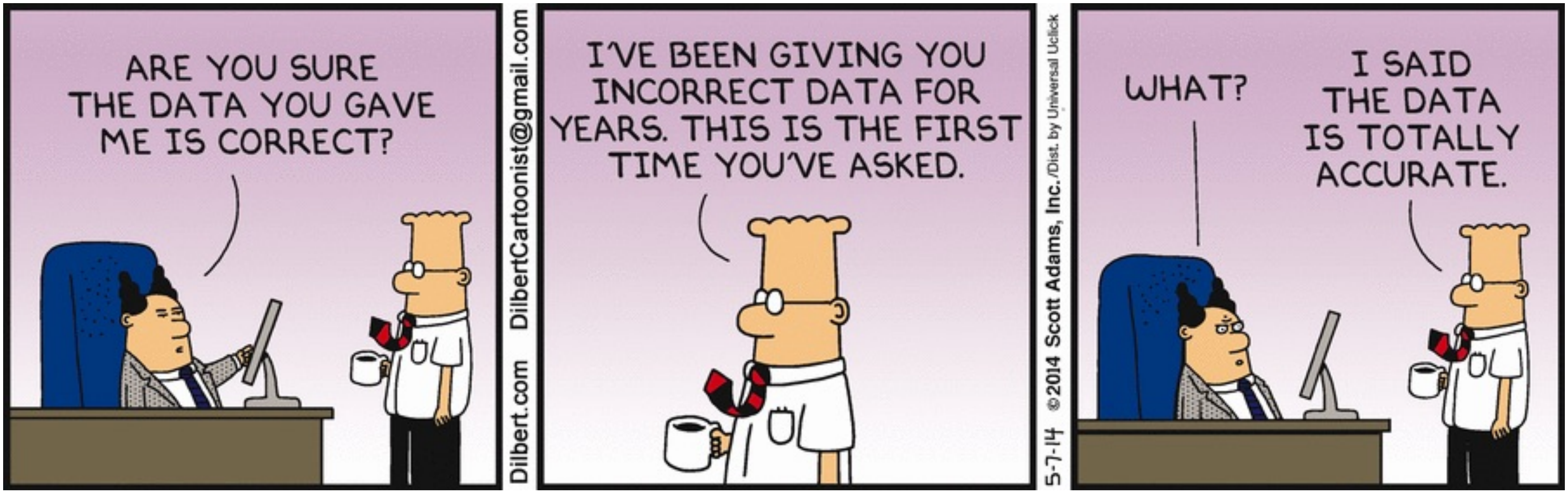
SLiMFast: Guaranteed Results for Data Fusion and Source Reliability

Theodoros Rekatsinas
@thodrek

joint work with
Manas Joglekar, Hector Garcia-Molina,
Aditya Parameswaran, and Christopher Ré

Reliable data = value

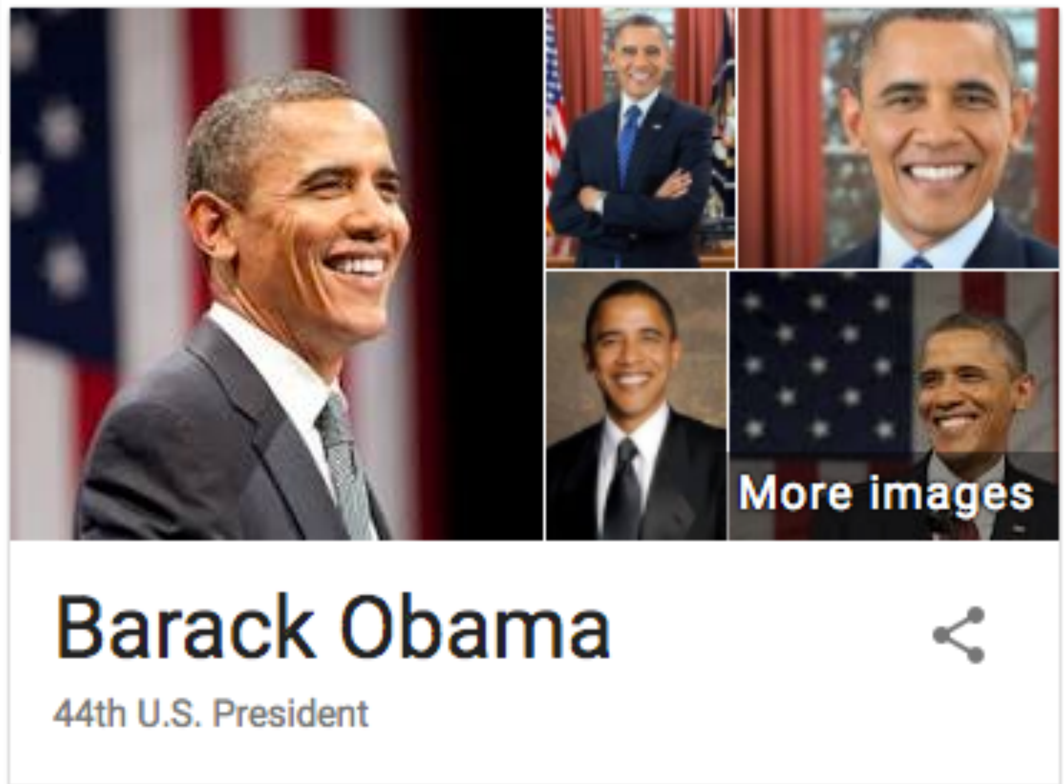


Today's take-away:

How to detect inaccurate data and hoax sources

Examples of inaccurate data: information extraction

Examples of inaccurate data: information extraction



Examples of inaccurate data: human annotations

"Is it a Dog or a Wolf?"

amazon mechanical turk

Examples of inaccurate data: alternative facts

Today’s Agenda

Data Fusion: A quick recap

SLiMFast: Use features to describe sources

SLiMFast’s Optimizer: Don’t worry about ML algorithms

Data fusion

We want to find the true value of noisy facts

"Ok Google, is Obama a king or a president?"

United States of America / King

Barack Obama

Barack Obama

44th U.S. President

Data fusion

We want to find the true value of noisy facts

"Ok Google, is Obama a king or a president?"

United States of America / King

Barack Obama

Barack Obama

44th U.S. President

Where does data fusion come up?

Knowledge base construction

"Is it a Dog or a Wolf?"

amazon mechanical turk

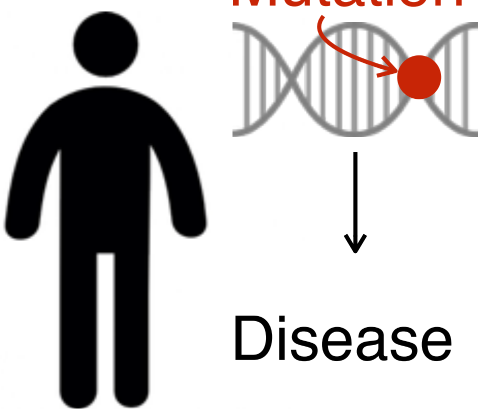
Crowdsourcing

Social sensing

Example: personalized medicine

Disease → Gene variant ?

Mutation



Gene

Disease

Knowledge Base Construction (KBC)

DeepDive

25 million articles

Goal: A disease-gene knowledge base to advance personalized medicine

Problems in knowledge base construction

Extractions

Source	Disease	Gene	CausedBy

Genetic Heterogeneity of Li-Fraumeni Syndrome

A second form of Li-Fraumeni syndrome (LFS2; 609265) is caused by mutation in the CHEK2 gene (604373), and an LFS locus (LFS3; 609266) has been mapped to chromosome 1q23.

Source: OMIM

Problems in knowledge base construction

Extractions

Source	Disease	Gene	CausedBy
OMIM	Li-Fraumeni Syndrome	CHEK2	**Yes**

Genetic Heterogeneity of Li-Fraumeni Syndrome

A second form of $\boxed{\text{Li-Fraumeni syndrome}}$ (LFS2; 609265) $\boxed{\text{is caused by}}$ mutation in the $\boxed{\text{CHEK2 gene}}$ (604373), and an LFS locus (LFS3; 609266) has been mapped to chromosome 1q23.

Source: OMIM

Problems in knowledge base construction

Extractions

Source	Disease	Gene	CausedBy
OMIM	Li-Fraumeni Syndrome	CHEK2	**Yes**
Paper	Li-Fraumeni Syndrome	CHEK2	**No**

Genetic Heterogeneity of Li-Fraumeni Syndrome

A second form of Li-Fraumeni syndrome (LFS2; 609265) is caused by mutation in the CHEK2 gene (604373), and an LFS locus (LFS3; 609266) has been mapped to chromosome 1q23.

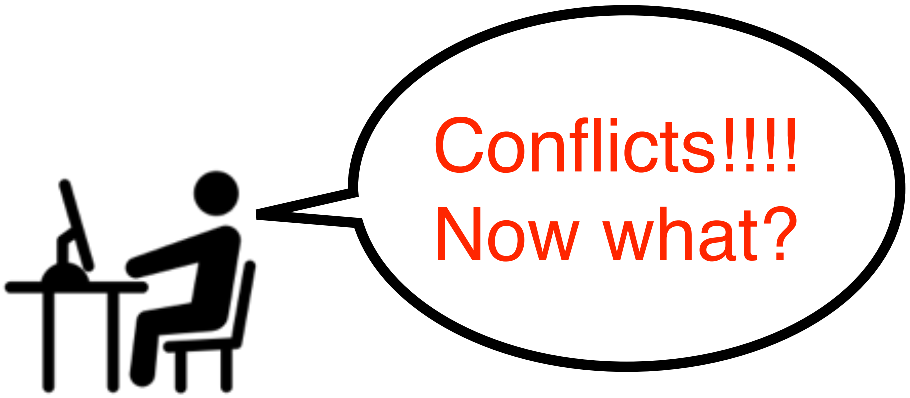
Source: OMIM

Increasing evidence that germline mutations in *CHEK2* do not cause Li-Fraumeni syndrome[†]

Nayanta Sodha, Richard S. Houlston, Sarah Bullock, Martin A. Yuille, Carol Chu, Gwen Turner, Rosalind A. Eeles

First published: 19 November 2002 Full publication history

Problems in knowledge base construction



Extractions

Source	Disease	Gene	CausedBy
OMIM	Li-Fraumeni Syndrome	CHEK2	**Yes**
Paper	Li-Fraumeni Syndrome	CHEK2	**No**

Genetic Heterogeneity of Li-Fraumeni Syndrome

A second form of Li-Fraumeni syndrome (LFS2; 609265) is caused by mutation in the CHEK2 gene (604373), and an LFS locus (LFS3; 609266) has been mapped to chromosome 1q23.

Source: OMIM

Increasing evidence that germline mutations in *CHEK2* do not cause Li-Fraumeni syndrome†

Nayanta Sodha, Richard S. Houlston, Sarah Bullock, Martin A. Yuille, Carol Chu, Gwen Turner, Rosalind A. Eeles

First published: 19 November 2002 Full publication history

Basic data fusion setup

Source observations

Source	Disease	Gene	CausedBy
OMIM	Li-Fraumeni Syndrome	CHEK2	**Yes**
Paper	Li-Fraumeni Syndrome	CHEK2	**No**

Knowledge base

CausedBy

Disease	Gene

Basic data fusion setup

Source observations

Source	**Disease**	**Gene**	**CausedBy**
OMIM	Li-Fraumeni Syndrome	CHEK2	**Yes**
Paper	Li-Fraumeni Syndrome	CHEK2	**No**

Object

Knowledge base

CausedBy

Disease	**Gene**

Basic data fusion setup

Source observations

Source	Disease	Gene	CausedBy
OMIM	Li-Fraumeni Syndrome	CHEK2	**Yes**
Paper	Li-Fraumeni Syndrome	CHEK2	**No**

Object

Source reports a value for Object

Knowledge base

CausedBy

Disease	Gene

Basic data fusion setup

Source observations

Source	Disease	Gene	CausedBy
OMIM	Li-Fraumeni Syndrome	CHEK2	Yes
Paper	Li-Fraumeni Syndrome	CHEK2	No

Conflict

Object

Source reports a value for Object

Knowledge base

CausedBy

Disease	Gene

Basic data fusion setup

Source observations

Source	Disease	Gene	CausedBy
OMIM	Li-Fraumeni Syndrome	CHEK2	**Yes**
Paper	Li-Fraumeni Syndrome	CHEK2	**No**

Knowledge base

CausedBy

Disease	Gene
Li-Fraumeni Syndrome	CHEK2

?

Object

Conflict

Source reports a value for Object

Object's true value

Basic data fusion setup

Source observations

Source	Disease	Gene	CausedBy
OMIM	Li-Fraumeni Syndrome	CHEK2	Yes
Paper	Li-Fraumeni Syndrome	CHEK2	No

Object

Conflict

Source reports a value for Object

Knowledge base

CausedBy

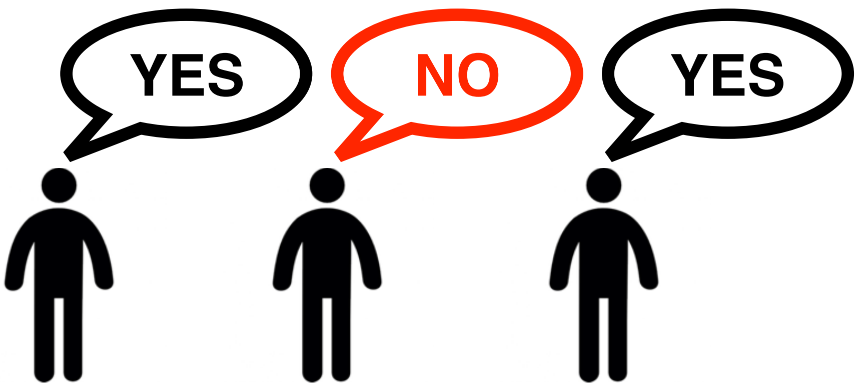
Disease	Gene
Li-Fraumeni Syndrome	CHEK2

?

Object's true value

How can we find the true value for each object?

Existing solutions to data fusion



Majority voting

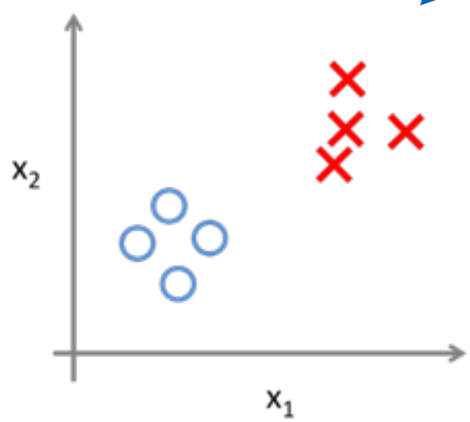
$$\text{Posterior} = \frac{\text{Likelihood} * \text{Prior}}{\text{Evidence}}$$

Probabilistic models

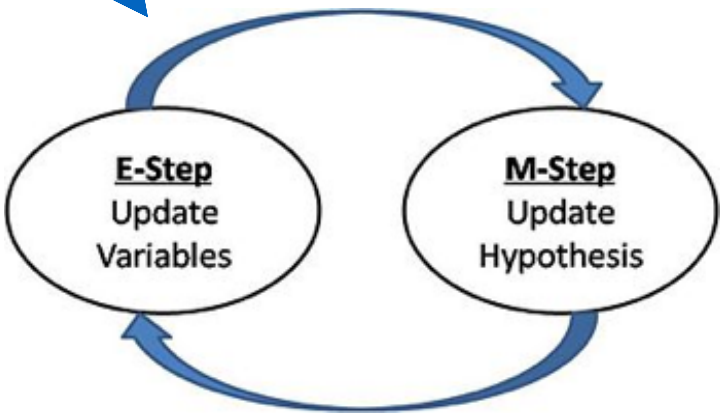
Existing solutions to data fusion

$$\text{Posterior} = \frac{\text{Likelihood} * \text{Prior}}{\text{Evidence}}$$

Probabilistic models

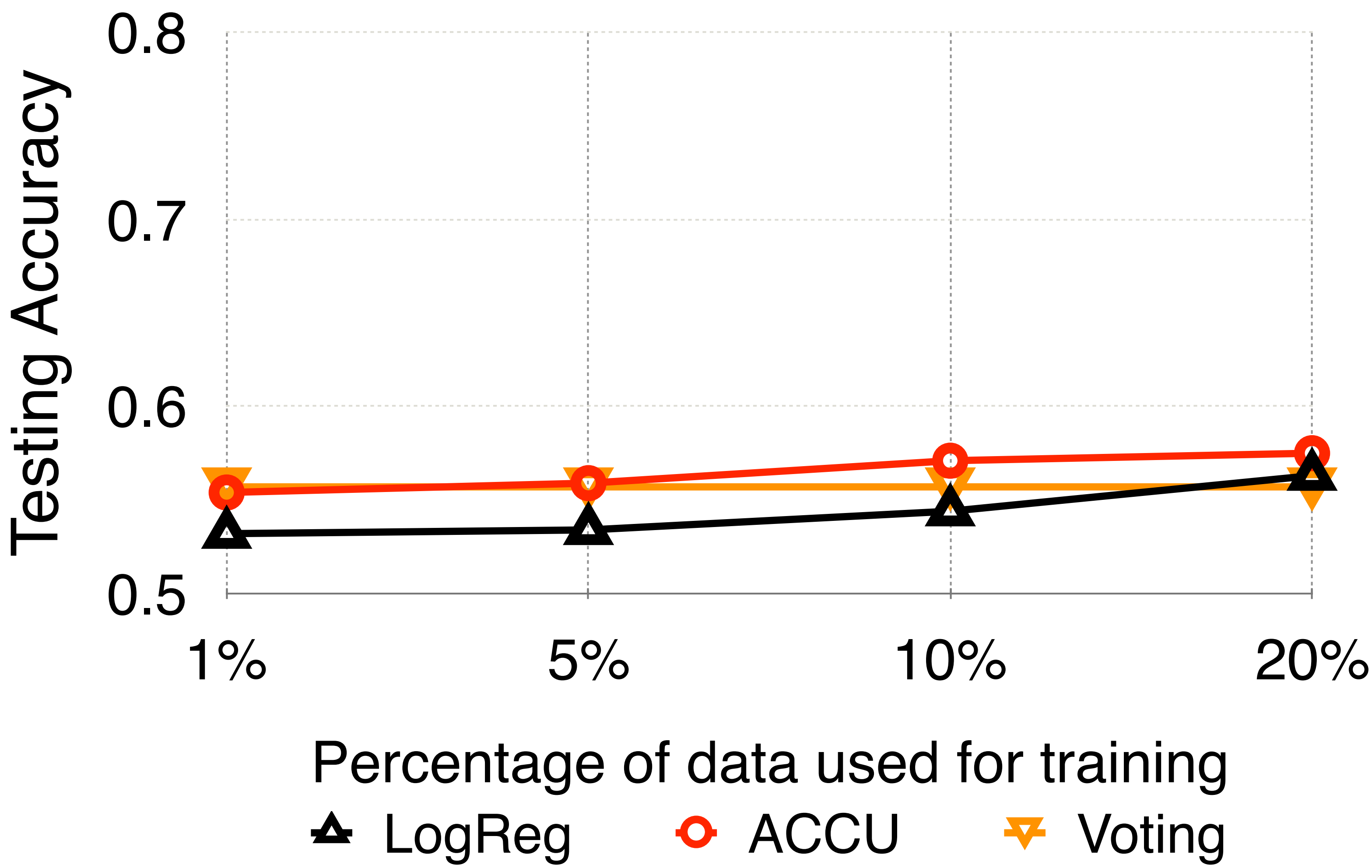


Supervised



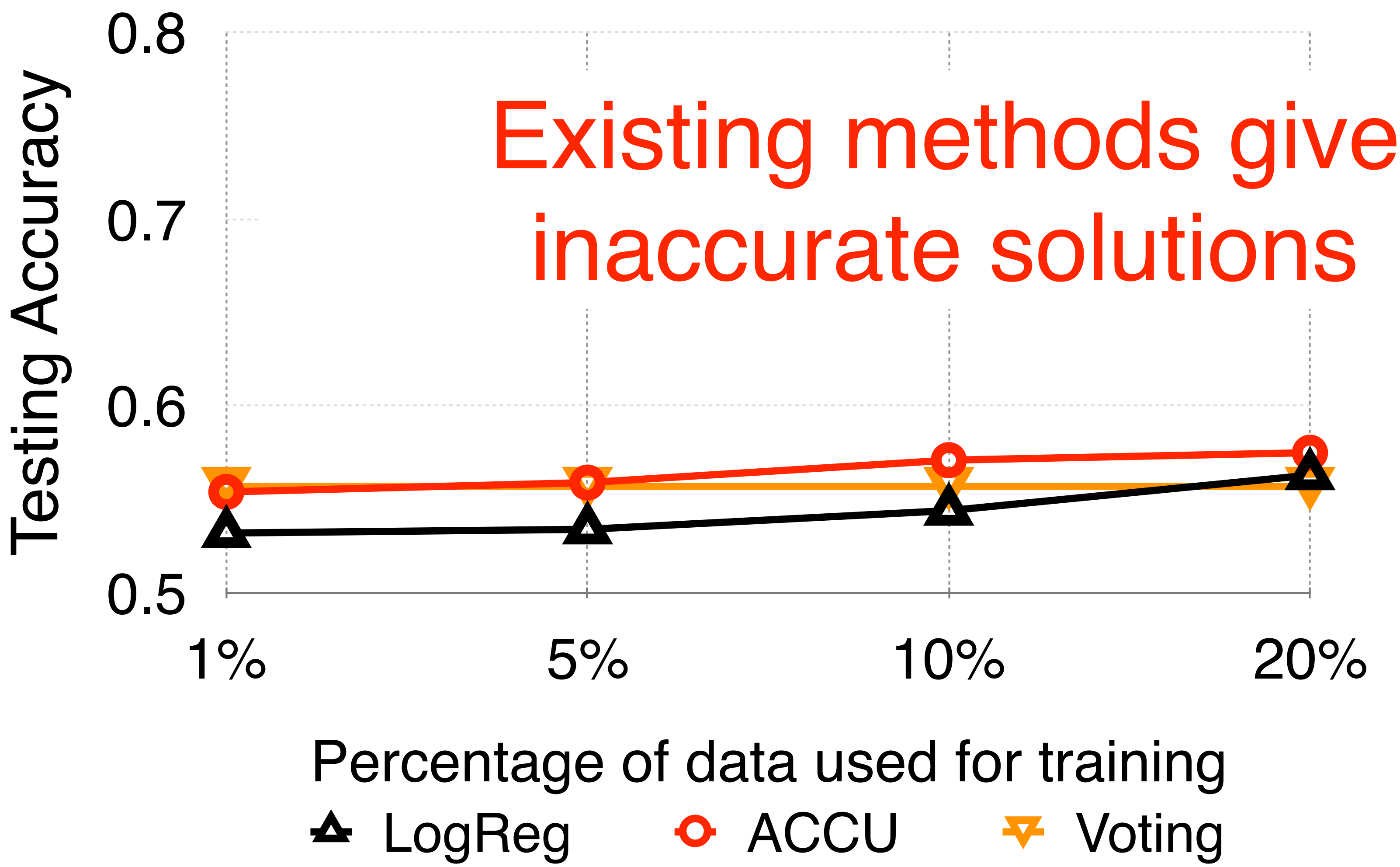
Un(semi-)supervised

Estimating the unknown true value for objects



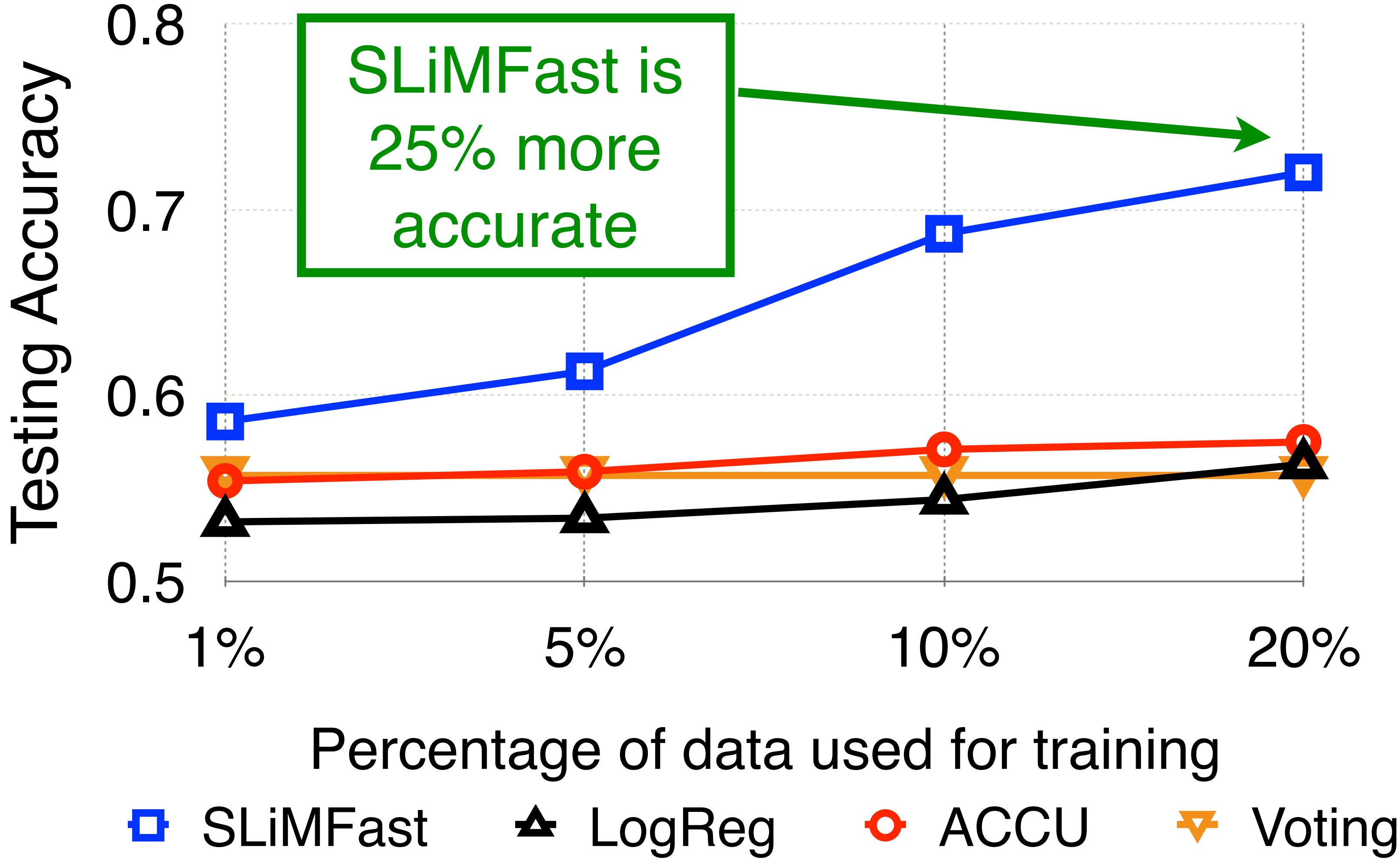
Genomics data: 2.7k sources (articles), 571 objects (gene-disease), 4 domain features (year, citation, author, journal)

Estimating the unknown true value for objects



Genomics data: 2.7k sources (articles), 571 objects (gene-disease), 4 domain features (year, citation, author, journal)

Estimating the unknown true value for objects



Genomics data: 2.7k sources (articles), 571 objects (gene-disease), 4 domain features (year, citation, author, journal)

SLiMFast

Step 1: Use probabilistic models to model source reliability

Step 2: Use domain-specific features to describe source accuracy

Step 3: Analyze the given data fusion instance to learn the model parameters

Probabilistic models for data fusion

Source observations

Source	Disease	Gene	CausedBy
OMIM	Li-Fraumeni Syndrome	CHEK2	**Yes**
Paper	Li-Fraumeni Syndrome	CHEK2	**No**

Knowledge base

CausedBy

Disease	Gene
Li-Fraumeni Syndrome	CHEK2

Probabilistic models for data fusion

Source observations

Source	Disease	Gene	CausedBy
OMIM	Li-Fraumeni Syndrome	CHEK2	**Yes**
Paper	Li-Fraumeni Syndrome	CHEK2	**No**

Knowledge base

CausedBy

Disease	Gene
Li-Fraumeni Syndrome	CHEK2

R.V.

Probabilistic models for data fusion

Source observations

Source	Disease	Gene	CausedBy
OMIM	Li-Fraumeni Syndrome	CHEK2	**Yes**
Paper	Li-Fraumeni Syndrome	CHEK2	**No**

+

-

Knowledge base

CausedBy

Disease	Gene
Li-Fraumeni Syndrome	CHEK2

R.V.

Probabilistic models for data fusion

Source observations

Source	Disease	Gene	CausedBy
OMIM	Li-Fraumeni Syndrome	CHEK2	**Yes**
Paper	Li-Fraumeni Syndrome	CHEK2	**No**

+

-

Knowledge base

CausedBy

Disease	Gene
Li-Fraumeni Syndrome	CHEK2

R.V.

Reliability scores (model parameters)

$$\Pr(\text{Object} = +1 | \text{Sources}) = \frac{1}{Z} \exp \sum_{S \,\in\, \text{Sources}} \sigma_S \cdot \text{I}[\text{S votes Object} = +1]$$

Normalizing constant (valid distribution)

Indicator function

$$\sigma_S = \log \left(\frac{\text{Accuracy of Source S}}{\text{1-Accuracy of Source S}} \right)$$

Accuracy = Probability that a source is correct

Supervised data fusion

$$\Pr(\text{Object} = +1 | \text{Sources}) = \frac{1}{Z} \exp \sum_{S \,\in\, \text{Sources}} \sigma_S \cdot \text{I}[\text{S votes Object} = +1]$$

In many cases corresponds to logistic regression

Boolean features

$$\text{I}[\text{S votes Object} = +1]$$

Supervised data fusion

$$\Pr(\text{Object} = +1|\text{Sources}) = \frac{1}{Z}\exp\sum_{S\,\in\,\text{Sources}} \sigma_S \cdot \text{I[S votes Object} = +1]$$

In many cases corresponds to logistic regression

Boolean features

$$\text{I[S votes Object} = +1]$$

No strong assumptions on:

1. independence of sources
2. accuracy being more than 0.5
3. number of observations per object

Supervised data fusion

$$\Pr(\text{Object} = +1|\text{Sources}) = \frac{1}{Z}\exp \sum_{S\,\in\,\text{Sources}} \sigma_S \cdot \text{I[S votes Object} = +1]$$

In many cases corresponds to logistic regression

Boolean features

$$\text{I[S votes Object} = +1]$$

No strong assumptions on:

1. independence of sources
2. accuracy being more than 0.5
3. number of observations per object

Simple trained model over known objects.
Highly scalable training algorithms (e.g., stochastic gradient descent).

The challenge of training data

How much data do we need to train the model?

Theorem: *We need a number of labeled examples proportional to the number of Sources.*

[On Discriminative versus Generative Classifiers, *Ng & Jordan, 2001]*

But the number of sources can be in the thousands or millions and training data is limited!!!

The challenge of training data

How can we make logistic regression practical?

$$\Pr(\text{Object} = +1|\text{Sources}) = \frac{1}{Z}\exp \sum_{S \,\in\, \text{Sources}} \sigma_S \cdot \mathrm{I}[\text{S votes Object} = +1]$$

Challenge: Limited labeled examples

The challenge of training data

How can we make logistic regression practical?

$$\Pr(\text{Object} = +1 | \text{Sources}) = \frac{1}{Z} \exp \sum_{S \,\in\, \text{Sources}} \sigma_S \cdot \text{I}[\text{S votes Object} = +1]$$

Challenge: Limited labeled examples

Limit the informative parameters of the model by using domain knowledge

The challenge of training data

How can we make logistic regression practical?

$$\Pr(\text{Object} = +1|\text{Sources}) = \frac{1}{Z} \exp \sum_{S \in \text{Sources}} \sigma_S \cdot \mathrm{I}[\text{S votes Object} = +1]$$

Challenge: Limited labeled examples

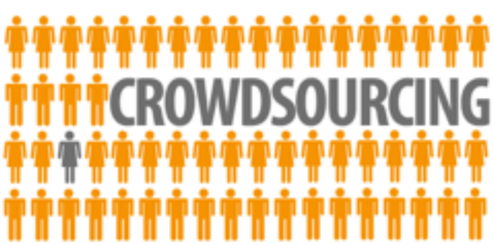
Limit the informative parameters of the model by using domain knowledge

Key Idea: Sources have (domain specific) features that are indicative of their accuracy

Source-accuracy features

(i) citations over time, (ii) journal, (iii) experimental methodology (e.g., population size), (iv) year

(i) newly registered similar to existing domain, (ii) traffic statistics, (iii) text quality (e.g., misspelled words, grammatical errors), (iv) sentiment analysis



(i) avg. time per task, (ii) number of tasks, (iii) market used

SLiMFast's data fusion model

$$\sigma_S = \log\left(\frac{\text{Accuracy of Source S}}{\text{1-Accuracy of Source S}}\right)$$

Key Idea: Sources have (domain specific) features that are indicative of their accuracy

SLiMFast's data fusion model

$$\sigma_S = \log\left(\frac{\text{Accuracy of Source S}}{\text{1-Accuracy of Source S}}\right)$$

Key Idea: Sources have (domain specific) features that are indicative of their accuracy

$$\boxed{\text{Accuracy of Source} = \text{Logistic Function}\left(\sum_{f \in \text{Features}} W_f \cdot \text{Source Value for Feature f}\right)}$$

SLiMFast's data fusion model

$$\sigma_S = \log\left(\frac{\text{Accuracy of Source S}}{\text{1-Accuracy of Source S}}\right)$$

Key Idea: Sources have (domain specific) features that are indicative of their accuracy

$$\boxed{\text{Accuracy of Source} = \text{Logistic Function}\left(\sum_{f \in \text{Features}} W_f \cdot \text{Source Value for Feature f}\right)}$$

$$\Pr(\text{Object} = +1|\text{Sources}) = \frac{1}{Z}\exp \sum_{S \in \text{Sources}} \sum_{f \in \text{Features}} \underbrace{W_f \cdot \text{Value}[f, S]} \cdot I[\text{S votes Object} = +1]$$

Normalizing constant (valid distribution)

Weighted features to capture accuracy

Indicator function

SLiMFast's data fusion model

$$\sigma_S = \log\left(\frac{\text{Accuracy of Source S}}{\text{1-Accuracy of Source S}}\right)$$

Key Idea: Sources have (domain specific) features that are indicative of their accuracy

$$\boxed{\text{Accuracy of Source} = \text{Logistic Function}\left(\sum_{f \in \text{Features}} W_f \cdot \text{Source Value for Feature f}\right)}$$

$$\Pr(\text{Object} = +1|\text{Sources}) = \frac{1}{Z}\exp \sum_{S \in \text{Sources}} \sum_{f \in \text{Features}} W_f \cdot \text{Value}[f, S] \cdot I[\text{S votes Object} = +1]$$

Still logistic regression but with **significantly fewer** parameters!

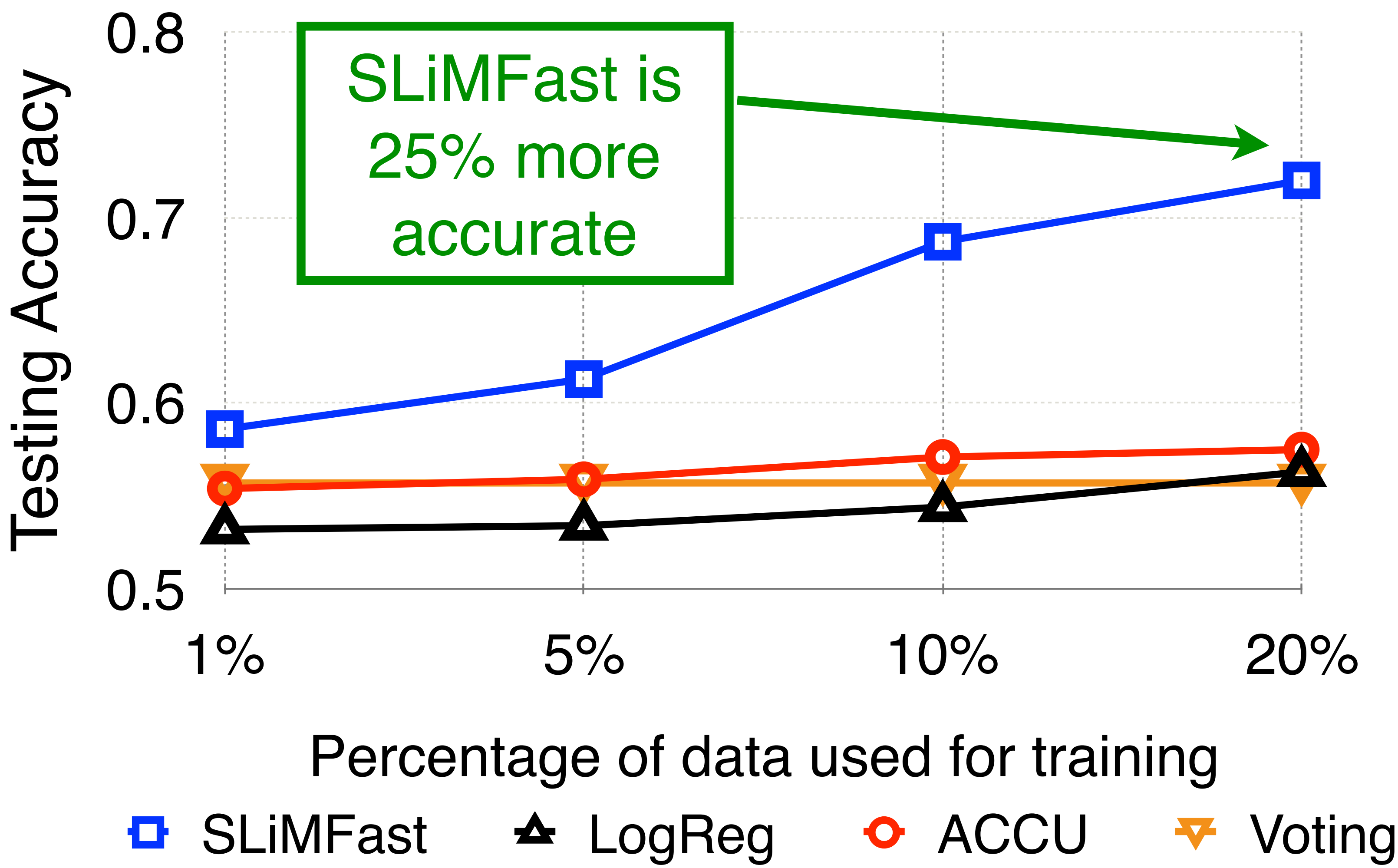
SLiMFast's guarantees for data fusion

Theorem. The error for both the estimated object values and the estimated source accuracies is proportional to $\sqrt{\frac{|K|}{|G|}}$ where $|G|$ is the number of labeled examples for objects and $|K|$ the number of features in SLiMFast.

We only need a number of labeled examples proportional to the number of Features!

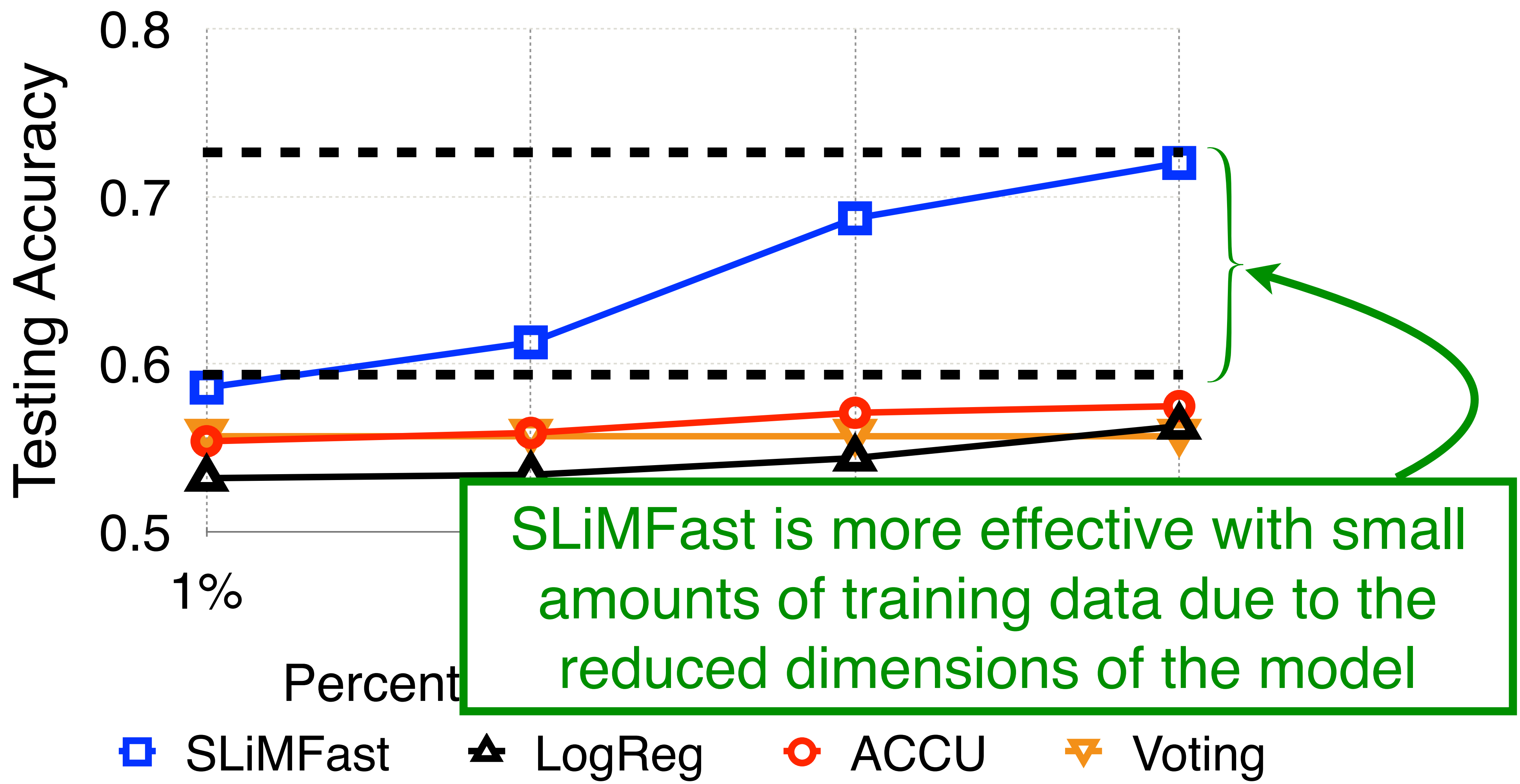
Few labeled examples are enough in practice.

SLiMFast in practice



Genomics data: 2.7k sources (articles), 571 objects (gene-disease), 4 domain features (year, citation, author, journal)

SLiMFast in practice



Genomics data: 2.7k sources (articles), 571 objects (gene-disease), 4 domain features (year, citation, author, journal)

SLiMFast achieves state-of-the-art performance

Financial data

Demonstration monitoring in the news

Crowdsourcing

SLiMFast yields accuracy improvements of up to **50% for identifying the true value of objects** and up to **10x lower error in source accuracy estimates**.

SLiMFast

Step 1: Use probabilistic models to model source reliability

Step 2: Use domain-specific features to describe source accuracy

Step 3: Analyze the given data fusion instance to learn the model parameters

Today's Agenda

Data Fusion: A quick recap

SLiMFast: Use features to describe sources

Step 1: Use probabilistic models to model source reliability

Step 2: Use domain-specific features to describe source accuracy

Step 3: Analyze the given data fusion instance to learn the model parameters

SLiMFast's Optimizer: Don't worry about ML algorithms

Beyond labeled data

In many cases labeled examples can be very limited!

How can we use SLiMFast when there is not enough training data to use supervised learning (ERM)?

Beyond labeled data

In many cases labeled examples can be very limited!

How can we use SLiMFast when there is not enough training data to use supervised learning (ERM)?

In SLiMFast we can also use unsupervised learning (e.g., EM).

Expectation Maximization

Initialize Source accuracies

1. infer Object's true value
2. adjust Src Accuracies

repeat

Beyond labeled data

In many cases labeled examples can be very limited!

How can we use SLiMFast when there is not enough training data to use supervised learning (ERM)?

In SLiMFast we can also use unsupervised learning (e.g., EM).

<u>Expectation Maximization</u>

Initialize Source accuracies

1. infer Object's true value
2. adjust Src Accuracies

repeat

Thm: We show that EM works only when there are many observations per object and when sources have an avg. accuracy $p > 0.5$

Beyond labeled data

In many cases labeled examples can be very limited!

How can we use SLiMFast when there is not enough training data to use supervised learning (ERM)?

In SLiMFast we can also use unsupervised learning (e.g., EM).

Expectation Maximization

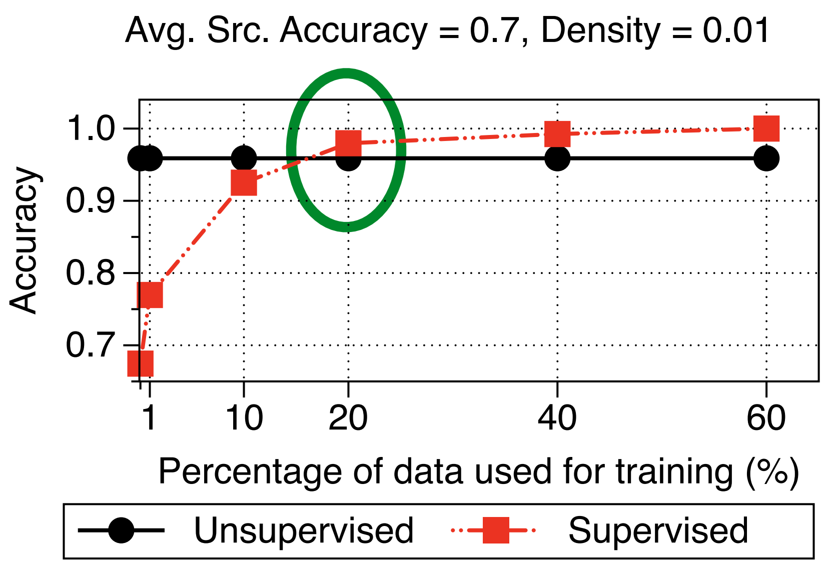
Initialize Source accuracies

1. infer Object's true value
2. adjust Src Accuracies

repeat

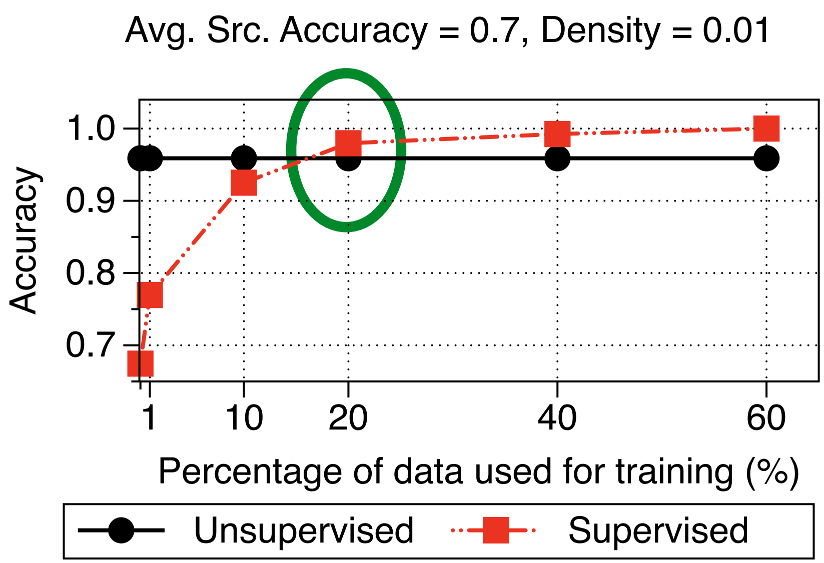
Choice: Supervised or unsupervised learning?

Our theoretical analysis says…



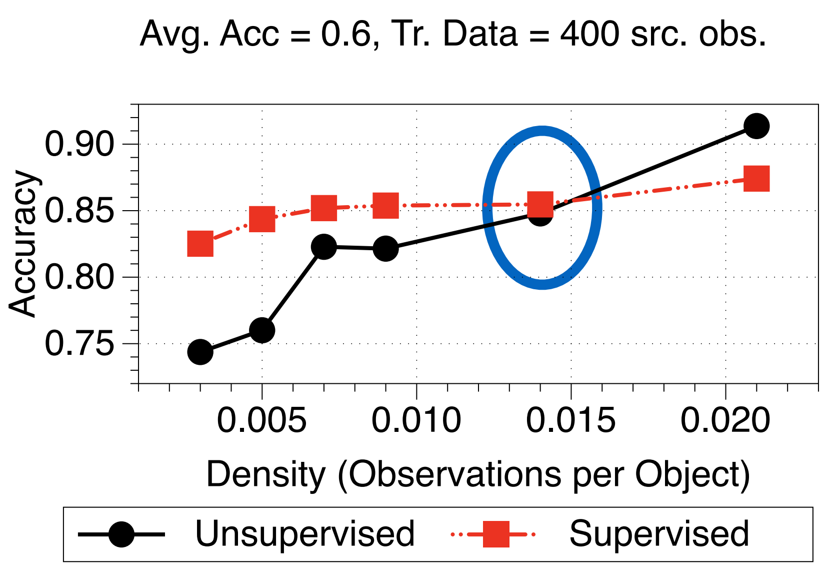
Supervised learning affected by **(i) amount of labeled data**

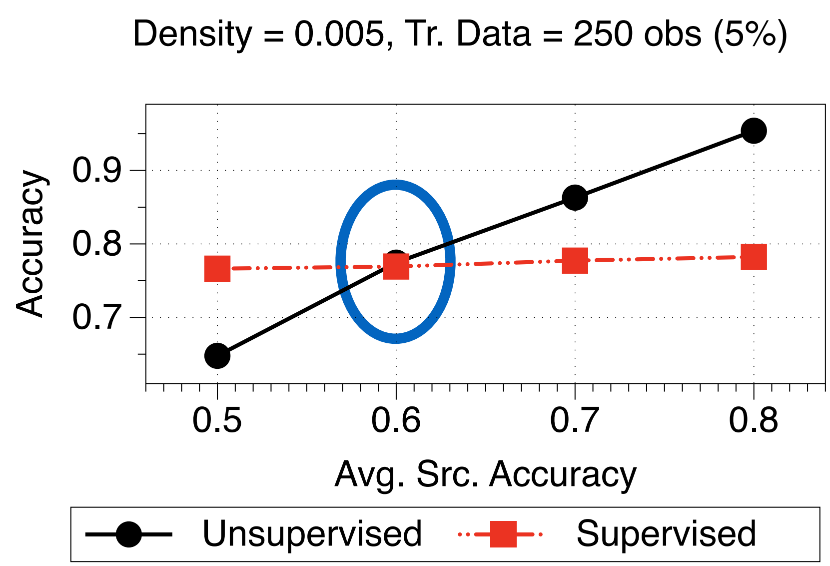
Our theoretical analysis says…



Supervised learning affected by **(i) amount of labeled data**

Unsupervised learning affected by **(ii) observation density** and **(iii) avg. src. accuracy**





The SLiMFast optimizer

Goal: Maximize accuracy of estimated true values of Objects

Choice: Supervised or unsupervised learning?

Labeled examples	Observations	Avg. src. accuracy

The SLiMFast optimizer

Goal: Maximize accuracy of estimated true values of Objects

Choice: Supervised or unsupervised learning?

Labeled examples	Observations	Avg. src. accuracy

Our theoretical analysis dictates that

G = *number of labeled examples*

IF G >> Features use *supervised learning*.

The SLiMFast optimizer

Goal: Maximize accuracy of estimated true values of Objects

Choice: Supervised or unsupervised learning?

Labeled examples	Observations	Avg. src. accuracy

Our theoretical analysis dictates that

G = *number of labeled examples*

IF G >> Features use *supervised learning*.

What if G >> Features does not hold?

The SLiMFast optimizer

Goal: Maximize accuracy of estimated true values of Objects

Choice: Supervised or unsupervised learning?

Labeled examples	Observations	Avg. src. accuracy

IF G < Features:

Each algorithm affected by different instance properties. How can we compare the two?

The SLiMFast optimizer

Goal: Maximize accuracy of estimated true values of Objects

Choice: Supervised or unsupervised learning?

Labeled examples

Observations

Avg. src. accuracy

IF G < Features:

Each algorithm affected by different instance properties. How can we compare the two?

Idea: Compare **bits of information** available to:

1. supervised learning via labeled examples
2. unsupervised learning via observations and src. accuracy

Bits of information: Supervised learning

If we are given the label for an Object the entropy of the corresponding random variable drops to zero.

From each labeled example we gain one bit of information

$$\boxed{\text{Bits = number of labeled examples}}$$

Bits of information: Unsupervised learning

How many bits of information are available in source observations?

Bits of information: Unsupervised learning

How many bits of information are available in source observations?

Expectation Maximization

Initialize Source accuracies

1. infer Object’s true value
2. adjust Src Accuracies

repeat

Bits of information: Unsupervised learning

How many bits of information are available in source observations?

Expectation Maximization

Initialize Source accuracies

1. **infer Object's true value**
2. adjust Src Accuracies

repeat

Idea: Estimate the expected number of correct object values after step 1

Bits of information: Unsupervised learning

How many bits of information are available in source observations?

Expectation Maximization
Initialize Source accuracies
1. **infer Object's true value**
2. adjust Src Accuracies
repeat

Idea: Estimate the expected number of correct object values after step 1

Use majority voting to approximate the bits of information available to unsupervised learning

Bits of information: Unsupervised learning

For each object:

1. *Compute* $p = \Pr(\text{MV gives the correct value})$

m is the number of sources
with observations for Object

Ex.: Binomial for +1,-1 values $p = 1 - \sum_{i=0}^{m/2} \binom{m}{i} A^i (1-A)^{m-i}$

Avg. accuracy of sources

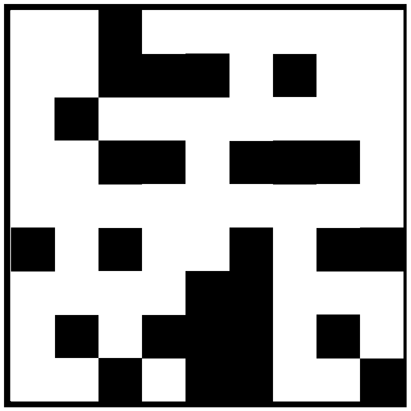
2. *Estimate bits of information*

$$\text{Bits} = 1 - \text{Entropy}(p)$$

Take into account density and average source accuracy.

Average source accuracy

Source agreement rate

X = 

$$X_{i,j} = \frac{\text{Agreements - Disagreements between Sources i and j}}{\text{Overlap between Sources i and j}}$$

The agreement rate depends on the source accuracies.
Assumptions: (i) independence, (ii) same accuracy

$$X_{i,j} = A^2 + (1 - A)^2 - 2A(1 - A)$$

Estimate average accuracy A *using the information in the entries of matrix* X

The SLiMFast optimizer

G *= number of labeled examples*

IF G >> Features use *supervised learning*.

Otherwise:

U *= bits of information for unsupervised learning*

IF G > U use *supervised learning* ELSE *unsupervised learning*.

The SLiMFast optimizer

G = *number of labeled examples*

IF G >> Features use *supervised learning*.

Otherwise:

U = *bits of information for unsupervised learning*

IF G > U use *supervised learning* ELSE *unsupervised learning*.

Our optimizer selects the right learning algorithm 19/20 cases (4 datasets, 5 setups)

SLiMFast: Data fusion with guarantees

1. Simple features can help identify inaccurate data and unreliable sources.

Think of source features not algorithms!

2. Use simple discriminative models; in most cases logistic regression is enough.

3. First optimizer to choose between ML algorithms.

SLiMFast: Data fusion with guarantees

1. Simple features can help identify inaccurate data and unreliable sources.

Think of source features not algorithms!

2. Use simple discriminative models; in most cases logistic regression is enough.

3. First optimizer to choose between ML algorithms.

Thank you!
thodrek@stanford.edu