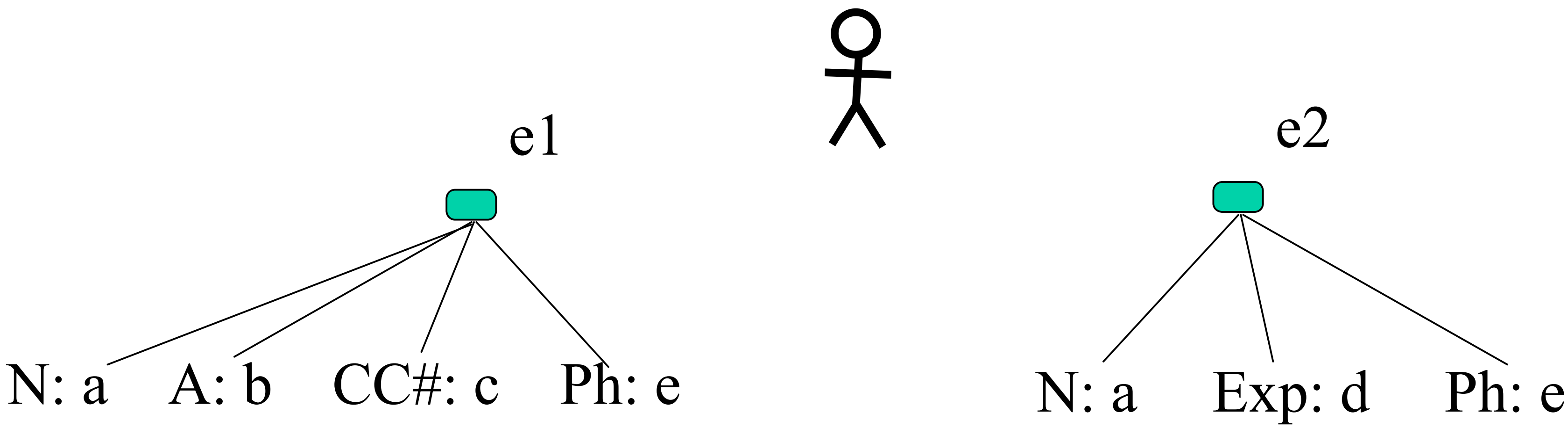
Evaluating Entity Resolution Results

David Menestrina, Steven Whang,

Hector Garcia-Molina

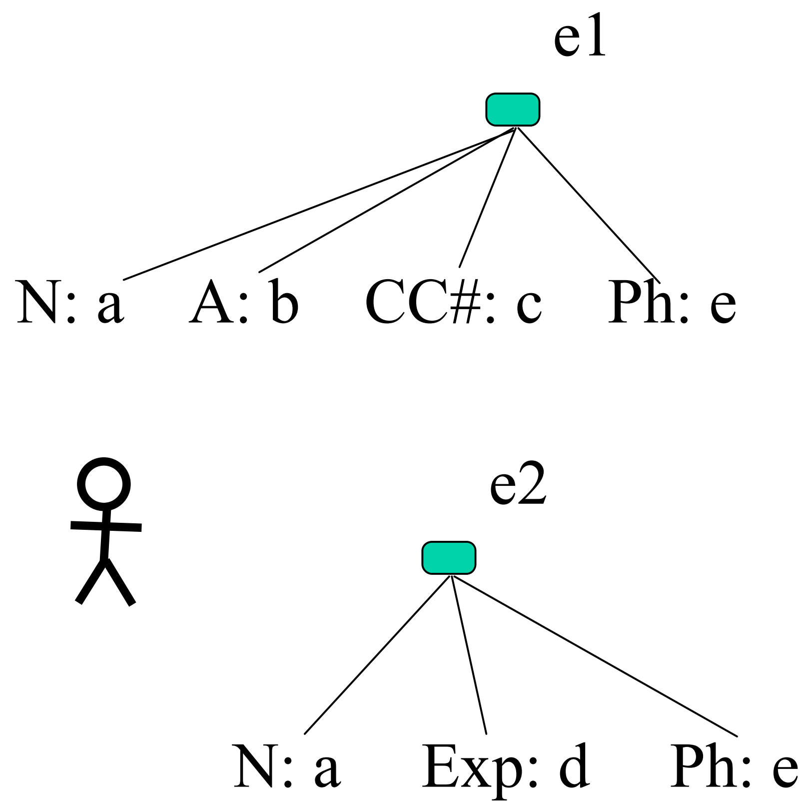
Stanford University

Entity Resolution



Applications

- comparison shopping
- mailing lists
- classified ads
- customer files
- counter-terrorism



Evaluating ER Results

R1 = a, b, c, d, efgh

R2 = ab, cd, ef, gh

G = ab, cd, efgh

Pairwise Recall

R1 = a, b, c, d, efgh

R2 = ab, cd, ef, gh

G = ab, cd, efgh

Pairwise Recall

R1 = a, b, c, d, efgh	R2 = ab, cd, ef, gh	G = ab, cd, efgh
Pairs: ef, eg, eh, fg, fh, gh	Pairs: ab, cd, ef, gh	Pairs: ab, cd, ef, eg, eh, fh, fh, gh

Pairwise Recall

R1 = a, b, c, d, efgh	R2 = ab, cd, ef, gh	G = ab, cd, efgh
Pairs: ef, eg, eh, fg, fh, gh	Pairs: ab, cd, ef, gh	Pairs: ab, cd, ef, eg, eh, fh, fh, gh
6 pairs, all in G		8 pairs

Pairwise Recall

R1 = a, b, c, d, efgh

Pairs:
ef, eg, eh,
fg, fh, gh

6 pairs, all in G

Recall = 6/8 = 75%

R2 = ab, cd, ef, gh

Pairs:
ab, cd, ef, gh

G = ab, cd, efgh

Pairs:
ab, cd, ef, eg,
eh, fh, fh, gh

8 pairs

Pairwise Recall

R1 = a, b, c, d, efgh	R2 = ab, cd, ef, gh	G = ab, cd, efgh
Pairs: ef, eg, eh, fg, fh, gh	Pairs: ab, cd, ef, gh	Pairs: ab, cd, ef, eg, eh, fh, fh, gh
6 pairs, all in G	4 pairs, all in G	8 pairs
Recall = 6/8 = 75%	Recall = 4/8 = 50%	

Pairwise Recall

R1 = a, b, c, d, efgh	R2 = ab, cd, ef, gh	G = ab, cd, efgh
Pairs: ef, eg, eh, fg, fh, gh	Pairs: ab, cd, ef, gh	Pairs: ab, cd, ef, eg, eh, fh, fh, gh
6 pairs, all in G	4 pairs, all in G	8 pairs
Recall = 6/8 = 75%	Recall = 4/8 = 50%	

Pairwise F1

$$PairPrecision(R,G) = \frac{|Pairs(R) \cap Pairs(G)|}{|Pairs(R)|}$$

$$PairRecall(R,G) = \frac{|Pairs(R) \cap Pairs(G)|}{|Pairs(G)|}$$

$$pF_1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

Merge Distance

R1 = a, b, c, d, efgh

R2 = ab, cd, ef, gh

G = ab, cd, efgh

Merge Distance

R1 = a, b, c, d, efgh

R2 = ab, cd, ef, gh

G = ab, cd, efgh

a, b → ab

c, d → cd

G = ab, cd, efgh

Merge Distance

R1 = a, b, c, d, efgh

R2 = ab, cd, ef, gh

G = ab, cd, efgh

a, b → ab
c, d → cd

ef, gh → efgh

G = ab, cd, efgh

G = ab, cd, efgh

Merge Distance

R1 = a, b, c, d, efgh

R2 = ab, cd, ef, gh

G = ab, cd, efgh

a, b → ab
c, d → cd

ef, gh → efgh

G = ab, cd, efgh

G = ab, cd, efgh

Distance = 2

Distance = 1

Merge Distance

Minimum number of splits and merges to get from R to G (splits first)

[Al-Kamha, et al. 2004]

Variation of Information

$$VI(R,G) = H(R) + H(G) - 2\mathrm{I}(R,G)$$

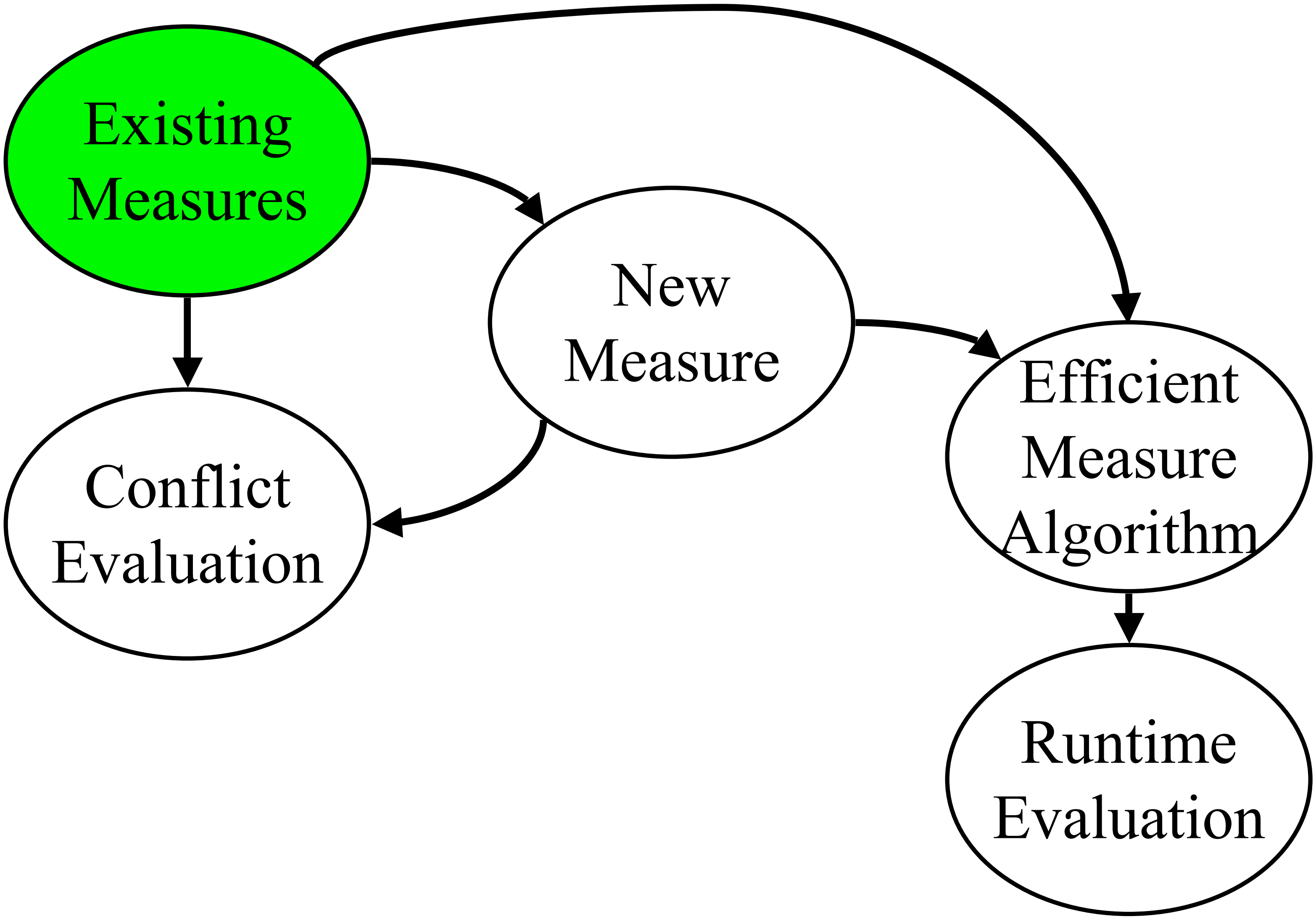
$$H(R) = -\sum_{r \in R} \frac{|r|}{N} \log \frac{|r|}{N}$$

$$I(R,G) = \sum_{r \in R} \sum_{g \in G} \frac{|r \cap g|}{N} \log \frac{|r \cap g| \times N}{|r| \times |g|}$$

Conflicts

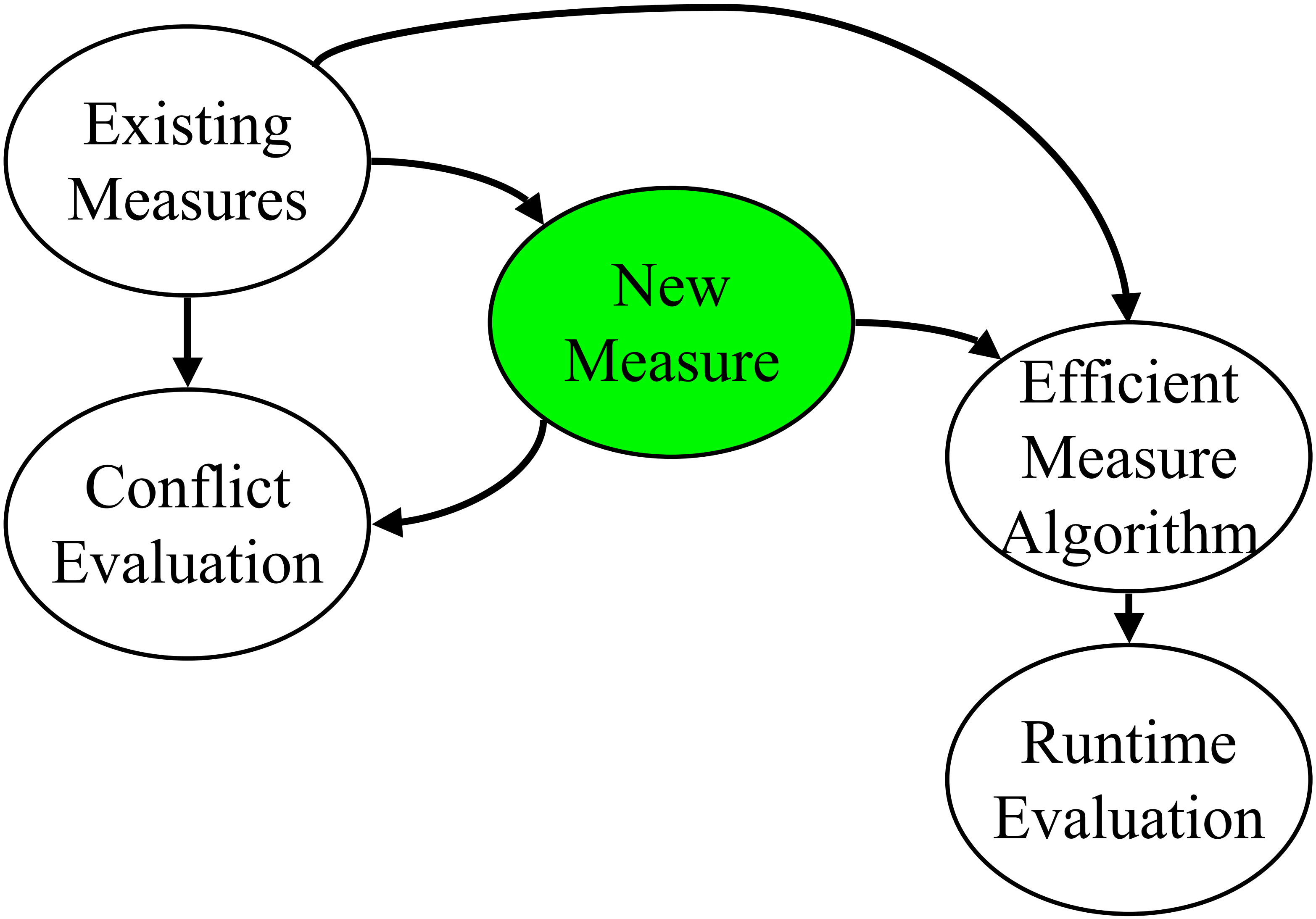
	R1	R2
Pairwise Recall	75%	50%
Merge Distance	2	1
Variation of Information	0.5	0.5

Road Map



Existing Measures
New Measure
Efficient Measure Algorithm
Conflict Evaluation
Runtime Evaluation

Road Map



Existing Measures
New Measure
Conflict Evaluation
Efficient Measure Algorithm
Runtime Evaluation

Generalized Merge Distance

- Cost of split, merge defined by functions:

  $f_s(x, y)$, $f_m(x, y)$

  e.g.,

  $f_m(x, y) = 1$

  $f_m(x, y) = xy$

- Distance = cost of minimum-cost path

Generalized Merge Distance

R1 = a, b, c, d, efgh

R2 = ab, cd, ef, gh

G = ab, cd, efgh

a, b → ab

c, d → cd

ef, gh → efgh

G = ab, cd, efgh

G = ab, cd, efgh

Distance
$= f_m(1, 1) + f_m(1, 1)$

Distance
$= f_m(2, 2)$

f(x, y) = 1

R1 = a, b, c, d, efgh

R2 = ab, cd, ef, gh

G = ab, cd, efgh

a, b → ab
c, d → cd

ef, gh → efgh

G = ab, cd, efgh

G = ab, cd, efgh

Distance
$= f_m(1, 1) + f_m(1, 1)$
$= 1 + 1 = 2$

Distance
$= f_m(2, 2)$
$= 1$

$f(x, y) = xy$

R1 = a, b, c, d, efgh	R2 = ab, cd, ef, gh	G = ab, cd, efgh

a, b → ab
c, d → cd

ef, gh → efgh

G = ab, cd, efgh	G = ab, cd, efgh
Distance $= f_m(1, 1) + f_m(1, 1)$	Distance $= f_m(2, 2)$

$f(x, y) = xy$

R1 = a, b, c, d, efgh

R2 = ab, cd, ef, gh

G = ab, cd, efgh

a, b → ab

c, d → cd

ef, gh → efgh

G = ab, cd, efgh

G = ab, cd, efgh

Distance
$= f_m(1, 1) + f_m(1, 1)$
$= 1 \times 1 + 1 \times 1 = 2$

Distance
$= f_m(2, 2)$
$= 2 \times 2 = 4$

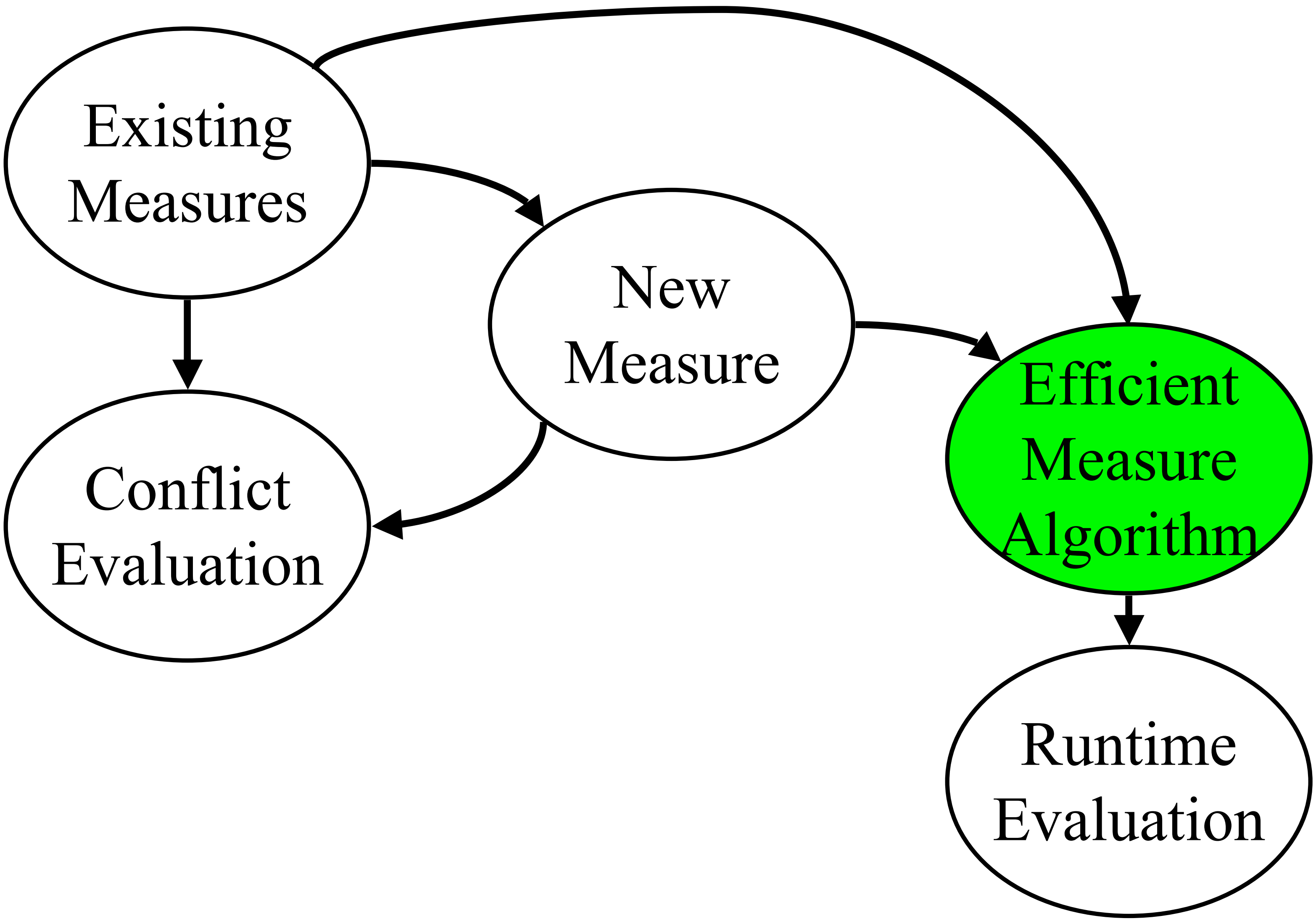
Relationships Between Measures

- Merge Distance: $f_m(x, y) = 1$, $f_s(x, y) = 1$
- Pairwise Recall: $f_m(x, y) = xy$, $f_s(x, y) = 0$
- Pairwise Precision: $f_m(x, y) = 0$, $f_s(x, y) = xy$
- Variation of Information:

  $f_m(x, y) = f_s(x, y) = h(x + y) - h(x) - h(y)$

$$h(x) = \frac{x}{N} \log \frac{x}{N}$$

Road Map



Existing Measures
New Measure
Conflict Evaluation
Efficient Measure Algorithm
Runtime Evaluation

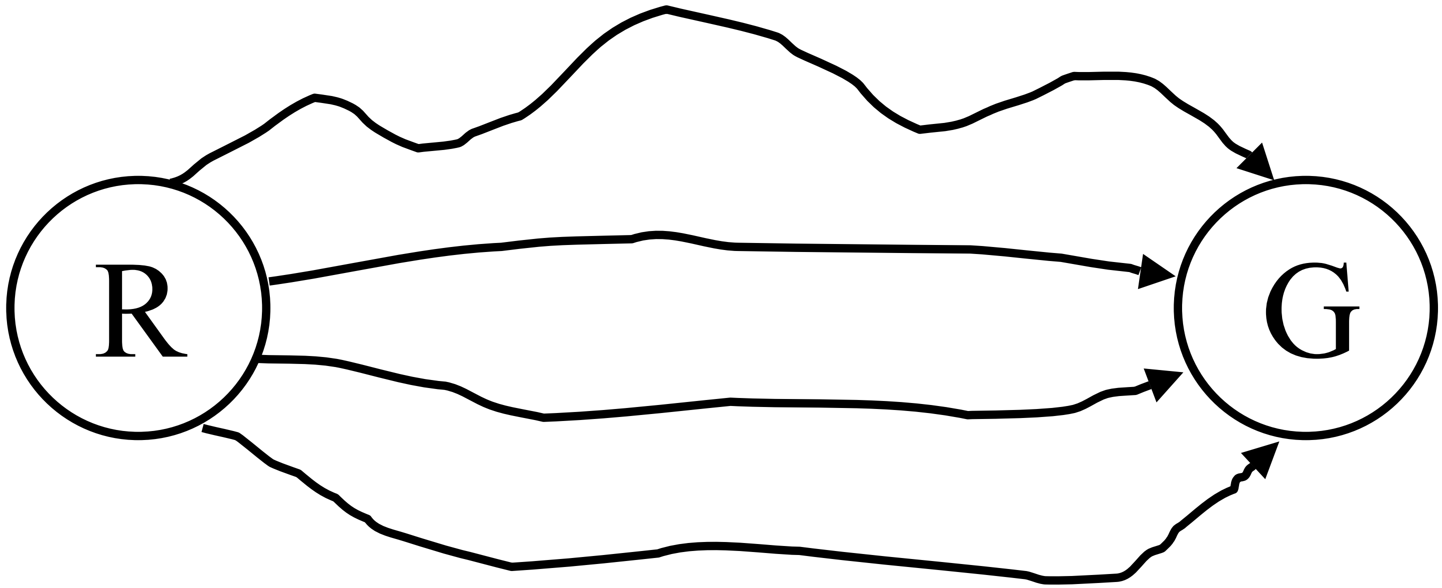
Slice Algorithm

- Linear time algorithm
- Extra property required:

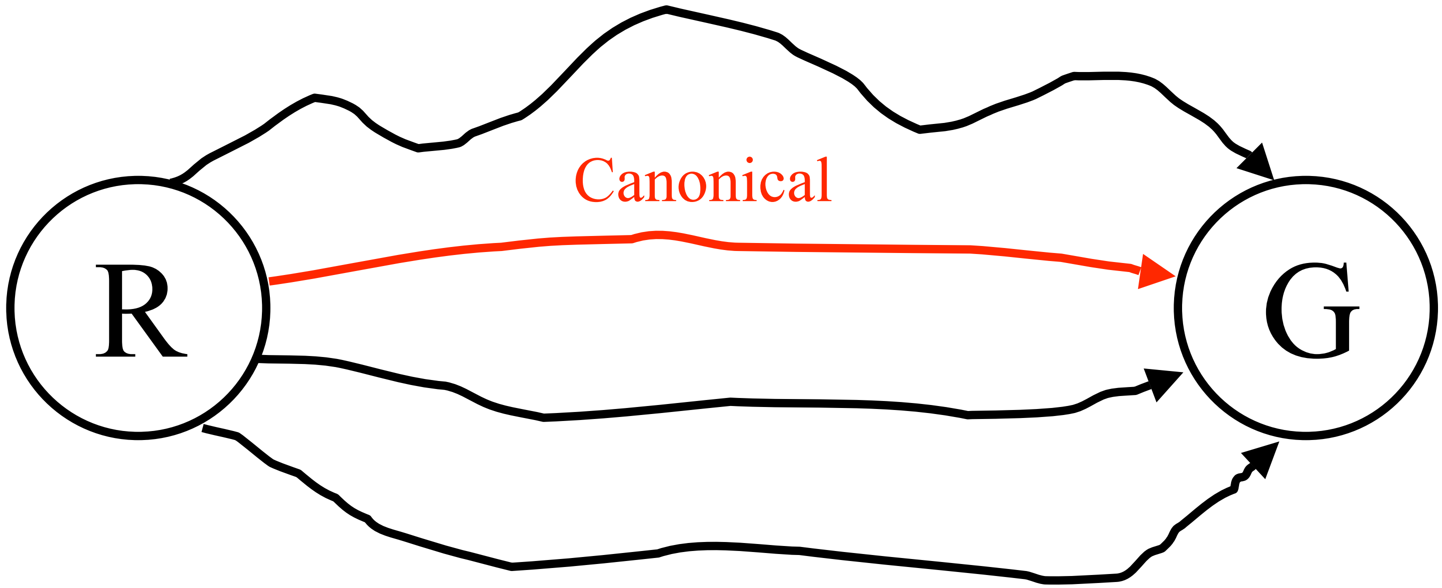
  $$f(x, y) + f(x+y, z) = f(x, z) + f(x+z, y)$$

- Cost functions for pairwise, merge distance, and variation of information (and many others) satisfy property

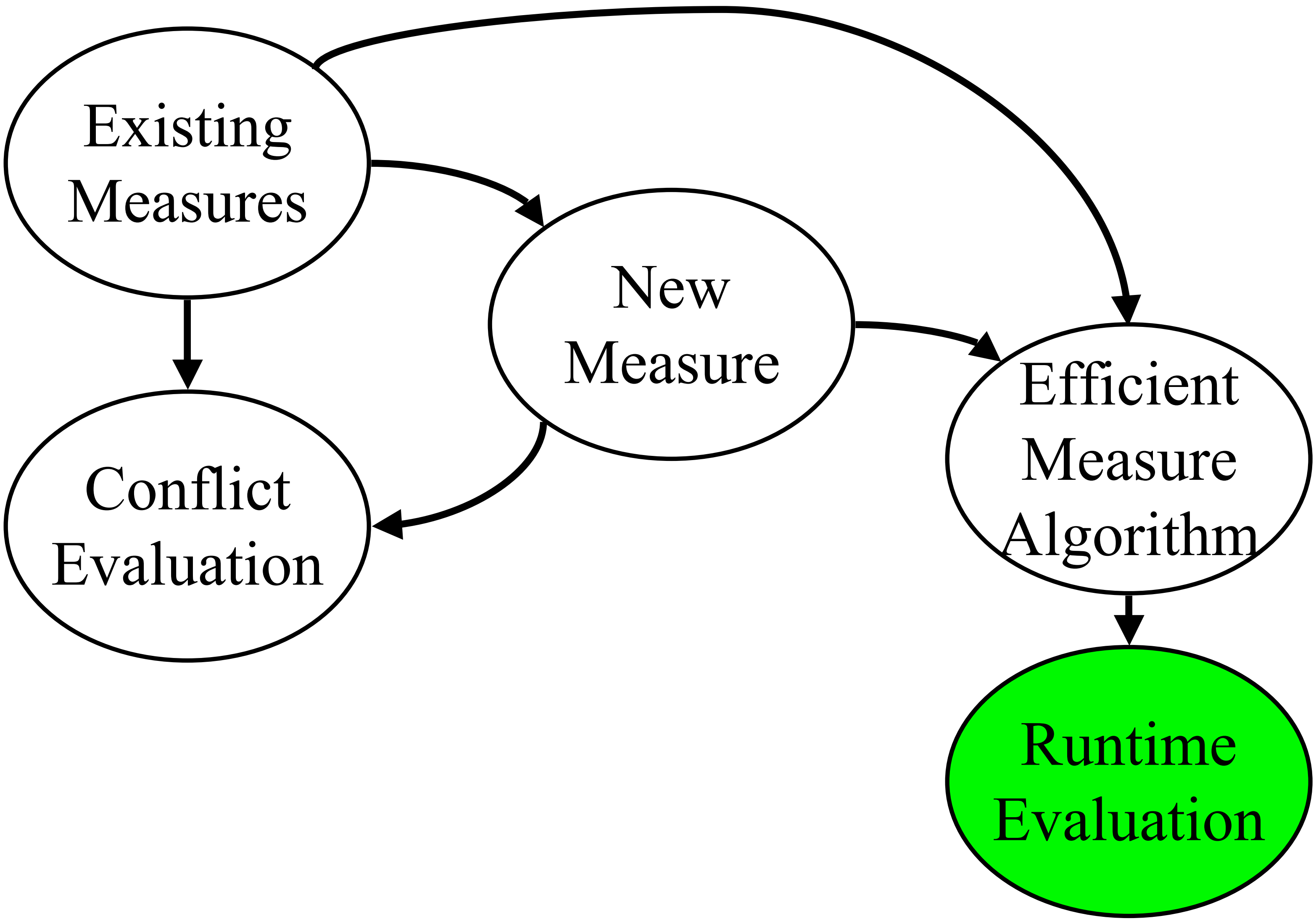
Slice Algorithm



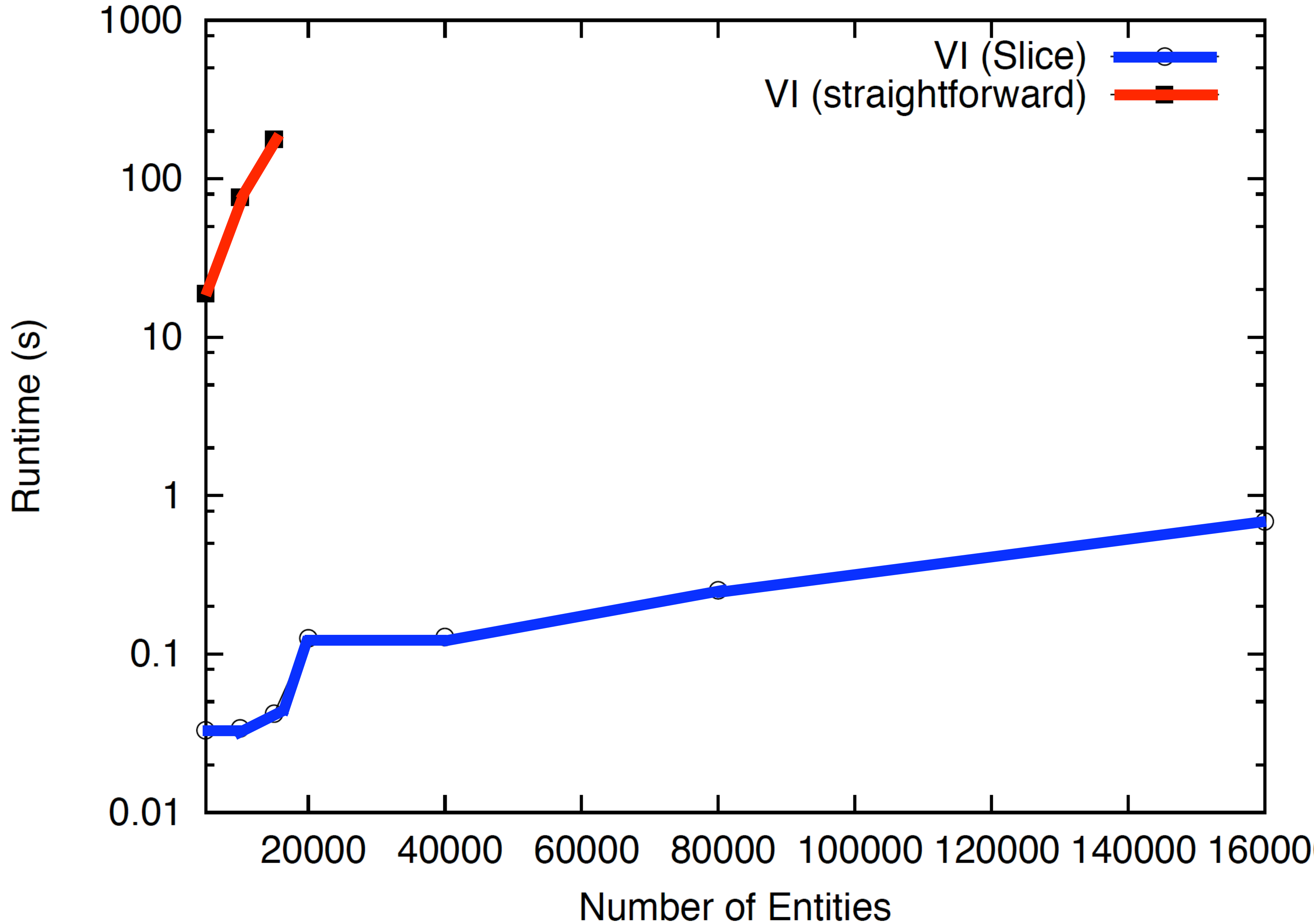
Slice Algorithm



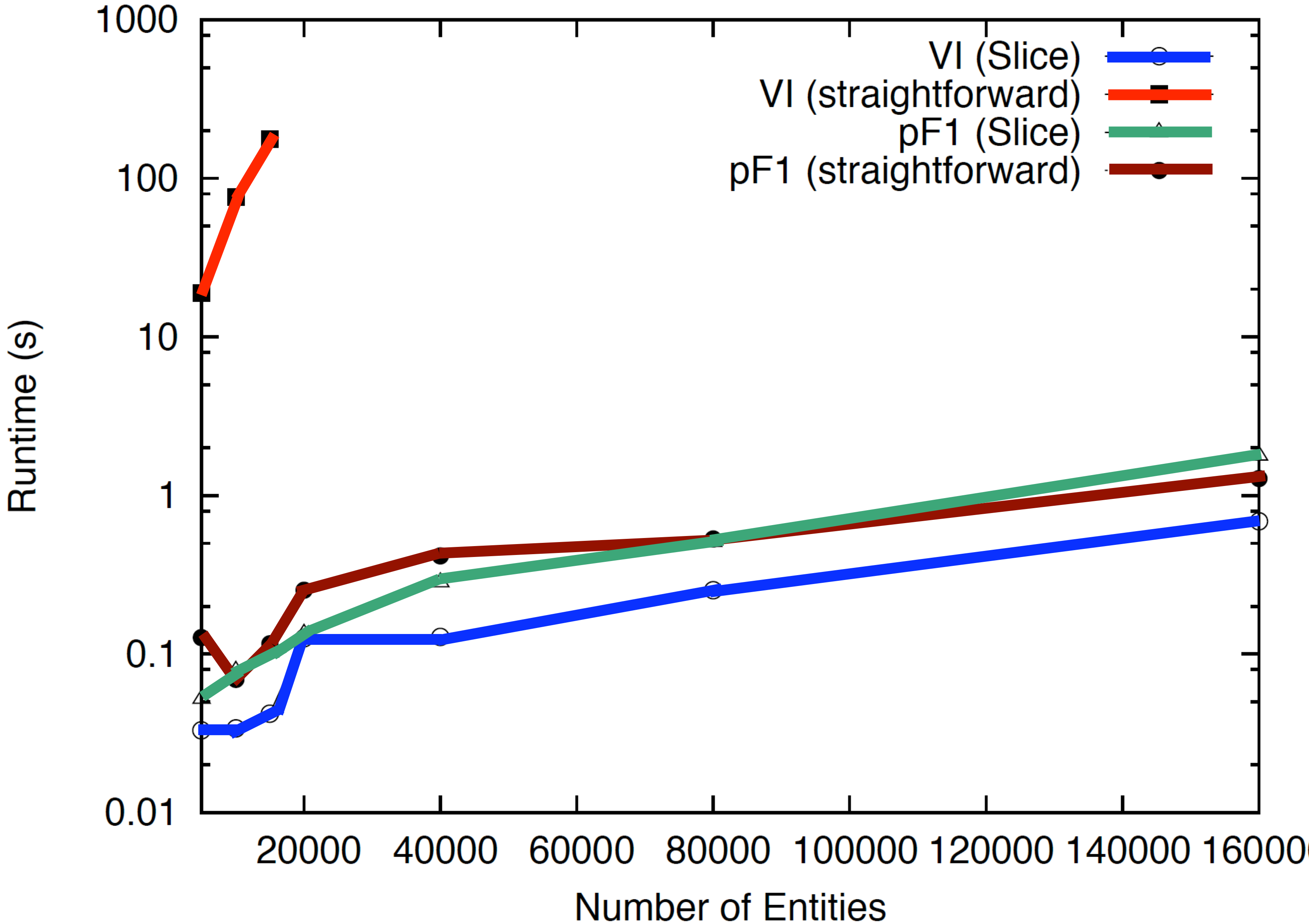
Road Map


Existing Measures
New Measure
Efficient Measure Algorithm
Conflict Evaluation
Runtime Evaluation

Slice Runtime



Slice Runtime



Conclusion

- Existing measures conflict
- Generalized merge distance provides
  - Configurability to suit different applications
  - Framework for exploring relationships between measures
  - Efficient algorithm (Slice) for computing many distance measures

Thanks!

Relationships Between Measures

- Merge Distance: GMD(R, G)
  where $f_m(x, y) = 1$, $f_s(x, y) = 1$
- Pairwise Recall: 1- GMD(R,G)/GMD(⊥,G)
  where $f_m(x, y) = xy$, $f_s(x, y) = 0$
- Pairwise Precision: 1- GMD(R,G)/GMD(R,⊥)
  where $f_m(x, y) = 0$, $f_s(x, y) = xy$
- Variation of Information: GMD(R, G)
  where $f_m(x, y) = f_s(x, y) = h(x + y) - h(x) - h(y)$

$$h(x) = \frac{x}{N} \log \frac{x}{N}$$