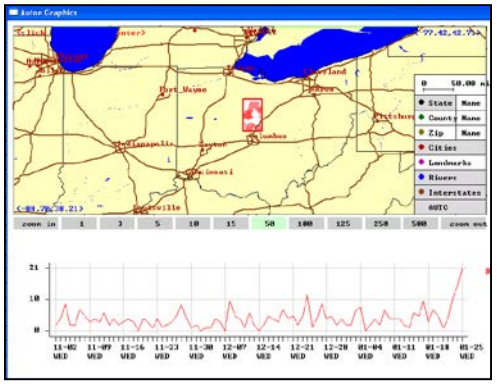
Machine Learning, Big Data, and Development

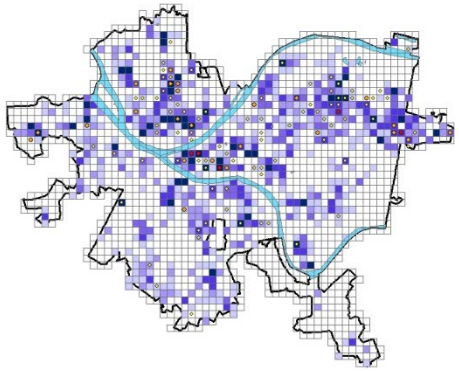
Daniel B. Neill, Ph.D.

H.J. Heinz III College
Carnegie Mellon University
E-mail: neill@cs.cmu.edu

Center for Urban Science and Progress
New York University
E-mail: daniel.neill@nyu.edu

Daniel B. Neill (neill@cs.cmu.edu)
Associate Professor of Information Systems, Heinz College, CMU
Director, Event and Pattern Detection Laboratory
Courtesy Associate Professor of Machine Learning and Robotics





Disease Surveillance: Very early and accurate detection of emerging outbreaks.

Law Enforcement: Detection, prediction, and prevention of "hot-spots" of violent crime.

Medicine: Discovering new "best practices" of patient care, to improve outcomes and reduce costs.

My research is focused at the intersection of **machine learning** and **public policy**, with two main goals:

1) Develop new machine learning methods for better (more scalable and accurate) **detection** and **prediction** of events and other patterns in massive datasets.
2) Apply these methods to improve the quality of public health, safety, and security.

Why machine learning?

Critical importance of addressing policy challenges: disease, crime, terrorism, poverty, environment…

Increasing size and complexity of available data, thanks to the rapid growth of new and transformative technologies.

Much more computing power, and scalable data analysis methods, enable us to extract actionable information from all of this data.

Machine learning techniques have become increasingly essential for policy analysis, and for the development of new, practical information technologies that can be directly applied to address critical challenges for the public good.

Today's lecture

Part 1 (9:30-11am):

- Motivation for using machine learning (ML) to analyze economic and development data.
- Overview of ML problem paradigms and commonly used methods; when to use each.

Part 2 (11:15am-12:30pm):

- Specific challenges and ML solutions for working with development data.
- Cutting edge ML methods and applications.

Assumptions about this audience

- Ph.D. economists familiar with basic data analysis techniques from statistics and econometrics, e.g., linear/logistic regression.
  - No prior knowledge of ML is assumed.
- Data-driven analyses of potential interest:
  - Predicting conflict, civil unrest, real GDP growth, inflation, poverty, loan defaults, currency collapse, trade flows, …

What is machine learning?

Machine Learning (ML) is the study of systems that improve their performance with experience (typically by **learning** from data).

“A computer program is said to learn from experience E wrt. some class of tasks T and performance measure P, if its performance at tasks in T as measured by P improves with experience.” (T. Mitchell)

“Learning denotes changes in the system that are adaptive in the sense that they enable it to do a task, or tasks drawn from the same population, more efficiently and effectively next time.” (H. Simon)

Learning as **generalization**: the ability to perform a task in a situation which has never been encountered before!

ML and related fields

Machine Learning (ML) is the study of systems that improve their performance with experience (typically by **learning** from data).

Artificial Intelligence (AI) is the science of automating complex behaviors such as learning, problem solving, and decision making.

Data Science (or Data Mining) is the process of extracting useful information from massive quantities of complex data.

I would argue that these are not three distinct fields of study! While each has a slightly different emphasis, there is a tremendous amount of overlap in the problems they are trying to solve and the techniques used to solve them.

Many of the techniques we will learn are **statistical** in nature, but are very different from classical statistics.

ML systems and methods:

Scale up to large, complex data
Learn and **improve** from experience
Perceive and **change** the environment
Interact with humans or other agents
Explain inferences and decisions
Discover new and useful patterns

ML and related fields

Machine Learning (ML) is the study of systems that improve their performance with experience (typically by **learning** from data).

Artificial Intelligence (AI) is the science of automating complex behaviors such as learning, problem solving, and decision making.

Data Science (or Data Mining) is the process of extracting useful information from massive quantities of complex data.

I would argue that these are not three distinct fields of study! While each has a slightly different emphasis, there is a tremendous amount of overlap in the problems they are trying to solve and the techniques used to solve them.

Many of the techniques we will learn are **statistical** in nature, but are very different from classical statistics.

<u>Compared to econometrics</u>:

More emphasis on prediction rather than causal inference (esp. classification).

Non-linear and often non-parametric.

Exploratory data analysis: clustering, anomaly detection, etc.

What benefits can ML provide?

… as compared to linear and logistic regression

1) Flexibility

- **Non-linear** and nonparametric models.
- Better modeling of complex relationships between variables without manual specification.
- Higher prediction accuracy, including **generalization** to out-of-sample data.

2) Interpretability

- Understanding **why** a given prediction was made.
- Relevant **variables** and **relationships** between variables.
- Can manage tradeoffs between accuracy & interpretability through choice of methods and parameter settings.

What benefits can ML provide?

… as compared to linear and logistic regression

3) Robustness

- Able to handle **noise** in the data, adjust for **biases**, and infer **missing values**.

4) Scalability and applicability for analyzing “big data”

- Able to draw useful conclusions from massive quantities of difficult data (high-dimensional, varied, noisy, complex, unstructured) including text, images, online social media.
- Able to deal with computational and statistical challenges: run time, multiple testing, overfitting, …
- Extracting relevant structure from emerging data sources such as **satellite imagery** (→ land use, urbanization, poverty, deforestation) or cell phone location traces.

Software tools for data analysis

Many different ML software tools are available for performing different large scale data analysis tasks.

Option 1. Python

Scikit-learn package is widely used, implements most standard ML techniques, data cleaning and preprocessing, and evaluation.

Easy to manipulate data with numpy/pandas packages.

Option 2. Weka data mining toolkit

Free Java open-source software, available for download at:
http://www.cs.waikato.ac.nz/ml/weka/
Weka contains classifiers, clustering, data visualization and preprocessing tools; graphical interface (no programming necessary)

Option 3. R

Lots of specific-function packages (glmnet, rpart, gbm randomForest, …) but less comprehensive coverage of ML.

Lots of other options… Matlab, writing your own code in C or Java, etc.

Today's lecture

Part 1 (9:30-11am):

- Motivation for using machine learning (ML) to analyze economic and development data.
- Overview of ML problem paradigms and commonly used methods; when to use each.

Part 2 (11:15am-12:30pm):

- Specific challenges and ML solutions for working with development data.
- Cutting edge ML methods and applications.

Common ML paradigms: prediction

In **prediction**, we are interested in explaining a specific attribute of the data in terms of the other attributes.

Classification: predict a discrete value

"What disease does this patient have, given his symptoms?"

Regression: estimate a numeric value

"How is a country's literacy rate affected by various social programs?"

Two main goals of prediction

Guessing unknown values for specific instances (e.g. diagnosing a given patient)

Explaining predictions of both known and unknown instances (providing relevant examples, a set of decision rules, or class-specific models).

Example 1: What socio-economic factors lead to increased prevalence of diarrheal illness in a developing country?

Example 2: Predicting yearly trade flows between a country and each of its trading partners given historical data.

Common ML paradigms: modeling

In **modeling**, we are interested in describing the underlying relationships between many attributes and many entities.

Our goal is to produce models of the "entire data" (not just specific attributes or examples) that accurately reflect underlying complexity, yet are simple, understandable by humans, and usable for decision-making.

Relations between entities	Relations between variables
Identifying link, group, and network structures	Identifying significant positive and negative correlations
Partitioning or "clustering" data into subgroups	Modeling dependence structure between multiple variables

Example 1: Can we cluster countries into "growth clubs" where countries in the same group tend to experience economic growth in the same time period?

Example 2: What are the interrelationships between poverty rate and various social determinants of health?

Common ML paradigms: detection

In **detection**, we are interested in identifying relevant patterns in massive, complex datasets.

Main goal: focus the user's attention on a potentially relevant subset of the data.

a) Automatically detect relevant individual records, or groups of records.

b) Characterize and explain the pattern (type of pattern, H_0 and H_1 models, etc.)

c) Present the pattern to the user.

Some common detection tasks

Detecting **anomalous** records or groups

Discovering **novelties** (e.g. new drugs)

Detecting **clusters** in space or time

Removing **noise** or **errors** in data

Detecting **specific patterns** (e.g. fraud)

Detecting emerging **events** which may require rapid responses.

Example 1: Detect emerging outbreaks of disease using electronic public health data from hospitals and pharmacies.

Example 2: Can we detect evidence of corruption in awarding of federal contracts (and correlate these with failed development projects)?

Overview of ML approaches

ML problem paradigms represent a **functional** grouping of methods by what we're trying to accomplish. A related grouping is based on what the data looks like, and in particular, whether we have **labeled** or **unlabeled** data.

Supervised Learning

Data/input Labels/output

$$\begin{array}{ll} x_1 & y_1 \\ x_2 & y_2 \\ \dots & \dots \\ x_N & y_N \end{array}$$

Learn dependence:

$$y = f(x)$$

Discrete y = classification
Continuous y = regression

Overview of ML approaches

ML problem paradigms represent a **functional** grouping of methods by what we're trying to accomplish. A related grouping is based on what the data looks like, and in particular, whether we have **labeled** or **unlabeled** data.

Supervised Learning

Data/input Labels/output

$$x_1 \quad y_1$$
$$x_2 \quad y_2$$
$$\dots \quad \dots$$
$$x_N \quad y_N$$

Learn dependence:

$$y = f(x)$$

Discrete y = classification
Continuous y = regression

Semi-supervised learning:

Only some data points are labeled; the goal is still typically prediction.

Active learning:

Choose which data points to label; the goal is still typically prediction.

Reinforcement learning:

Sequential actions with delayed rewards; goal is to learn optimal action in each state.

Unsupervised learning:

No labels, just input data x_i. Various goals including clustering, modeling, anomaly detection, etc.

Supervised learning in basic stats

ML problem paradigms represent a **functional** grouping of methods by what we're trying to accomplish. A related grouping is based on what the data looks like, and in particular, whether we have **labeled** or **unlabeled** data.

Supervised Learning

Data/input Labels/output

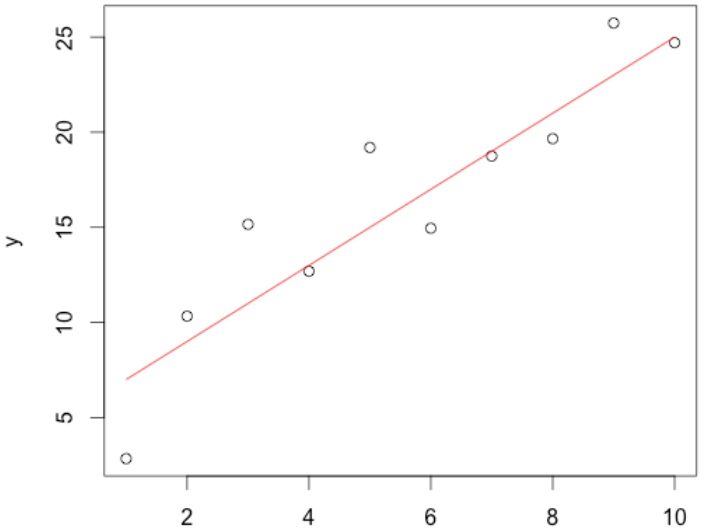
$$\begin{matrix} x_1 & y_1 \\ x_2 & y_2 \\ \dots & \dots \\ x_N & y_N \end{matrix}$$

Learn dependence:

$$y = f(x)$$

Discrete y = classification
Continuous y = regression

Linear regression



$$y = w_1 x + w_0 + \varepsilon$$

$$y = w^T x + \varepsilon$$

$$p(y|x, w, \sigma) = \mathcal{N}(y|w^T x, \sigma^2)$$

Supervised learning in basic stats

ML problem paradigms represent a **functional** grouping of methods by what we're trying to accomplish. A related grouping is based on what the data looks like, and in particular, whether we have **labeled** or **unlabeled** data.

Supervised Learning

Data/input $x_1, x_2, \ldots, x_N$

Labels/output $y_1, y_2, \ldots, y_N$

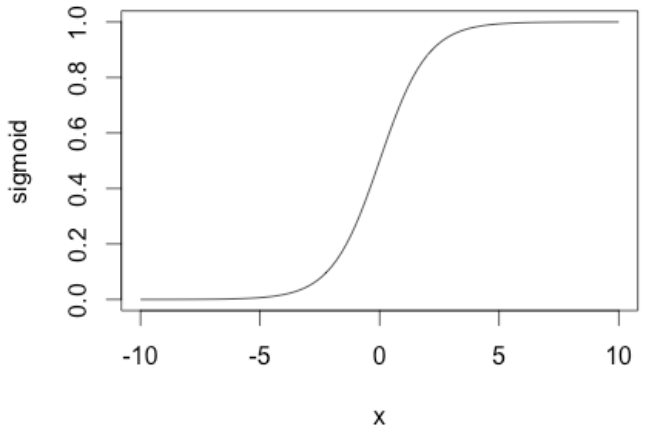
Learn dependence:

$$y = f(x)$$

Discrete y = classification
Continuous y = regression

Logistic regression

(= generalized LR for classification)



$$y \sim Bernoulli\,(f(w^T x))$$

$$f(x) = \sigma(x) = \frac{e^x}{1 + e^x} = \frac{1}{1 + e^{-x}}$$

ML methods: supervised learning

Classification and regression are the most developed areas of machine learning, with many possible approaches to use. How to choose which one?

Supervised Learning

Data/input Labels/output

$$x_1 \quad y_1$$
$$x_2 \quad y_2$$
$$\dots \quad \dots$$
$$x_N \quad y_N$$

Learn dependence:

$$y = f(x)$$

Discrete y = classification
Continuous y = regression

Three main criteria:

- **Applicability**
  - Classification vs. regression
    - Do we need class probabilities?
  - Input variable type (real/discrete/mixed)
  - Dataset size and dimensionality
    - Computationally feasible?
  - Assumptions/inductive bias
- **Performance** (measured empirically)
  - Accuracy (classification)
  - MSE (regression)
  - Precision/recall for skewed data
  - In sample vs. out of sample?
- **Interpretability**
  - Explaining relationships between vars.
  - Explaining individual predictions
  - Often tradeoffs with accuracy

ML methods: supervised learning

Classification and regression are the most developed areas of machine learning, with many possible approaches to use. How to choose which one?

Supervised Learning

Data/input Labels/output

$$x_1 \quad y_1$$
$$x_2 \quad y_2$$
$$\ldots \quad \ldots$$
$$x_N \quad y_N$$

Learn dependence:

$$y = f(x)$$

Discrete y = classification
Continuous y = regression

Approaches we will consider:

- Decision Trees
- Random Forests
- Bayesian Networks
- Naïve Bayes
- Gaussian Processes
- Deep Learning (neural networks)

Lots of other choices, such as k-nearest neighbor, support vector machines, boosting, …

Case study: predicting burden of disease

Diarrheal illness is one of the world's main causes of preventable death.

As of 2015, diarrhea was the 9th leading cause of death globally and was responsible for 8.6% of deaths among children <5 years.

As an illustrative example, I will show results of various machine learning algorithms for predicting disease burden of diarrheal illness at the country level, as measured in disability-adjusted life years lost per 1000 population.

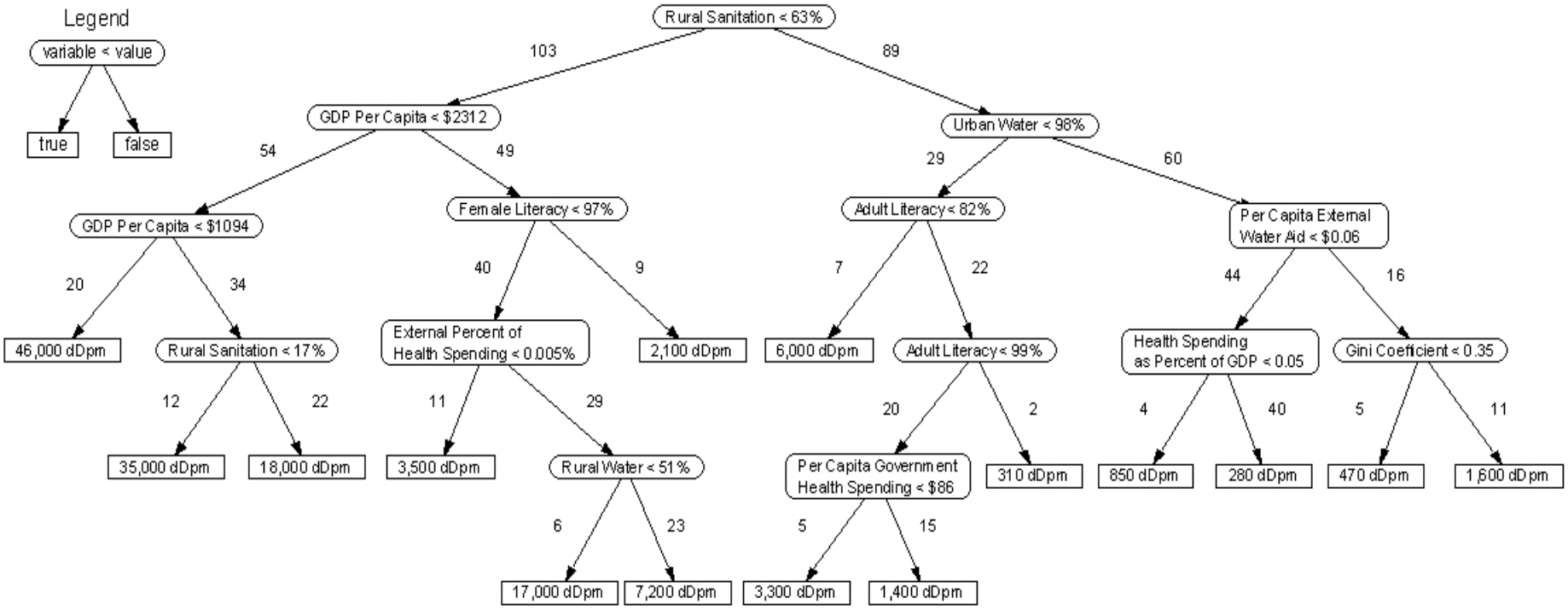
130 countries, 14 predictor variables:

Fraction of peace in last 10 years
ODA for water (per capita, in dollars)
Renewable water resources (cubic meters per capita per year)
Fractions of urban/rural populations with sustainable access to improved water/sanitation (4)
Total health expenditures as a percentage of GDP
Fractions of total health expenditures provided by the government and external sources (2)
Per capita government health expenditures at the average exchange rate
GDP per capita (in dollars)
Adult and female literacy rates (2)

Data from: ST Green et al., Determinants of national diarrheal disease burden, *Environ. Sci. Technol.*, 2009.

Decision Trees (a.k.a. CART)

An ML approach for classification or regression that learns a tree-structured set of decision rules through top-down recursive partitioning.



<u>Learning the tree is straightforward and scalable:</u>

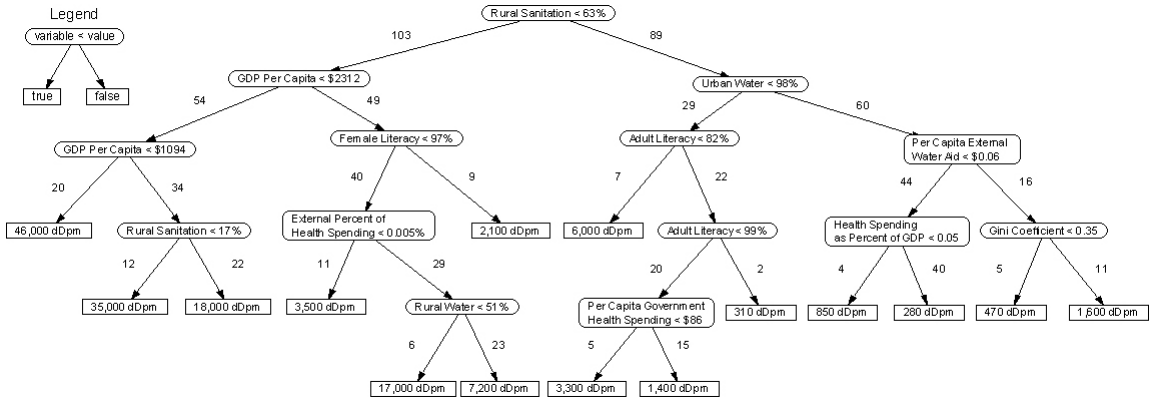
At each step, choose the question (true or false?) that best splits the data.

For classification: maximize information gain; regression: minimize MSE.

Once you have the tree, you can easily make predictions for new data points.

Decision Trees (a.k.a. CART)

How do decision trees stack up to the three criteria that we discussed earlier?



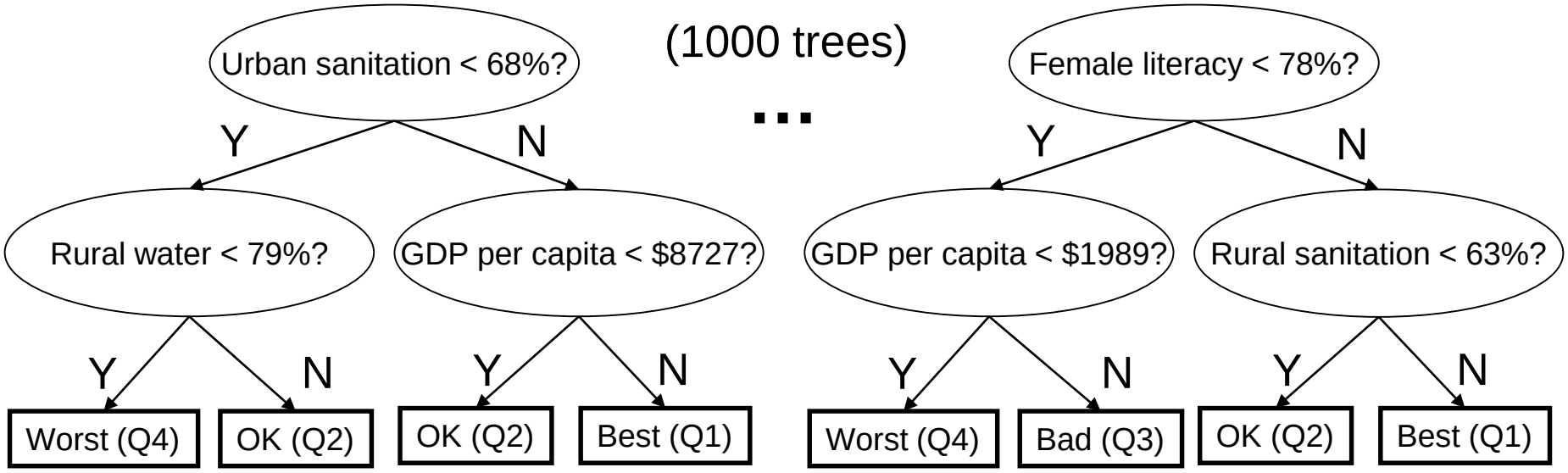
- **Applicability – high**
  - Good for classification or regression; can handle real, discrete, or mixed inputs.
  - Very scalable: can handle large numbers of records and attributes.
  - Easy to use out of the box.
  - Can model non-linear relationships & interactions between multiple variables.
- **Performance – moderate**
  - Decent accuracy, but lower than other, more complex techniques we'll discuss.
  - For this particular problem, about the same RMS as linear regression, or about the same accuracy as logistic regression for predicting quartiles (63% vs. 65%).
- **Interpretability – high**
  - Can identify which variables most impact prediction, and how.
  - Irrelevant variables will not be included → automatic feature selection.
  - Easy to see the sequence of decisions that led to each individual prediction.

Random Forests

A natural extension of decision trees, which gains increased accuracy at the expense of interpretability, is to learn **many** trees instead of just one.

Each tree is learned from a **random subsample** of the data:
Bootstrap sample of data records (draw randomly with replacement)
For each split of the data, consider only a randomly chosen subset of attributes.

Average all trees' predictions (regression) or take a **majority vote** (classification).



Each tree may be less accurate (~60%) but majority vote is **more** accurate (71%)!

Random Forests

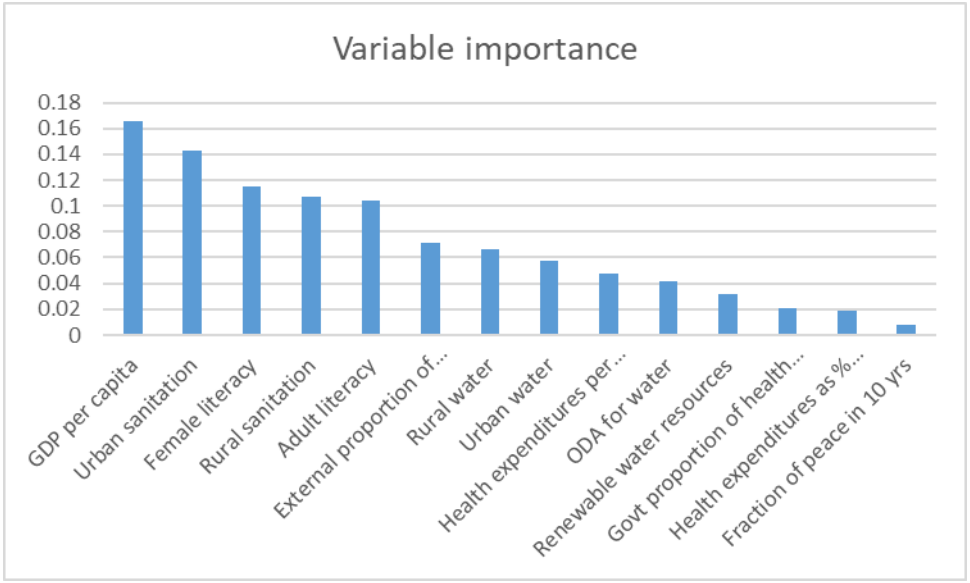
How do random forests stack up to our criteria?

- **Applicability – high**
  - Good for classification or regression
  - Can handle real, discrete, or mixed inputs.
  - Scalable: can handle large # of records and attributes.
    - Computation time multiplied by # trees, but easy to parallelize
  - Easy to use out of the box.
  - Can model non-linear relationships and interactions between multiple variables.

Random Forests

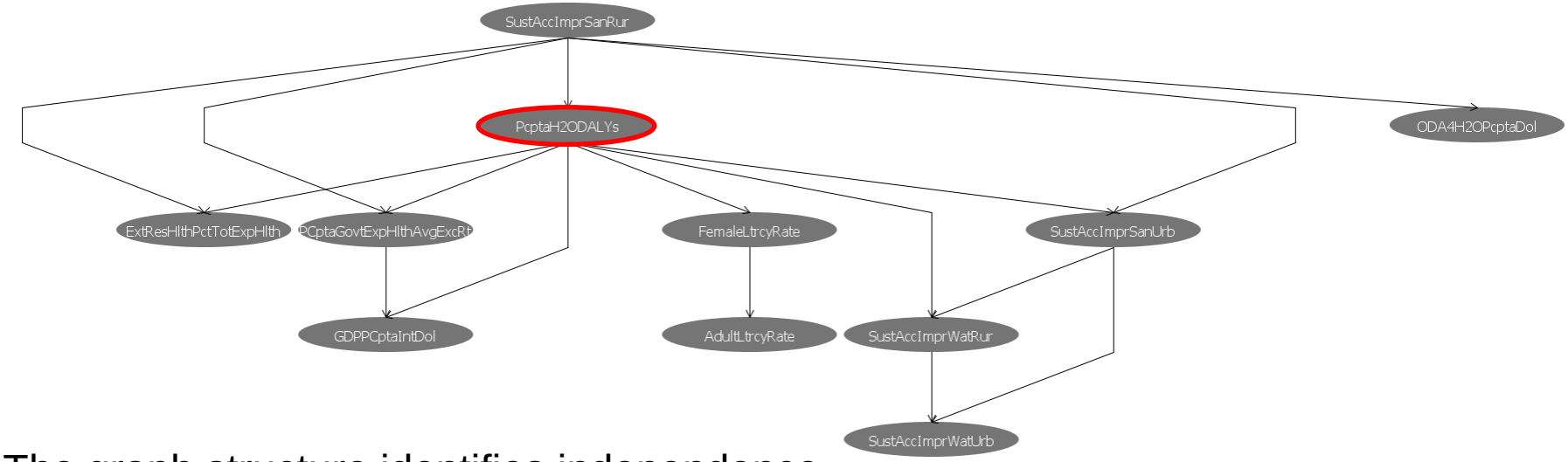
How do random forests stack up to our criteria?

- **Performance – high**
  - High accuracy: near optimal performance for many problems.
  - Performance + ease of use → very popular ML approach!
- **Interpretability – low to moderate**
  - 1,000 trees are much harder to interpret than a single tree! Can measure **variable importance** but hard to see how each variable impacts prediction.



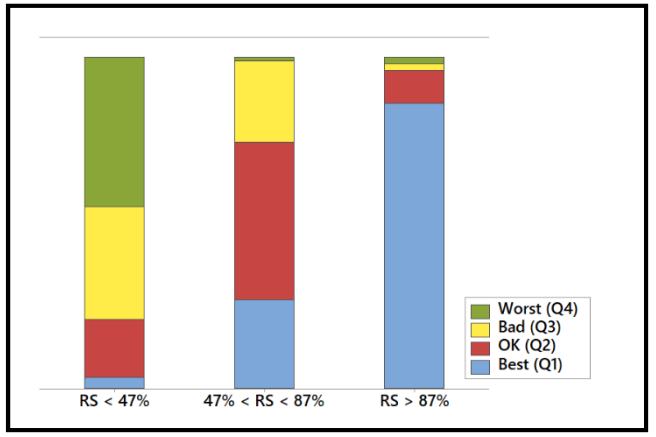
Bayesian Networks

Bayes Nets are an approach for modeling the probabilistic or causal relationships between multiple variables. They can be learned from data and used for prediction.



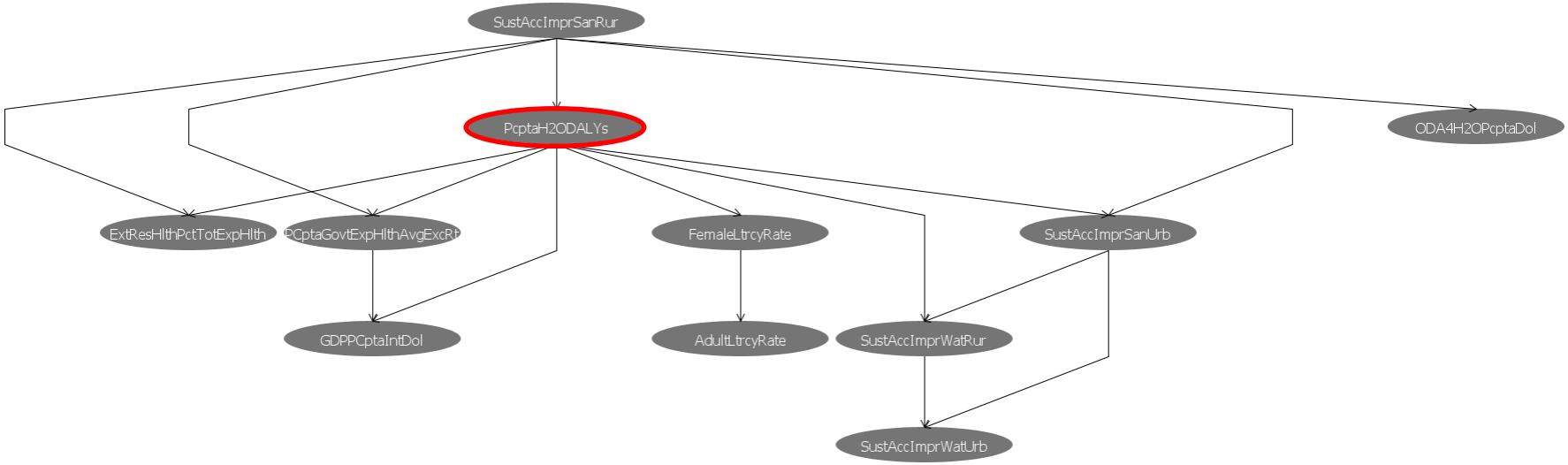
The graph structure identifies independence between variables: burden of disease is conditionally independent of urban water given rural water and urban sanitation.

Each variable in the graph also has an associated table representing its distribution conditional on its parents' values. Here's burden of disease conditional on rural sanitation.



Bayesian Networks

Bayes Nets are an approach for modeling the probabilistic or causal relationships between multiple variables. They can be learned from data and used for prediction.



Once we have learned the Bayes Net, we can infer the distribution of any variable conditional on any subset of the other variables.

For example, the central African country of Chad, where only 1% of the rural population has access to improved sanitation, and female literacy rate is 39%, has a predicted 89% chance of belonging to the highest burden of disease quartile (>27,000 dDpm). Actual value: 44,930.

Bayesian Networks

Bayes Nets are an approach for modeling the probabilistic or causal relationships between multiple variables. They can be learned from data and used for prediction.

Different Bayes Net learning methods can be used depending on whether we want to optimize **prediction accuracy** for a particular variable, **model fit** across all variables, or **causal validity**. Here's the causal Bayes net learned by the PC structure learning algorithm.

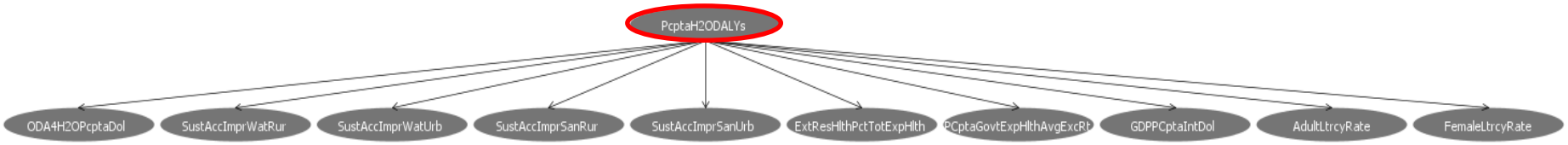
Bayesian Networks

How do Bayes Nets stack up to our criteria?

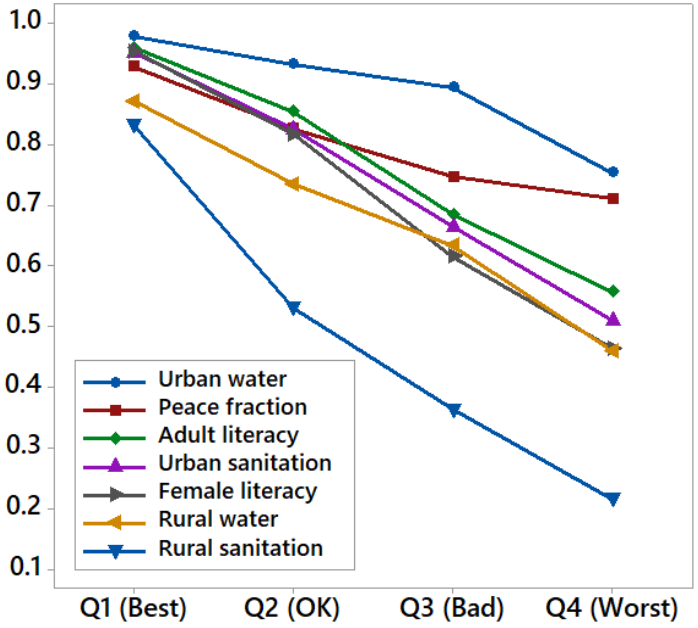
- **Applicability – moderate**
  - Generally used for **classification** only; can predict **probability** of each class.
  - Typically discrete-valued inputs (if real-valued, discretize or treat as Gaussian).
  - Can handle large # of records, but structure search and inference can be computationally expensive or infeasible if # of attributes is large.
- **Performance – moderate to high**
  - Prediction accuracy depends on the learned graph structure.
    - Class-conditional models (target variable as parent of other attributes) often outperform causal models (target variable as child of its direct causes).
  - For our case study, accuracy varied from 63% (~ DTs) to 72% (~RFs).
- **Interpretability – moderate to high**
  - Graph structure models probabilistic or causal relationships between variables.
  - Conditional independence relationships → which vars will impact prediction. ("Markov blanket": target variable's parents, children, and parents of children.)
  - Learned conditional probability tables are informative re. the direction of effect.
  - Can detect **anomalies**: lowest-likelihood records given the Bayes Net model.

Naïve Bayes

One very simple and highly interpretable approach is **Naïve Bayes**, a special case of Bayes Nets where, instead of learning the dependencies, we **assume** that all other variables are conditionally independent given the target class to be predicted.



- **Applicability – moderate to high**
  - Fast and scalable to large numbers of records and attributes.
- **Performance – moderate**
  - Good for small amounts of data and when NB assumption is reasonable.
  - Can fail for redundant variables.
  - Typically Bayes Nets perform better.
- **Interpretability – high**
  - Very easy to interpret the class-conditional distribution of each var.



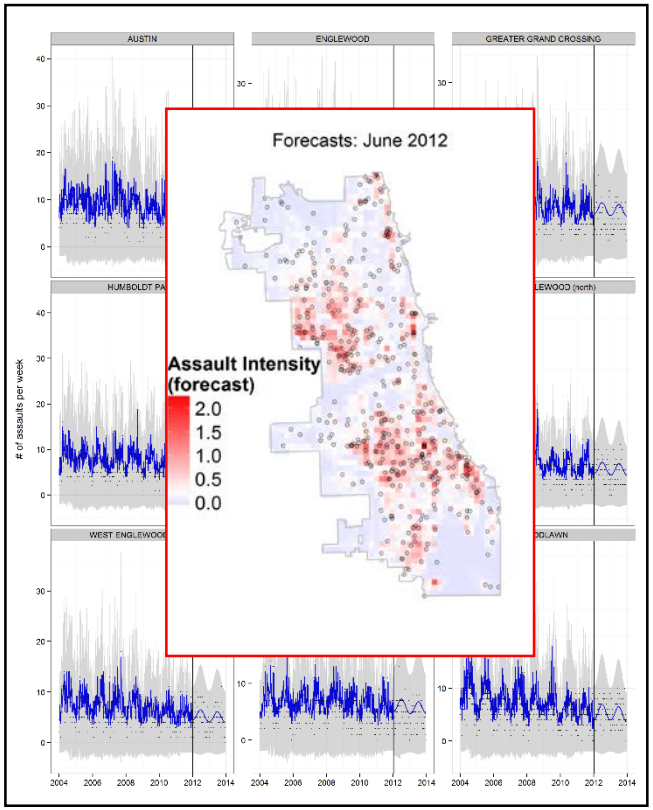
Gaussian Processes

Gaussian processes are a useful way to **model** and **predict** with **dependent** data (e.g., time series forecasting, spatial regression). They can **learn** the dependence structure from data, and produce prediction intervals rather than just point estimates.

Observations are assumed to be jointly Gaussian but not independent. Correlation is typically a decreasing function of distance; can also model seasonal trends, etc.

In Flaxman et al. (2015), we applied GPs to long-term, local-area spatio-temporal forecasting of crime in Chicago. We were able to forecast crime levels up to 12 months in advance with higher accuracy than the previous state of the art.

Learning an **interpretable model** of the correlation structure revealed five components: a long-term trend of decreasing crime, yearly (seasonal) and longer-term periodic trends, and correlations on two short time scales.

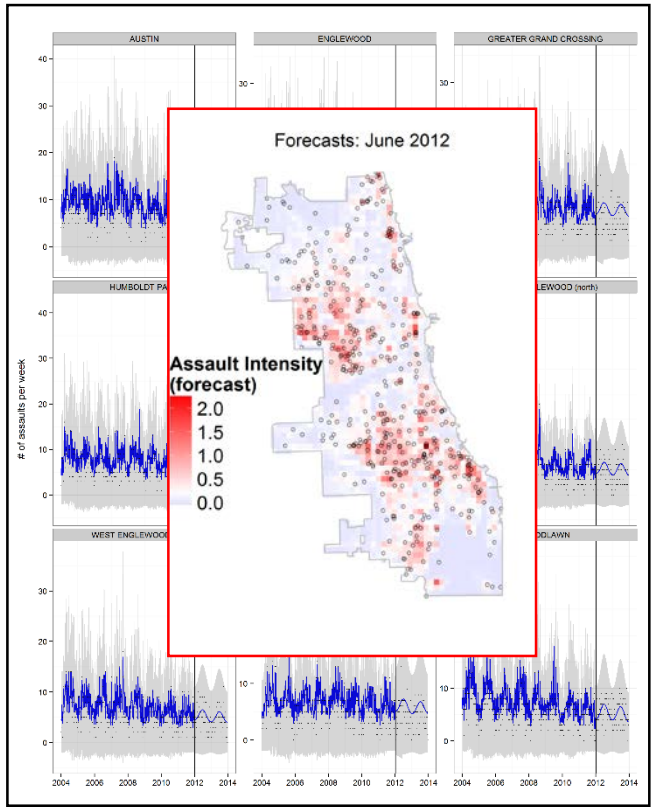


Gaussian Processes

Gaussian processes are a useful way to **model** and **predict** with **dependent** data (e.g., time series forecasting, spatial regression). They can **learn** the dependence structure from data, and produce prediction intervals rather than just point estimates.

Observations are assumed to be jointly Gaussian but not independent. Correlation is typically a decreasing function of distance; can also model seasonal trends, etc.

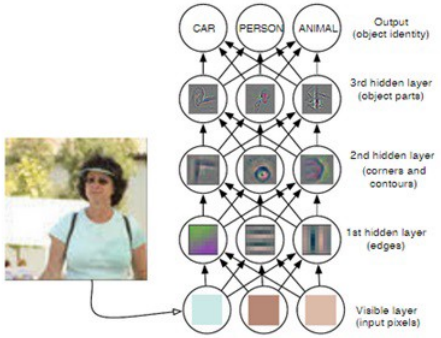
- **Applicability – moderate/specialized**
  - Often used for regression with spatial and temporal data, or other cases where the typical assumption of iid data is violated.
  - Computation is very expensive for large datasets, but can be made more efficient.
- **Performance – very high**
  - State of the art for many time series forecasting and spatial regression tasks.
- **Interpretability – low to moderate**
  - Black box prediction w/ confidence intervals.
  - Learn dependence structure from data.

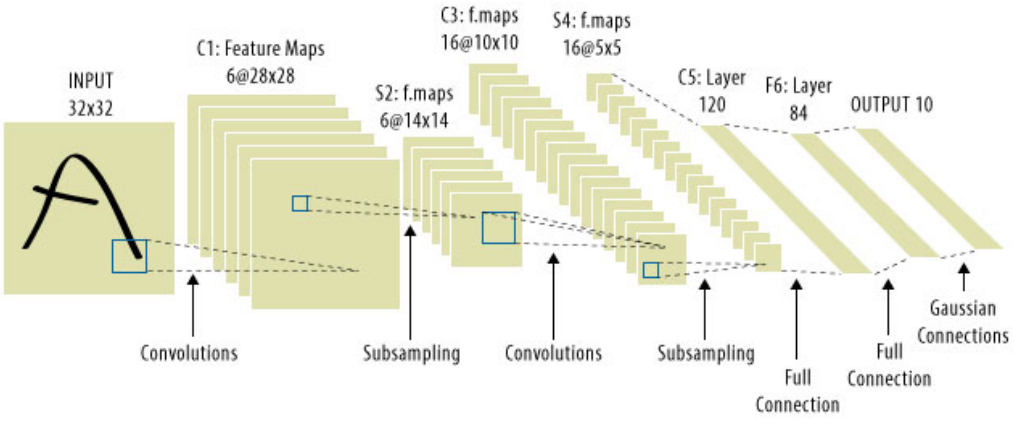


Deep Learning with Neural Networks

Deep neural networks have become the hottest subarea of ML and achieve state-of-the-art performance (by far!) on many tasks ranging from object recognition to game playing to autonomous vehicle control.

They are particularly good at difficult tasks such as **computer vision**, where they can learn abstract feature representations that outperform manually constructed features.





Nodes of the network are arranged in many layers.

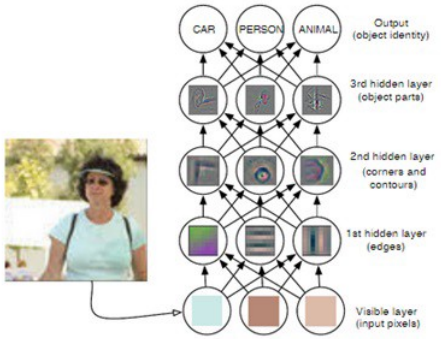
Each node applies a function to its inputs, with weights learned jointly from the training data.

Result: can accurately fit arbitrarily complex functions, given sufficient data.

Deep Learning with Neural Networks

Deep neural networks have become the hottest subarea of ML and achieve state-of-the-art performance (by far!) on many tasks ranging from object recognition to game playing to autonomous vehicle control.

- **Applicability – moderate/specialized**
  - Computer vision and other tasks involving classification, regression, or sequential decisions from complex, unstructured data.
  - Requires huge amount of data and processing power to achieve high performance.
  - Ease of use: low but improving. Lots of engineering and tweaking models.



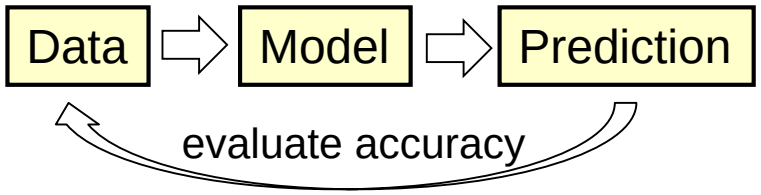
- **Performance – very high**
  - State of the art performance for learning very complex functions given sufficient data.
  - Can be fooled by adversarial examples.

- **Interpretability – low**
  - Black box predictor, very hard to interpret. Much recent work on interpreting layers for computer vision.

Evaluating and tuning prediction approaches

One key idea from machine learning is the use of **data splitting** to both **measure** and **optimize** a method's out-of-sample performance.

Q: What's wrong with this picture?



Answer: Overfitting!

The model may just learn to regurgitate the training data, so that it classifies all examples correctly but has no ability to generalize to previously unseen data.

For the burden of disease example, imagine a very large decision tree with one leaf for each country name: IF Country = "Algeria" THEN predict dDpm = 9448, etc.

Perfect performance in sample, terrible performance out of sample!

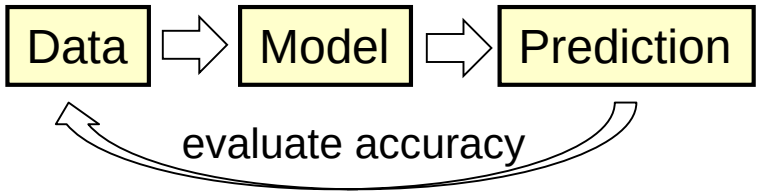
Solution: split the data into a "training" set (80%) and a "test" set (20%). Learn the model on the training set, evaluate prediction accuracy on the test set.

The resulting estimate of out-of-sample accuracy is nearly unbiased (actually, slightly conservative since we are using less training data).

Evaluating and tuning prediction approaches

One key idea from machine learning is the use of **data splitting** to both **measure** and **optimize** a method's out-of-sample performance.

Q: What's wrong with this picture?



Answer: Overfitting!

The model may just learn to regurgitate the training data, so that it classifies all examples correctly but has no ability to generalize to previously unseen data.

For the burden of disease example, imagine a very large decision tree with one leaf for each country name: IF Country = "Algeria" THEN predict dDpm = 9448, etc.

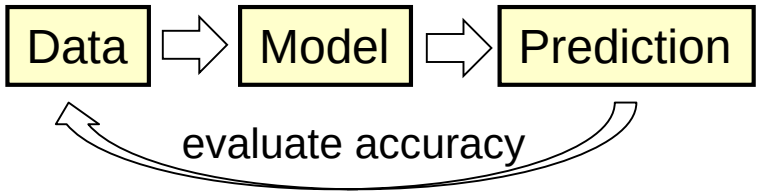
Perfect performance in sample, terrible performance out of sample!

Even better: average performance over multiple train/test splits. Still nearly unbiased, but a much better (lower variance) estimate of performance.

Evaluating and tuning prediction approaches

One key idea from machine learning is the use of **data splitting** to both **measure** and **optimize** a method's out-of-sample performance.

Q: What's wrong with this picture?



Answer: Overfitting!

The model may just learn to regurgitate the training data, so that it classifies all examples correctly but has no ability to generalize to previously unseen data.

For the burden of disease example, imagine a very large decision tree with one leaf for each country name: IF Country = "Algeria" THEN predict dDpm = 9448, etc.

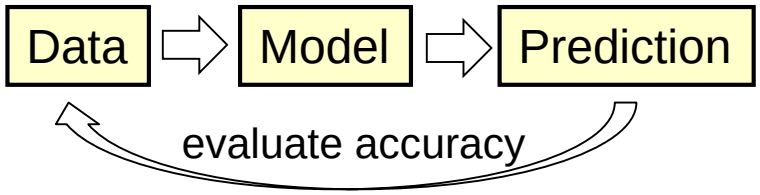
Perfect performance in sample, terrible performance out of sample!

Even better: average performance over multiple train/test splits. Still nearly unbiased, but a much better (lower variance) estimate of performance.

Evaluating and tuning prediction approaches

One key idea from machine learning is the use of **data splitting** to both **measure** and **optimize** a method's out-of-sample performance.

Q: What's wrong with this picture?



Answer: Overfitting!

The model may just learn to regurgitate the training data, so that it classifies all examples correctly but has no ability to generalize to previously unseen data.

For the burden of disease example, imagine a very large decision tree with one leaf for each country name: IF Country = "Algeria" THEN predict dDpm = 9448, etc.

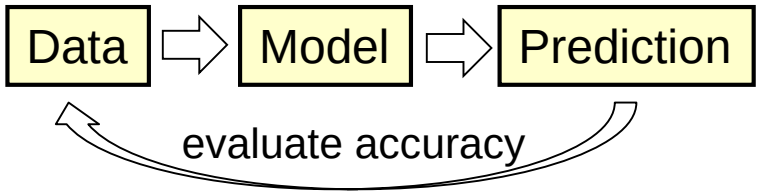
Perfect performance in sample, terrible performance out of sample!

Even better: average performance over multiple train/test splits. Still nearly unbiased, but a much better (lower variance) estimate of performance.

Evaluating and tuning prediction approaches

One key idea from machine learning is the use of **data splitting** to both **measure** and **optimize** a method's out-of-sample performance.

Q: What's wrong with this picture?



Answer: Overfitting!

The model may just learn to regurgitate the training data, so that it classifies all examples correctly but has no ability to generalize to previously unseen data.

For the burden of disease example, imagine a very large decision tree with one leaf for each country name: IF Country = "Algeria" THEN predict dDpm = 9448, etc.

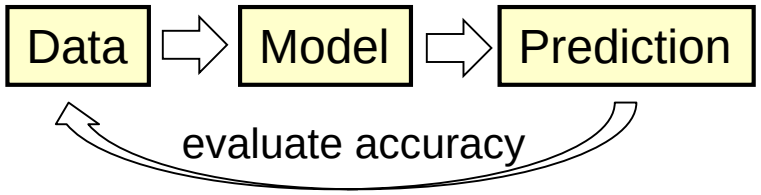
Perfect performance in sample, terrible performance out of sample!

Even better: average performance over multiple train/test splits. Still nearly unbiased, but a much better (lower variance) estimate of performance.

Evaluating and tuning prediction approaches

One key idea from machine learning is the use of **data splitting** to both **measure** and **optimize** a method's out-of-sample performance.

Q: What's wrong with this picture?



Answer: Overfitting!

The model may just learn to regurgitate the training data, so that it classifies all examples correctly but has no ability to generalize to previously unseen data.

For the burden of disease example, imagine a very large decision tree with one leaf for each country name: IF Country = "Algeria" THEN predict dDpm = 9448, etc.

Perfect performance in sample, terrible performance out of sample!

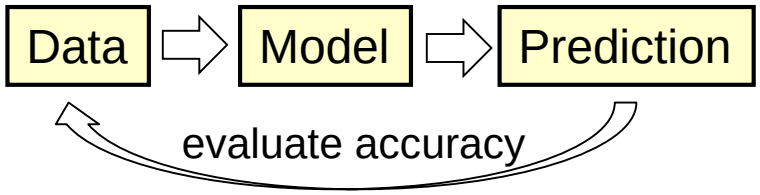
Even better: average performance over multiple train/test splits. Still nearly unbiased, but a much better (lower variance) estimate of performance.

This very useful and commonly used approach is called **cross-validation**.

Evaluating and tuning prediction approaches

One key idea from machine learning is the use of **data splitting** to both **measure** and **optimize** a method's out-of-sample performance.

Q: What's wrong with this picture?



Answer: Overfitting!

The model may just learn to regurgitate the training data, so that it classifies all examples correctly but has no ability to generalize to previously unseen data.

For the burden of disease example, imagine a very large decision tree with one leaf for each country name: IF Country = "Algeria" THEN predict dDpm = 9448, etc.

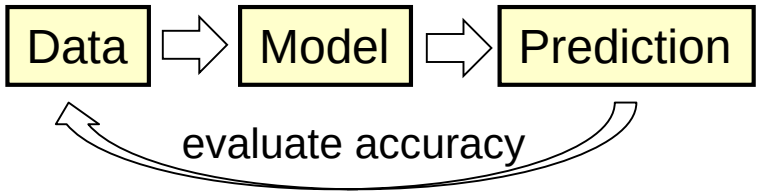
Perfect performance in sample, terrible performance out of sample!

Cross-validation can also be used to optimize prediction methods (both model selection and parameter tuning) by further splitting within the training dataset.

Evaluating and tuning prediction approaches

One key idea from machine learning is the use of **data splitting** to both **measure** and **optimize** a method's out-of-sample performance.

Q: What's wrong with this picture?



Answer: Overfitting!

The model may just learn to regurgitate the training data, so that it classifies all examples correctly but has no ability to generalize to previously unseen data.

For the burden of disease example, imagine a very large decision tree with one leaf for each country name: IF Country = "Algeria" THEN predict dDpm = 9448, etc.

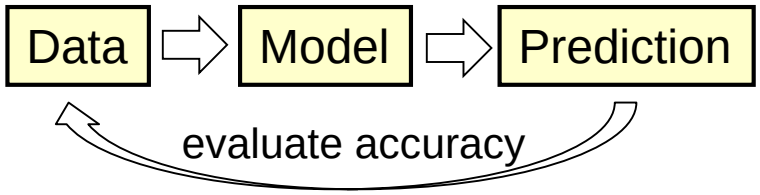
Perfect performance in sample, terrible performance out of sample!

Cross-validation can also be used to optimize prediction methods (both model selection and parameter tuning) by further splitting within the training dataset.

Evaluating and tuning prediction approaches

One key idea from machine learning is the use of **data splitting** to both **measure** and **optimize** a method's out-of-sample performance.

Q: What's wrong with this picture?



Answer: Overfitting!

The model may just learn to regurgitate the training data, so that it classifies all examples correctly but has no ability to generalize to previously unseen data.

For the burden of disease example, imagine a very large decision tree with one leaf for each country name: IF Country = "Algeria" THEN predict dDpm = 9448, etc.

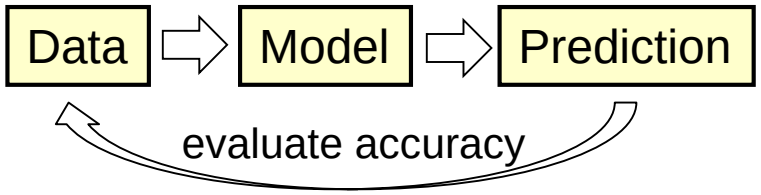
Perfect performance in sample, terrible performance out of sample!

Cross-validation can also be used to optimize prediction methods (both model selection and parameter tuning) by further splitting within the training dataset.

Evaluating and tuning prediction approaches

One key idea from machine learning is the use of **data splitting** to both **measure** and **optimize** a method's out-of-sample performance.

Q: What's wrong with this picture?



Answer: Overfitting!

The model may just learn to regurgitate the training data, so that it classifies all examples correctly but has no ability to generalize to previously unseen data.

For the burden of disease example, imagine a very large decision tree with one leaf for each country name: IF Country = "Algeria" THEN predict dDpm = 9448, etc.

Perfect performance in sample, terrible performance out of sample!

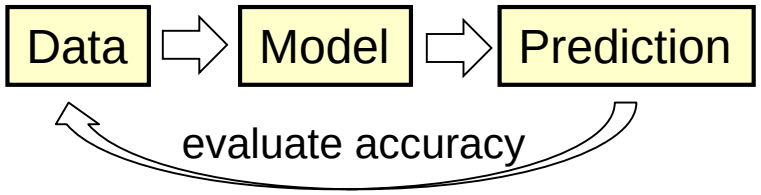
Cross-validation can also be used to optimize prediction methods (both model selection and parameter tuning) by further splitting within the training dataset.

We compare many models and many parameter settings, learning each from part of the training set and evaluating on the held-out portion ("validation set"). Then choose the parameters that **maximize accuracy** on the validation set.

Evaluating and tuning prediction approaches

One key idea from machine learning is the use of **data splitting** to both **measure** and **optimize** a method's out-of-sample performance.

Q: What's wrong with this picture?



Answer: Overfitting!

The model may just learn to regurgitate the training data, so that it classifies all examples correctly but has no ability to generalize to previously unseen data.

For the burden of disease example, imagine a very large decision tree with one leaf for each country name: IF Country = "Algeria" THEN predict dDpm = 9448, etc.

Perfect performance in sample, terrible performance out of sample!

Cross-validation can also be used to optimize prediction methods (both model selection and parameter tuning) by further splitting within the training dataset.

I would argue that parameter tuning via cross-validation is the "secret sauce" that enables machine learning methods to accurately fit complex models to data (without overfitting), leading to high out-of-sample performance.

Questions on Part 1?

Today's lecture

Part 1 (9:30-11am):

- Motivation for using machine learning (ML) to analyze economic and development data.
- Overview of ML problem paradigms and commonly used methods; when to use each.

Part 2 (11:15am-12:30pm):

- Specific challenges and ML solutions for working with development data.
- Cutting edge ML methods and applications.

Machine learning for development

- ML methods have the potential to contribute greatly to human welfare by addressing numerous problems in the developing world.
  - Agriculture, education, governance, poverty reduction, microfinance, human rights, public safety, health care, disease surveillance, disaster response, etc…
- Approach 1: Data-driven policy analysis.
  - Analysis of existing data (combining multiple noisy, incomplete sources, deciding what new data to collect)
- Approach 2: Incorporation of ML into deployed information systems to improve public services.
  - For use by local government, NGOs, etc.

Defining ML4D as a field of study

(De Arteaga, Herlands, Neill, and Dubrawski, 2018, in press)

Problem	Solution
1. Applications and data geographically constrained to developing countries. 2. Problem concerns a critical development area for the region of interest.	4. Solution substantially uses ML as an integral element.
3. Problem and/or contextual elements are such that existing or plausible solutions in developed regions are not viable, and proposed solution effectively addresses these differences.	

The specific challenges of development problems (and data) may require us to use existing ML techniques in novel ways or to develop new ML methodology.

ML4D application examples

Broad areas: social, economic, environmental, institutional, health

McBride and Nichols (2015) perform **poverty targeting** using "proxy means tests" (i.e., prediction from easily observable household characteristics). They show that **random forests** outperform linear regression on data from USAID Poverty Assessment Tools.

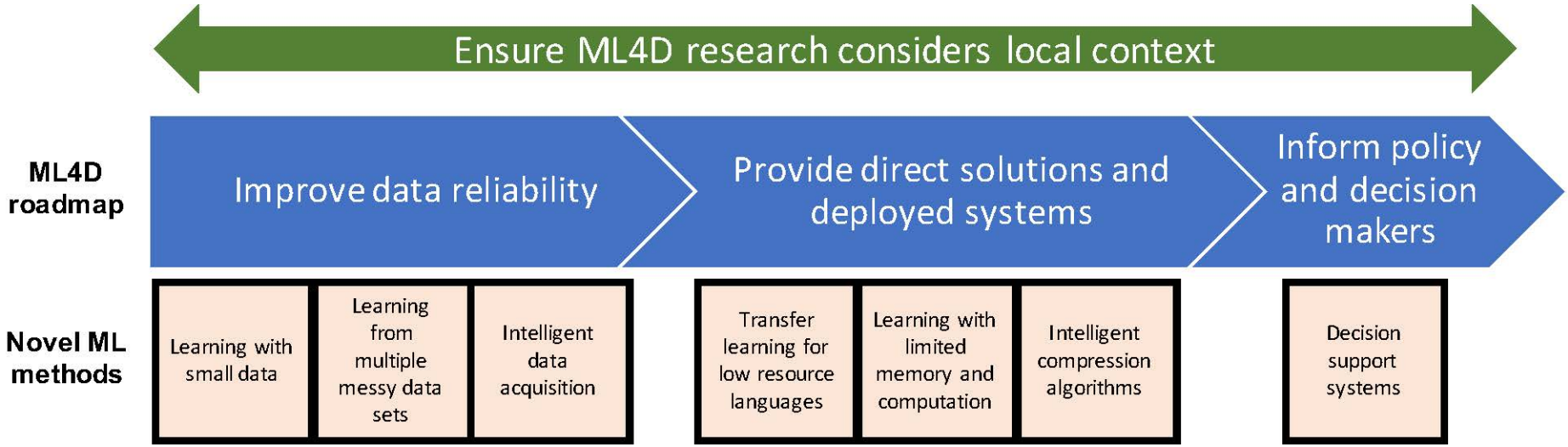
Knippenberg, Jensen, and Constas (2018) predict **food insecurity** at the household level using random forests and penalized regression.

Tien Bui et al. (2012) predict **landslide susceptibility** using decision trees, naïve Bayes, and support vector machine classifiers.

Mwebaze et al. (2010) learn causal Bayesian networks for **famine prediction** in Uganda, while Okori and Obua (2011) use support vector machines, k-nearest neighbors, naïve Bayes, and decision trees.

A roadmap for ML4D

(De Arteaga, Herlands, Neill, and Dubrawski, 2018, in press)

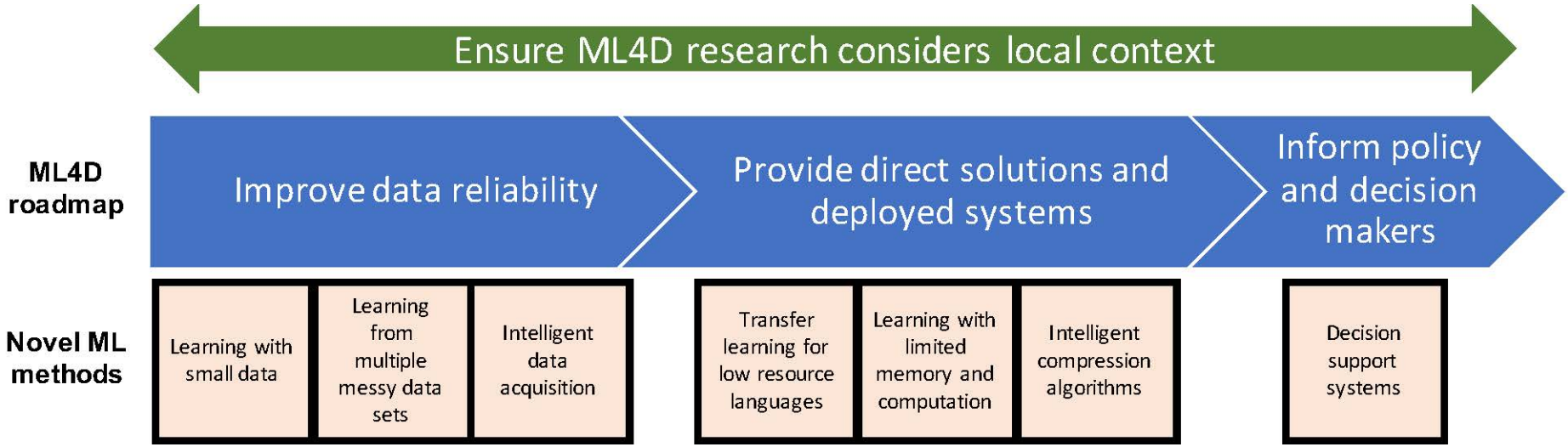


Common features and challenges across many development problems suggest the value of ML solutions, including difficult data, uncertain outcomes, many confounding variables, and costliness of data collection.

"Difficult" data could be biased, incomplete, noisy, or otherwise "messy"; either overly massive or insufficient; unstructured (text, images) or have complex, heterogeneous network structure (e.g., online social media).

A roadmap for ML4D

(De Arteaga, Herlands, Neill, and Dubrawski, 2018, in press)



Common features and challenges across many development problems suggest the value of ML solutions, including difficult data, uncertain outcomes, many confounding variables, and costliness of data collection.

Cleaning (structured) data

Structuring unstructured data

Prioritizing data for collection

What data might be available?

- **Survey data**- typically historical rather than current, coarse spatial resolution, often incomplete, noisy, and suffering from sampling biases.
- **Satellite imagery**- climate data, wildfires, land use, urbanization, access to electric lighting, migratory or displaced populations.
- **Cell phone** data of various types:
  - Calls and SMS text messages.
  - Location, movement, and interaction data (GPS, Twitter) → behavior patterns, event detection, social structure…
  - "Financial" data: cell-phones for mobile banking (MPESA); more widely, cell-phone airtime used as informal currency.
- **Internet data**- penetration much lower; usage patterns different, e.g. public kiosks; low but rapidly increasing smart phone penetration.
- **Crowdsourced data** (e.g. txteagle, collaborative sensing).
- Drop-in **sensor networks** (e.g. for water quality monitoring).

Some practical challenges

- Poverty (individual, business, government)- even “inexpensive” solutions may be impossible w/o subsidies.
- Illiteracy, lack of education/training → user interface challenges.
- Diversity of languages → need for machine (or human) translation.
- Lack of power and communication infrastructure: low Internet penetration; frequent outages of electricity and connectivity require specialized solutions; rapidly increasing use of cell phones.
- Migratory/transitory populations; weak transportation infrastructure.
- Corrupt government, misuse of funds, low rule of law.
- Cultural differences, racial/religious/tribal conflicts, mistrust of authorities, outsiders, and top-down solutions.
- Challenges of field research: remote locations, security concerns, need to partner with local governments/NGOs.
- Low amount and quality of collected data → robust analyses needed.
- Challenges of measuring and sustaining impact of new technologies.

Some practical challenges

- Poverty (individual, business, [illegible] "inexpensive" solutions may [illegible] ssible
- Illiteracy, lack o[illegible]
- Diversity of [illegible]
- Lack [illegible] net penetrat[illegible] equire specialize[illegible]
- [illegible] structure.
- Cont[illegible]
- Cultural [illegible], mistrust of auth[illegible] utions.
- Challenges of [illegible] emote [illegible] cations, security concerns, need to partner with [illegible] governments/NGOs.
- Low <u>amount</u> a[illegible] <u>quality</u> of collected data → robust analyses needed.
- Challenges of <u>measuring</u> and <u>sustaining</u> impact of new technologies.

We need to keep all of these challenges in mind, as they can greatly impact our ability to collect data and to develop and deploy practical information systems.

But for today, let's focus on a few challenges which can be readily addressed using ML techniques.

Challenge #1: Low data quality

- Missing data (no values for some record-attribute pairs)
  - Data can be missing completely at random (easiest), missing at random, or missing not at random (hardest).
  - Need different approaches in each case!
- Noisy data (incorrect/altered values for some record-attribute pairs)
  - Easiest case: i.i.d., Gaussian, additive noise
  - Often not true in practice: dependent errors, anomalous values, noise distribution unknown (must be inferred).
- Systematic biases in data (known? unknown? how to infer?)
  - Convenience sampling, selection bias, reporting bias, false info.
- Different sources report different, often conflicting data
  - Each source has its own limitations/biases, how to integrate information and obtain an accurate “big picture”?
  - Extreme case- crowdsourcing!

Challenge #1: Low data quality

- Much work in ML has considered data quality issues:
  - Inferring missing values (and incorporating the uncertainty about these values into predictions and other results).
  - Resampling the data to account for known distributional biases.
  - Evaluating robustness of different learning methods to noise.
  - Detecting, examining, and correcting anomalous values.
- Expectation-maximization (EM) is commonly used:
  - Iterate between E) inferring missing values (or correcting noisy values) given the current solution, and M) using the inferred values to improve the solution.
- Dealing with (unknown) sample selection bias for prediction:
  - “Covariate shift” methods: mapping from input to output is identical for training and test data, but distribution of inputs different.
  - “Transfer learning” methods: different (but related) mappings from input to output for training and test data. (very hard problem!)

Challenge #2: Which data to collect?

- Data collection in the developing world is often difficult and expensive, thanks to logistical challenges and lack of existing infrastructure.
- Available data may be **sparse** or **nonexistent**:
  - Need ML methods that can deal with low quantity of data (e.g. by incorporating models, priors, distributional assumptions)…
  - … and/or clever workarounds (e.g. use of non-traditional data sources which can be more easily collected).
- Typical problem: which data to collect given limited resources and costliness of acquisition?
  - **Active learning** problem: given unlabeled data, which points should we ask an oracle to label?
  - Goal: maximum classification accuracy with minimum query cost.
  - Many approaches (e.g. query points with maximum uncertainty or near the decision boundary, choose diverse set of points, etc).

Other ML solutions for development

- Modeling the relationships between many variables:
  - Bayesian networks and other graphical models to learn conditional dependencies.
  - Predictive analysis (e.g. identification of “leading indicators”).
  - Variable selection and variable importance: which variables are most relevant, and which are irrelevant?
  - Causal structure learning to predict the effects of interventions.
- Detection of emerging events and patterns:
  - Outbreak detection, disaster early warning systems, detection or prediction of conflict, regime change, crime, corruption, etc…
- Optimization under constraints:
  - Location planning for shared resources (schools, clinics)
  - Informing individual decisions (e.g. agriculture- which crops to plant, where to sell them, how much to charge…)
  - Probabilistic modeling of risk at micro and macro level.

Today's lecture

Part 1 (9:30-11am):

- Motivation for using machine learning (ML) to analyze economic and development data.
- Overview of ML problem paradigms and commonly used methods; when to use each.

Part 2 (11:15am-12:30pm):

- Specific challenges and ML solutions for working with development data.
- Cutting edge ML methods and applications.

Structuring “Big Data”, Part 1: Learning from Satellite Imagery

Remote sensing of poverty

- Remote sensing with satellite imagery has been used to infer a variety of development data.
  - Urban land cover
  - Forest cover/deforestation
  - Air pollution
  - Resources (minerals, fish)
  - Flood/mudslide risk
  - Agricultural crops
  - Roads and travel
  - Building types/materials
  - Poverty (income, electricity use…)
- Total visible light emitted at night (“night lights”) is common as a proxy for local economic activity.
- Benefits as compared to poverty surveys: faster, cheaper, lower time lag (response to disasters, internal displacement, other emerging events).

Remote sensing of poverty

- Remote sensing with satellite [illegible] been used [illegible]
  - [illegible]
  - [illegible]

"Even one of the most rudimentary sources of satellite data– luminosity at night, with annual frequency, 1-km resolution, and just one useable band– has enabled us to answer important questions in new, convincing ways.

This only serves to underscore the great potential that lies ahead for the use of remote sensing in economics."

(Donaldson and Storeygard, 2016)

[illegible] is [illegible]y.

- Benef[illegible] surveys: faster, cheaper, lower time [illegible]nse to disasters, internal displacement, other emerging events).

Combining satellite imagery and ML to predict poverty (Jean, Burke, et al., 2016)

- Night lights have some major limitations:
  - In impoverished areas, nighttime luminosity levels are low and show little variation → can't distinguish between below and 2-3x the extreme poverty line ($1.90 per day).
  - Trouble distinguishing between poor, densely populated and rich, sparsely populated areas.
- Other data sources such as cell phone call data records (CDR), while promising, rely on disparate, proprietary data, making them difficult to scale.
  - Nevertheless, some nice work along these lines (Steele et al., 2017) combining CDR with relevant covariates from night lights to predict poverty in Bangladesh.

Combining satellite imagery and ML to predict poverty (Jean, Burke, et al., 2016)

Solution:

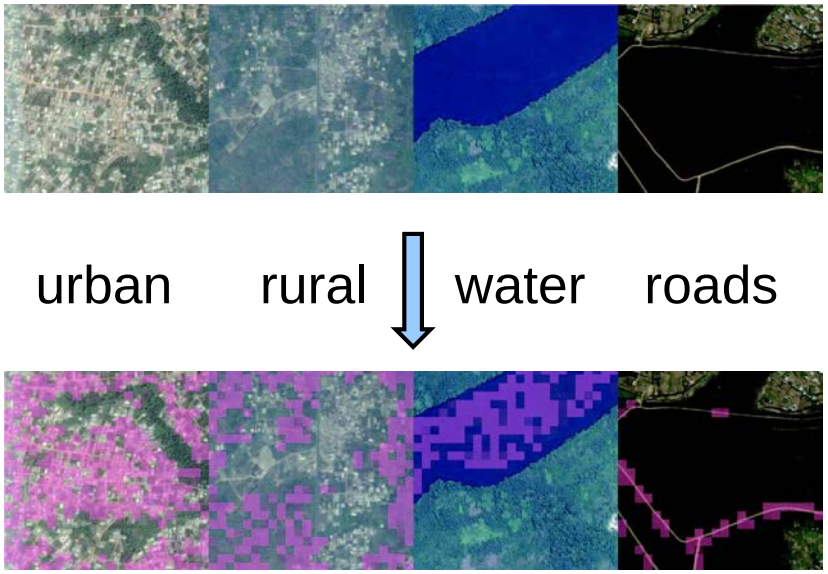
- Combine night lights with high-resolution **daytime** satellite imagery (can capture other indicators such as roads, roofing materials, etc.)
- Train ML models using ground truth from surveys; use for prediction.
- **Deep learning** with convolutional neural networks (CNN): best known approach for extracting features and predicting from raw image data.
- But requires lots of labeled training data, and survey data is scarce!

- Brilliant idea by Jean et al.: a multi-step "transfer learning" approach
  - Start by training a CNN on ImageNet (huge, public image classification dataset) → learns low-level image features (e.g., edges, corners) common to many vision tasks.
  - Fine-tune the CNN on a new task: predicting nighttime light intensities corresponding to daytime satellite imagery → learns mapping from raw image to relevant features (concise vector of numeric values per image).
  - Learn a regression model, using the structured data extracted from each raw image to predict cluster-level poverty survey results.

Combining satellite imagery and ML to predict poverty (Jean, Burke, et al., 2016)

Results:

- Dramatically outperforms nightlights for both assets & consumption
- Deep learning captures relevant features that generalize across countries
- Predictions from out-of-country models are almost as good as in-country



"The model is able to discern semantically meaningful features such as urban areas, roads, bodies of water, and agricultural areas, even though there is no direct supervision—that is, the model is told neither to look for such features, nor that they could be correlated with economic outcomes of interest. It learns on its own that these features are useful for estimating nighttime light intensities."

Other identified features: roofing materials, distance to urban centers, vegetation, agriculture, …

Structuring “Big Data”, Part 2: Event Detection and Prediction from Online Social Media

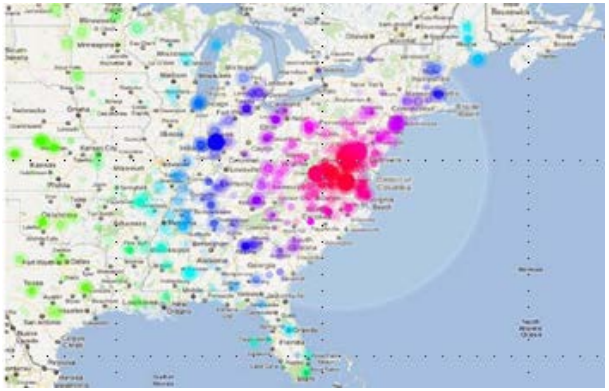
Event Detection from Social Media

(Chen and Neill, KDD 2014)

Protest in Mexico, 7/14/2012

2012 Washington D.C. Traffic

Tweet Map for 2011 VA Earthquake



Social media is a real-time “sensor” of large-scale population behavior, and can be used for early detection of emerging events...

… but it is very complex, noisy, and subject to biases.

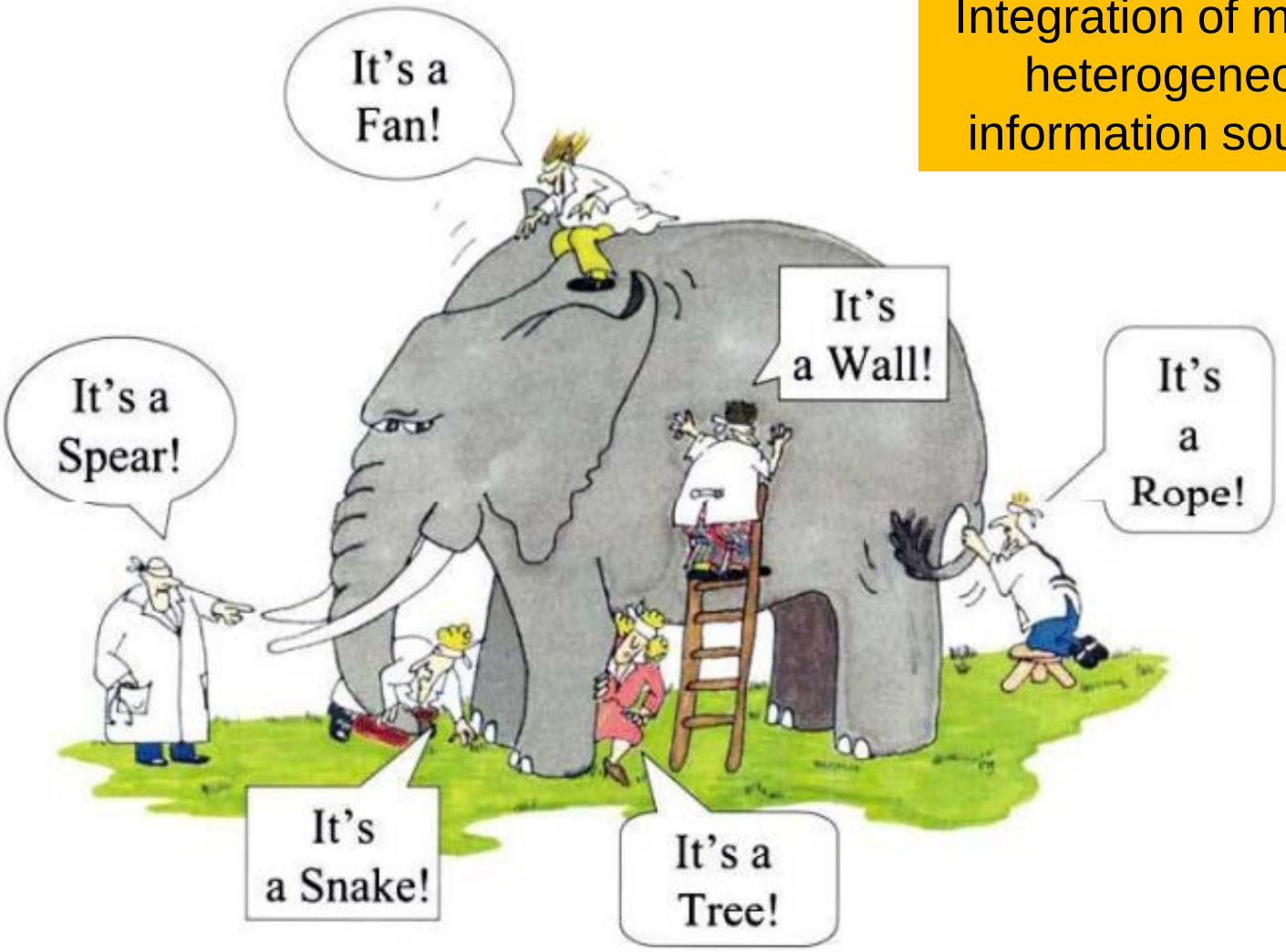
We have developed a new event detection methodology:

“Non-Parametric Heterogeneous Graph Scan” (NPHGS)

Applied to: civil unrest prediction, rare disease outbreak detection, and early detection of human rights events.

Technical Challenges

Integration of multiple heterogeneous information sources!



Technical Challenges

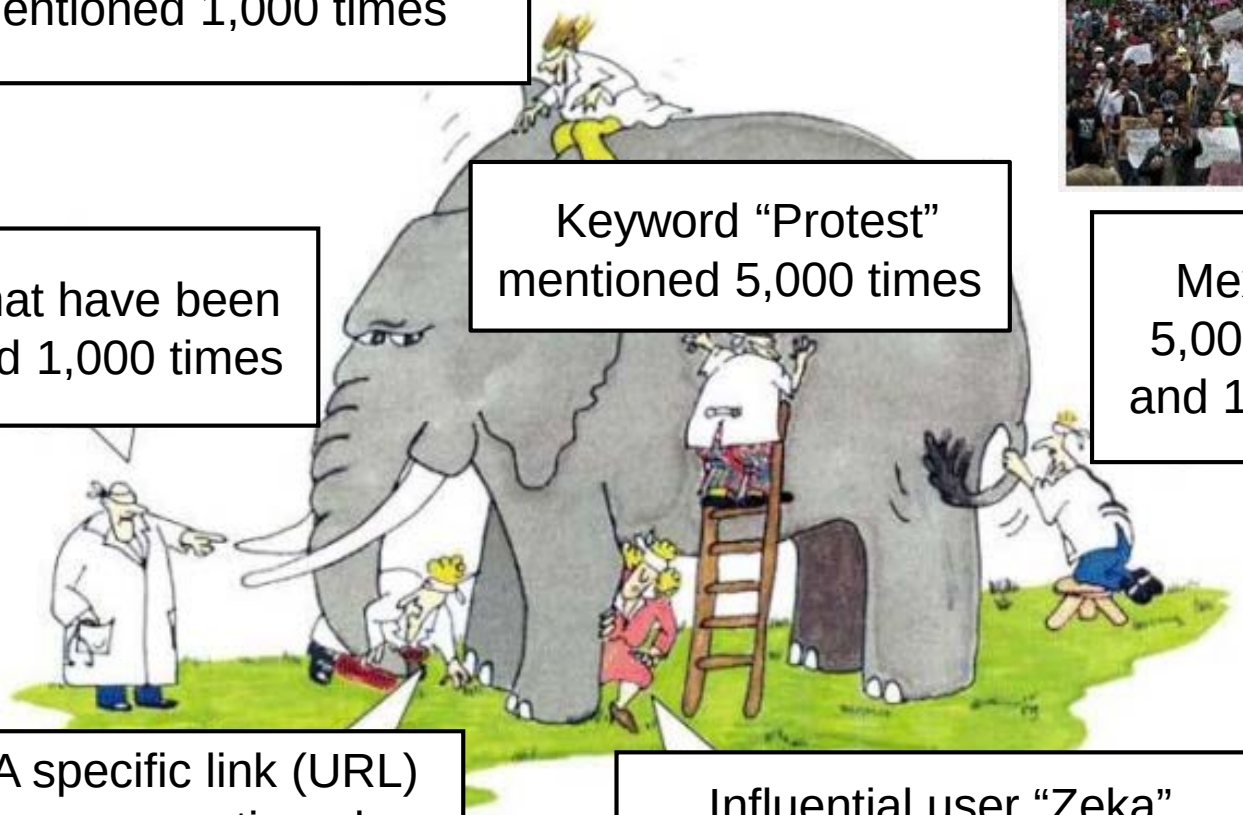
One week before Mexico's 2012 presidential election:

Hashtag "#Megamarch" mentioned 1,000 times

Keyword "Protest" mentioned 5,000 times

Tweets that have been re-tweeted 1,000 times

Mexico City has 5,000 active users and 100,000 tweets



A specific link (URL) was mentioned 866 times

Influential user "Zeka" posted 10 tweets

Technical Challenges

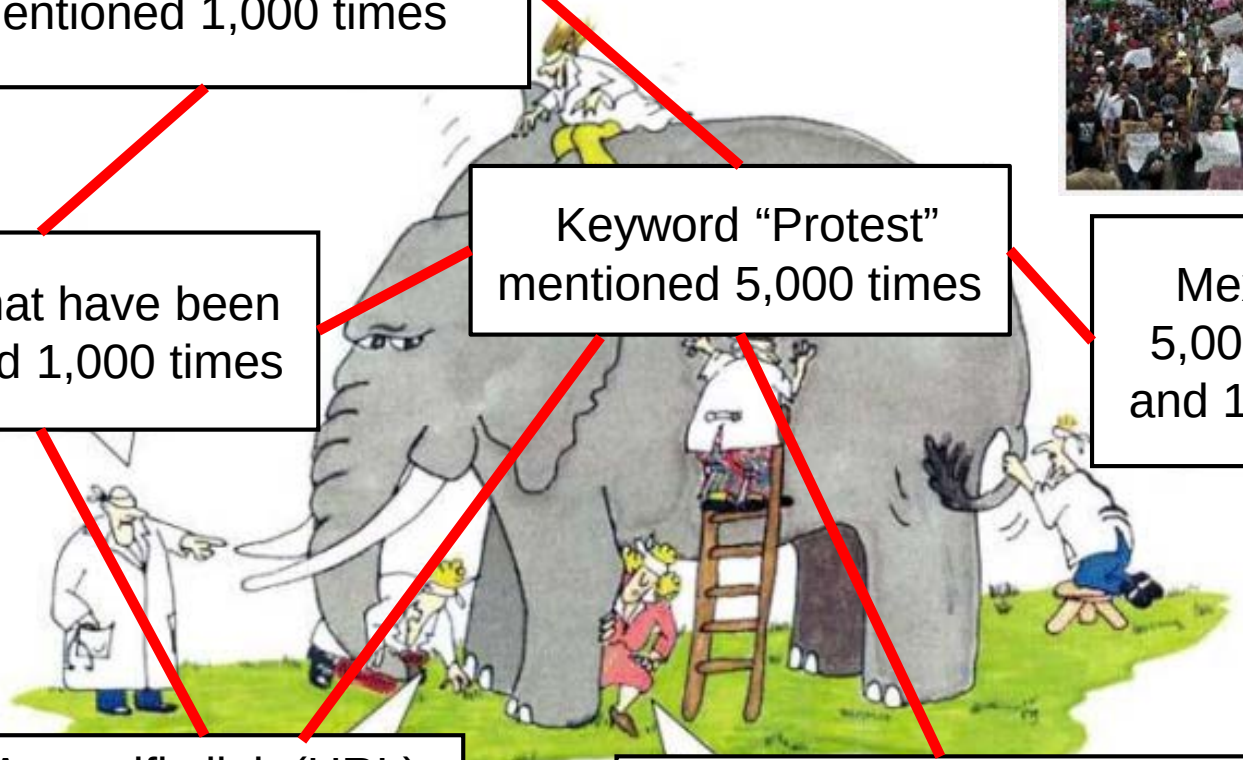
One week before Mexico's 2012 presidential election:

Hashtag "#Megamarch" mentioned 1,000 times

Keyword "Protest" mentioned 5,000 times

Tweets that have been re-tweeted 1,000 times

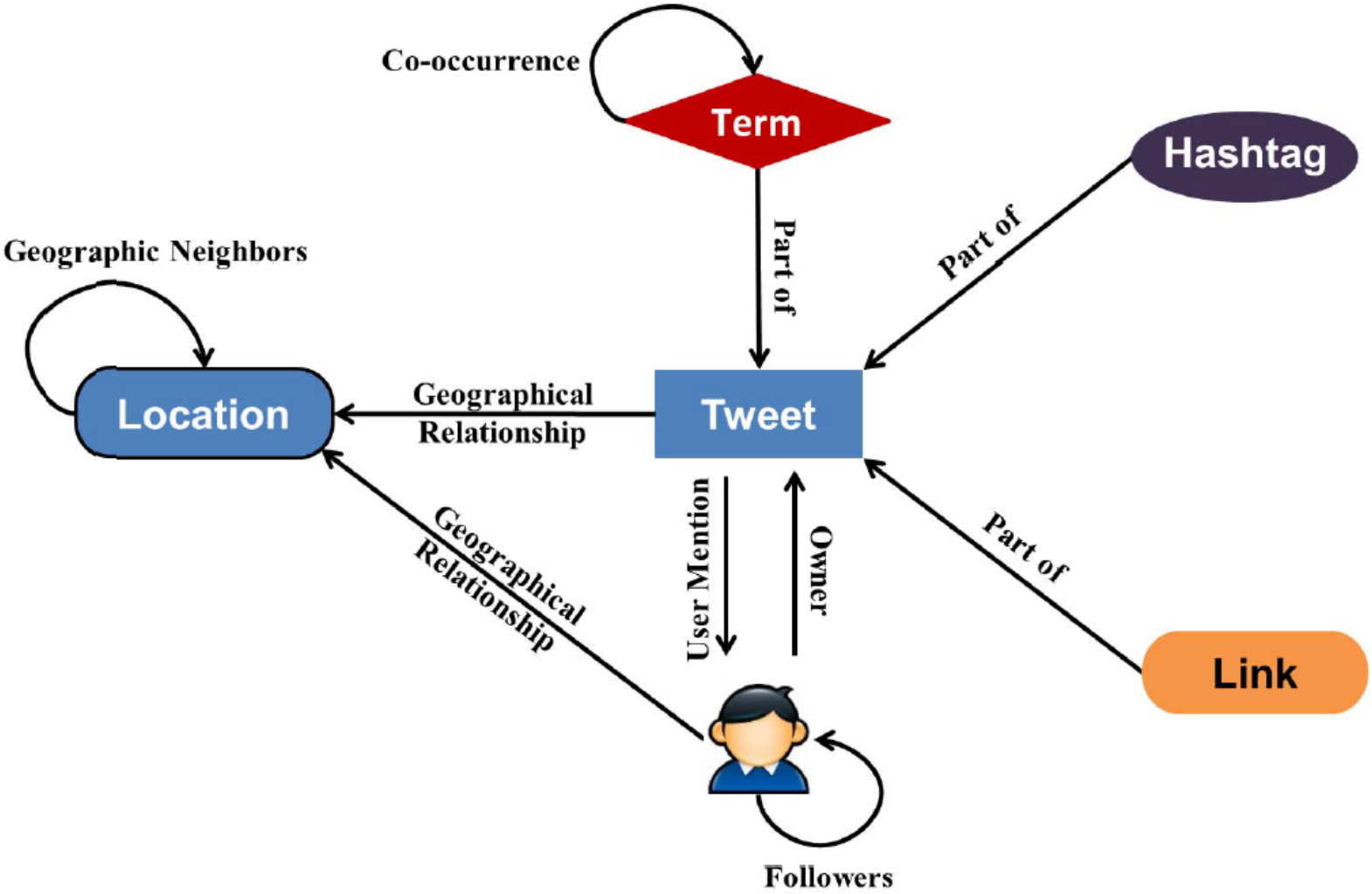
Mexico City has 5,000 active users and 100,000 tweets



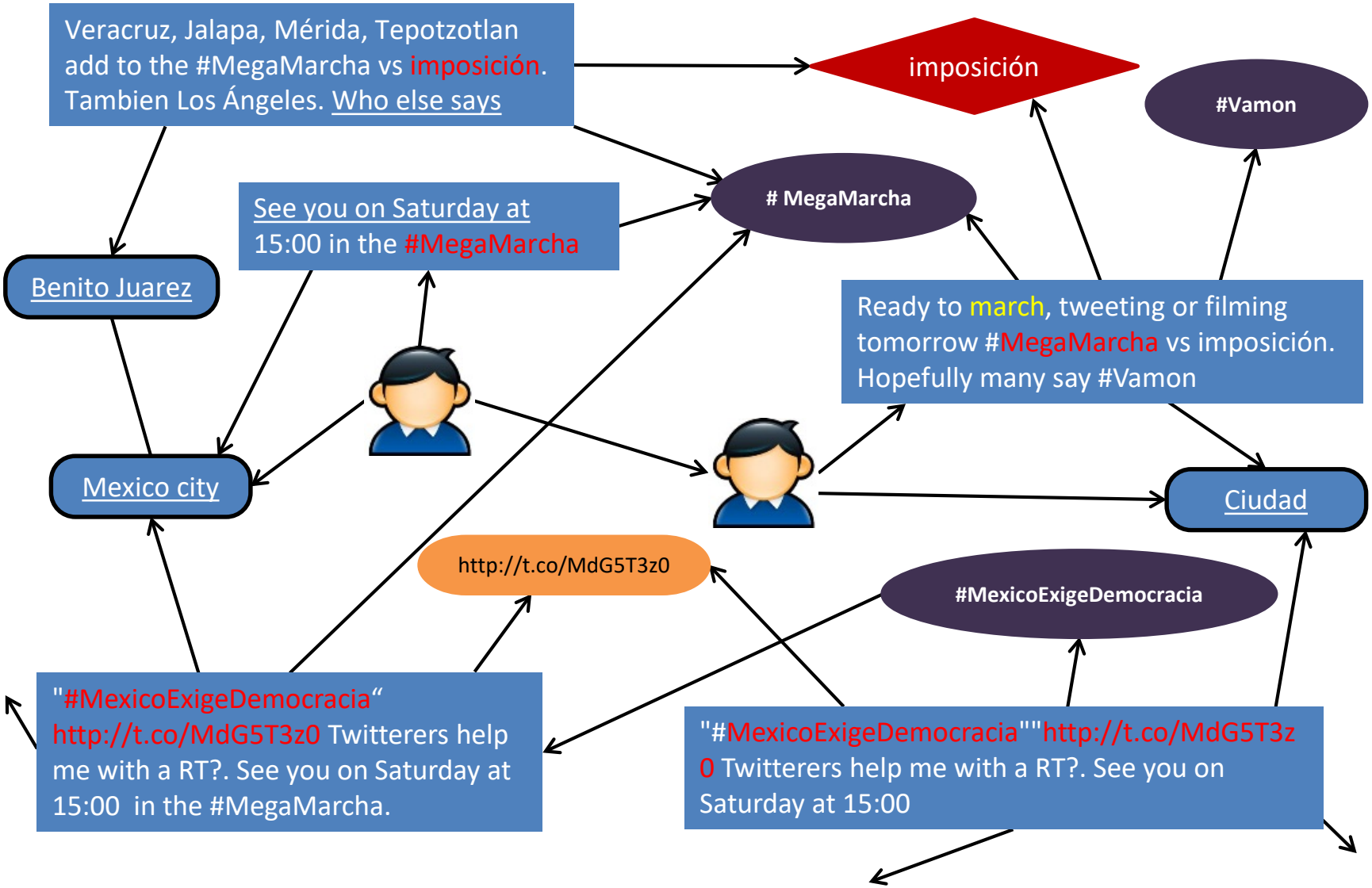
A specific link (URL) was mentioned 866 times

Influential user "Zeka" posted 10 tweets

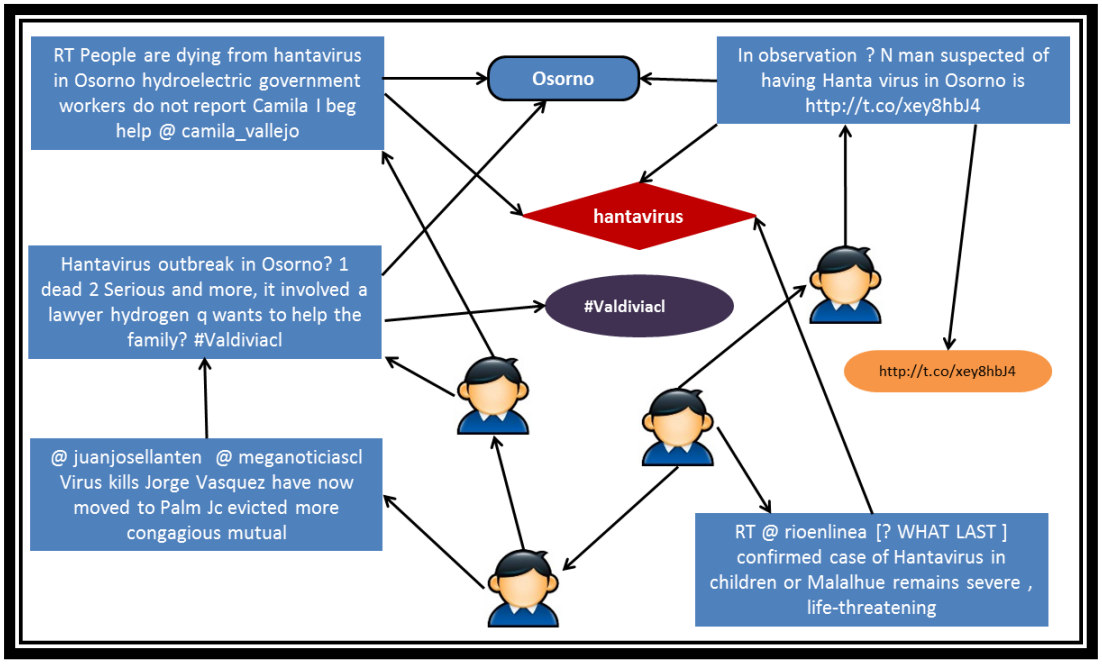
Twitter Heterogeneous Network

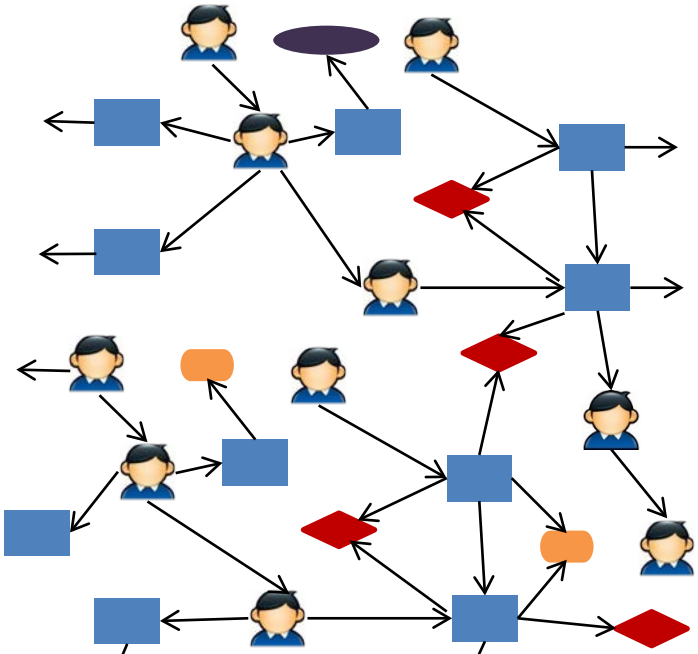


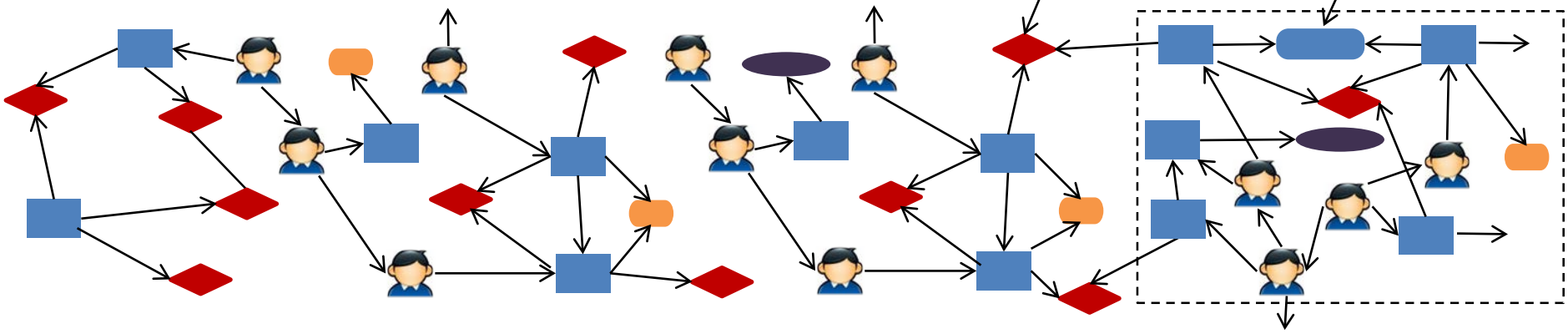
Twitter Heterogeneous Network



Twitter Heterogeneous Network







Nonparametric Heterogeneous Graph Scan

(Chen and Neill, KDD 2014)

1) We model the heterogeneous social network as a **sensor network**.

   Each node senses its local neighborhood, computes multiple features, and reports the overall degree of anomalousness.

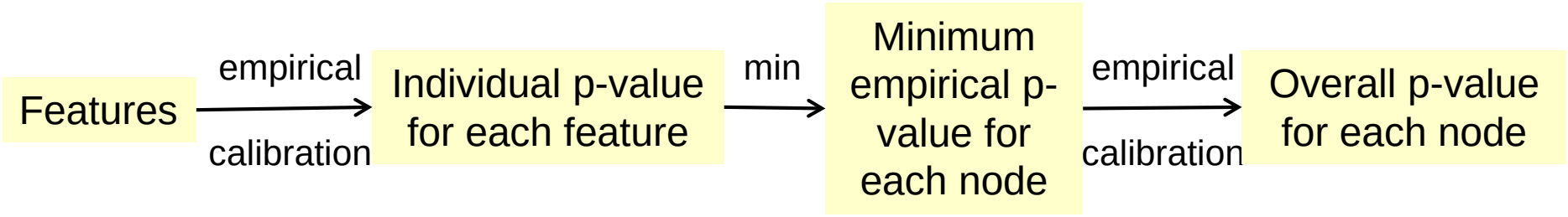
2) We compute an **empirical p-value** for each node:
   - Uniform on [0,1] under the null hypothesis of no events.
   - We search for subgraphs of the network with a higher than expected number of low (significant) empirical p-values.

3) We can scale up to very large heterogeneous networks:
   - Heuristic approach: **iterative subgraph expansion** ("greedy growth" to subset of neighbors on each iteration).
   - We can efficiently find the best subset of neighbors, ensuring that the subset remains connected, at each step.

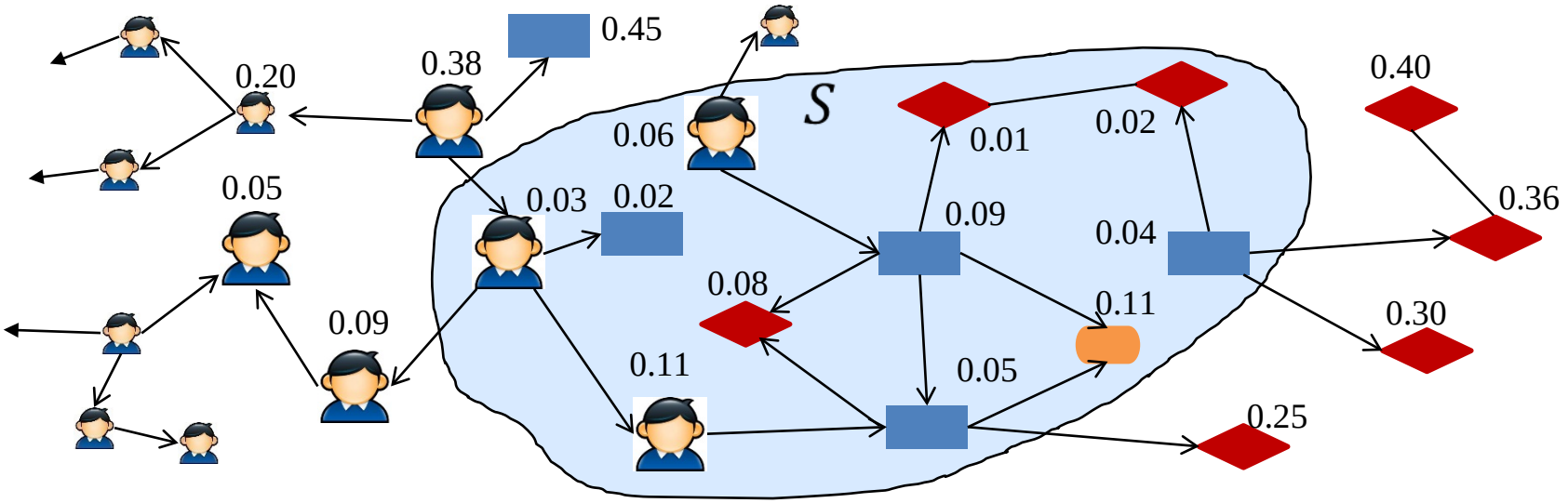
Sensor network modeling

Each node reports an empirical p-value measuring the current level of anomalousness for each time interval (hour or day).

Object Type	Features
User	# tweets, # retweets, # followers, #followees, #mentioned_by, #replied_by, diffusion graph depth, diffusion graph size
Tweet	Klout, sentiment, replied_by_graph_size, reply_graph_size, retweet_graph_size, retweet_graph_depth
City, State, Country	# tweets, # active users
Term	# tweets
Link	# tweets
Hashtag	# tweets



Nonparametric graph scanning



$$S^{\star} = \underset{S \in V:\, S \text{ is connected}}{\operatorname{argmax}} F(S)$$

We propose a fast, approximate algorithm which identifies the most anomalous subgraphs. A subgraph is anomalous if it has a higher-than-expected number of low (significant) p-values.

NPHGS evaluation- civil unrest

Country	# of tweets	News source*
Argentina	29,000,000	Clarín; La Nación; Infobae
Chile	14,000,000	La Tercera; Las Últimas Notícias; El Mercurio
Colombia	22,000,000	El Espectador; El Tiempo; El Colombiano
Ecuador	6,900,000	El Universo; El Comercio; Hoy

Gold standard dataset: 918 civil unrest events between July and December 2012.

Example of a gold standard event label:

PROVINCE = “El Loa” COUNTRY = “Chile”
DATE = “2012-05-18” LINK = “http://www.pressenza.com/2012/05/...”
DESCRIPTION = “A large-scale march was staged by inhabitants of the northern city of Calama, considered the mining capital of Chile, who demanded the allocation of more resources to copper mining cities”

We compared the detection performance of our NPHGS approach to homogeneous graph scan methods and to a variety of state-of-the-art methods previously proposed for Twitter event detection.

NPHGS results- civil unrest

Method	FPR (FP/Day)	TPR (Forecasting)	TPR (Forecasting & Detection)	Lead Time (Days)	Lag Time (Days)	Run Time (Hours)
ST Burst Detection	0.65	0.07	0.42	1.10	4.57	30.1
Graph Partition	0.29	0.03	0.15	0.59	6.13	18.9
Earthquake	0.04	0.06	0.17	0.49	5.95	18.9
RW Event	0.10	0.22	0.25	0.93	5.83	16.3
Geo Topic Modeling	0.09	0.06	0.08	0.01	6.94	9.7
NPHGS (FPR=.05)	0.05	0.15	0.23	0.65	5.65	38.4
NPHGS (FPR=.10)	0.10	0.31	0.38	1.94	4.49	38.4
NPHGS (FPR= .15)	0.15	0.37	0.42	2.28	4.17	38.4
NPHGS (FPR=.20)	0.20	0.39	0.46	2.36	3.98	38.4

Table 3: Comparison between NPHGS and Existing Methods on the civil unrest datasets

NPHGS outperforms existing representative techniques for both event detection and forecasting, increasing **detection power**, **forecasting accuracy**, and **forecasting lead time** while reducing **time to detection**.

Similar improvements in performance were observed on a second task:

Early detection of rare disease outbreaks, using gold standard data about 17 hantavirus outbreaks from the Chilean Ministry of Health.

Detected Hantavirus outbreak, 10 Jan 2013

First news report: 11 Jan 2013

Paraguay

Uruguay

Temuco and Villarrica, Chile

Locations
Users
Keywords
Hashtags
Links
Videos

NPHGS results- human rights

We performed an exploratory analysis of human rights-related events in Mexico from January 2013 to June 2014, using Twitter data (10% sample, filtered using relevant keywords).

The top 50 identified clusters over the entire study period were analyzed manually to identify:

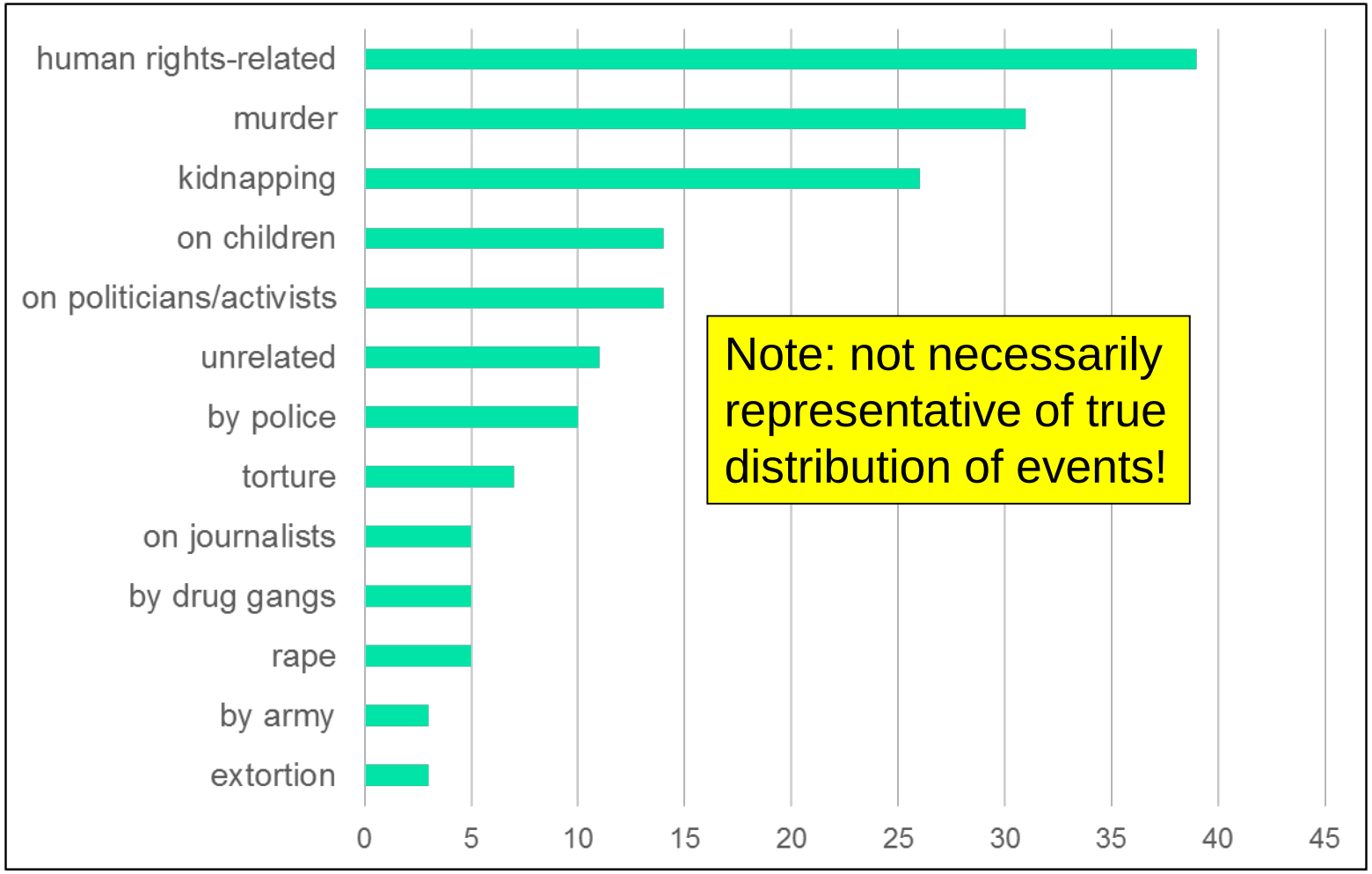
(1) whether the cluster was human rights related
(2) the types of human rights violations
(3) the victims of the violations
(4) the alleged perpetrators.

NPHGS was able to identify some human rights events of interest before international news sources…

… and in some cases, before local news sources.

Cluster characteristics

(top-50 detected clusters)



Questions and Open Discussion

References & suggested reading

- Chen F, Neill DB. Human rights event detection from heterogeneous social media graphs. *Big Data* 3(1): 34-40, 2015.
- Chen F, Neill DB. Non-parametric scan statistics for event detection and forecasting in heterogeneous social media graphs. *Proc. 20th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining*, 1166-1175, 2014.
- De Arteaga M, Herlands W, Neill DB, Dubrawski A. Machine learning for the developing world. *ACM Transactions on Management Information Systems*, 2018, in press.
- Donaldson D, Storeygard A. The view from above: applications of satellite data in economics. *Journal of Economic Perspectives* 30(4): 171-198, 2016.
- Einav L, Levin J. The data revolution and economic analysis. In *Innovation Policy and the Economy* Vol. 14: 1-24, 2014.
- Flaxman SR, Wilson, AG, Neill DB, Nikisch H, Smola AJ. Fast Kronecker inference in Gaussian processes with non-Gaussian likelihoods. *Proc. 32nd International Conference on Machine Learning, JMLR: W&CP* 37, 2015.
- Jean N, Burke M, Xie M, Davis WM, Lobell DB, Ermon S. Combining satellite imagery and machine learning to predict poverty. *Science* 353(6301): 790-794, 2016.
- Mullainathan S, Spiess J. (2017). Machine Learning: An Applied Econometric Approach. *Journal of Economic Perspectives* 31(2): 87-106, 2017.
- Steele JE, Sundsoy PR, Pezzulo C, *et al.* Mapping poverty using mobile phone and satellite data. *Journal of the Royal Society Interface* 14: 20160690, 2017.
- Varian HR. Big data: new tricks for econometrics. *Journal of Economic Perspectives* 28(2): 3-28, 2014.
- The proceedings of the recent Workshop on Machine Learning for Development gives many additional examples of promising emerging work in the field. https://sites.google.com/site/ml4development/

Thanks for listening!

More details on our web site:
http://epdlab.heinz.cmu.edu

Or e-mail me at both addresses:
neill@cs.cmu.edu
daniel.neill@nyu.edu