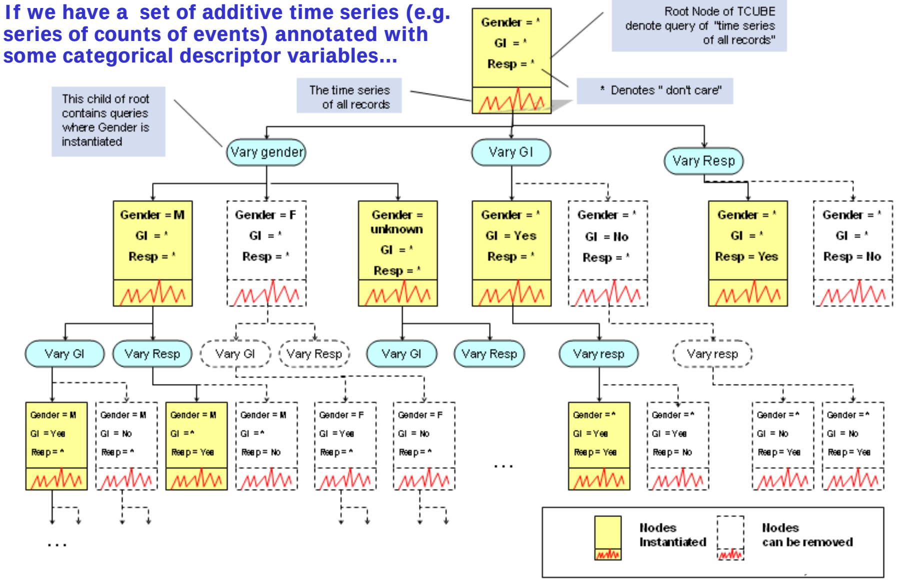
Dramatic speedups of data access time with respect to other, previously considered efficient, methods:

- **AD-Trees:** 1-4 orders of magnitude savings in processing time required by computationally intensive data mining processes (attainable if the data is at least partially correlated)

  Example: 1,580,000 Galaxies, 27 binary attributes per galaxy, time to build AD-Tree: 4 minutes, tree memory: 2 Megs, time required to execute 50,000 iterations of Bayesian network structure search: less than 2 minutes (previously 1.3 days)

- **Kd-Trees:** useful to represent multivariate continuous data up to ~8-D
  - Gaussian mixture density modeling and clustering: Speedups of 8 to 1000 times vs. an efficient, but not kd-tree based implementation [Moore 1999]
  - K-means clustering: Speedups of 150+ times on real-world astrophysical data [Pelleg & Moore 1999]
  - Non-parametric multi-resolution regression: Speedups of 3—100 times [Deng & Moore 1995]
  - Spatial Scan Statistic: 50 times speedups in analyzing nationwide OTC pharmacy sales [Neill 2006]
- **Metric Trees:** useful to represent highly multivariate continuous data
  - Mining data up to 10,000 (yes! ten thousand!) dimensions reveals 2.5 to 2,000-fold speedups w.r.t. otherwise efficient classical approaches to k-means clustering, grouping attributes and non-parametric anomaly detection [Moore 2000].

T-Cube: Extending AD-Trees to Represent Time Series



If we have a set of additive time series (e.g. series of counts of events) annotated with some categorical descriptor variables…

T-Cube: Evaluation in a Controlled Environment (as of 2007)

- Commercial data cube tools evaluated:
  - TimesTen (Oracle),
  - ANTS Data Server (ANTs Software),
  - extremeDB (McObject),
  - TimeSeries DataBlade (IBM, designed for time series).

	Memory	Complex Query Response Time
Tool 1	**330 MB**	**6.8s**
Tool 2	**231 MB**	**7.6s**
Tool 3	**1+ GB**	**3.5s**
T-Cube	**236 MB**	**22ms**
T-Cube	**845 MB**	**5ms**

All tested on 12 million transactional records of synthetic data with 3 attributes (arities of 1,000; 10 and 5); on a Windows XP machine with 2GB RAM and 2.4GHz CPU.

Format of an example data set

Date	Gender	Place	Complaint	Count
1/1/2006	M	100	GI	4
1/1/2006	M	300	Resp	3
1/1/2006	F	300	Fever	11
1/1/2006	M	200	Resp	3
1/1/2006	F	400	Fever	2
1/2/2006	M	200	GI	1
1/2/2006	F	400	GI	4
1/2/2006	M	300	Resp	2
1/2/2006	F	300	Fever	5
1/2/2006	M	200	GI	6
1/3/2006	M	200	GI	2
1/3/2006	F	300	Resp	1
1/3/2006	M	100	Fever	4
1/3/2006	F	300	GI	2
1/3/2006	F	400	GI	3

Summary: So, How Efficient Representation of Data Can Support Biomedical Security?

1. By enabling automated exhaustive searches for events of interest through large collections of data
   - Exhaustive search guarantees that important patterns are never missed
   - Efficient representation of data enables exhaustive search where it was never considered feasible
2. By enabling interactive, ad-hoc, drill-down investigations
   - Quick response times to ad-hoc queries make it possible to interactively navigate data for clues or for confirmations of the automatic detections
3. By enabling automated explanation of detected patterns
   - Fast access to data also supports answering follow-up questions such as “What aspects contribute the most to the observed patterns?”
   - Also, some previously prohibitively expensive but helpful statistical tests become feasible.

Demo

Application of T-Cube Web Interface
in the Real-Time Bio-surveillance Project
being deployed in
Sri Lanka and Tamil Nadu

What's Behind and What's Ahead

- We do a lot of fundamental and applied research
  - Clever data structures
  - Smart, computationally efficient and numerically accurate algorithms
  - Striving for Practical Autonomy
  - Complementing evidence available in data with human expertise
  - Our research is motivated by the actual needs of the end users
- Today, the data available to analysts is too great for them to internalize
- In 10-20 years, the complexity of the concepts will be too great to internalize
- At the same time, the nature and types of objectives will be changing too quickly to permit off-line analyses
- Efficient (computationally and statistically) machine learning will make the timely and accurate analyses feasible.

Acknowledgements

Funding for the presented research came from:

- United States Department of Agriculture
- Centers of Disease Control and Prevention (USA)
- National Science Foundation (USA)
- International Development Research Centre of Canada (via LIRNEasia)

Contact

Carnegie Mellon

Auton Lab

Carnegie Mellon University
5000 Forbes Avenue, NSH 3121
Pittsburgh, PA 15213-3890, USA

Artur Dubrawski
Director, Auton Lab
Systems Scientist, Robotics Institute
Adjunct Professor, Heinz School of Public Policy and Management

Tel: 412-268-6233
Fax: 412-268-7350
E-mail: awd@cs.cmu.edu

www.autonlab.org