



Mathematical Innovations for Heterogeneous Data Access

Stefanos Damianakis
CEO, Netrics

Tuesday June 24, 2008 – 9am

Decades of Figuring Out How to Interoperate

- **Client/Server, CORBA, EIA, EII, ESB, SOA, WOA, ...**
- **We are slowly converging and evolving...**
 - *one will dominate*
 - *but there will never be only one*

The Dreaded “Silos of Information”

- Silos need to be opened up and connected
- Getting silos to talk to each other is a hard problem

Services Success!

- The Services approach solved the interoperability problem
- Is the whole problem solved?
 - *No way!*

Silo Data is Not Perfect!

“More than 25% of critical data within large businesses is somehow inaccurate or incomplete.”

– Gartner

Silo Data is Not Perfect!

- **Any Services solution that assumes perfect silo data is doomed to failure**
 - *What good are silos “connected” at the application level but not the data level?*

Heterogeneous Data Access

- **Implement a distributed Services solution**
 - *Add mathematics to help deal with imperfect data*
 - *Complement any query language*
 - *e.g.: SQL, ODBC, JDBC, SPARQL, XQuery, etc...*

Why Does this Problem Happen?

- Data is not “perfect” and never will be...
- Inconsistencies which wouldn't trouble a human, make data useless to a computer:
 - *It can't retrieve it*
 - *It can't compare it*
 - *It can't match it*
 - *It can't link it*
- Data problems go undetected and unresolved

Solving the Whole Problem

**Try to make the data perfect...
(Sisyphus anyone?)**

OR

**Enable Services to handle the
inconsistencies
(which humans do naturally...)**

What Makes Connecting Silo Data a Difficult Problem?

- **Data is never perfect – it will never be perfect**
 - *data changes over time in ways that change*
 - *CPUs and DBMSes are based on an “exact” world*
- **Human expertise is impossible to extract and program explicitly**

The Core Problem: A Human-to-Computer Gap

- **Humans usually perceive information approximately and easily tolerate data errors and variations**
 - *Humans and computers introduce inconsistencies*
 - *Other inconsistencies are intentionally introduced (e.g. by customers trying to avoid billing)*
- **Computer software is usually exact and unforgiving**
 - *Determining equality or inequality is easy*
 - *“Damianakis” = “Damianakis”*
 - *“Damianakis” ≠ “Smith”*
 - *Determining similarity is difficult*
 - *“Damianakis” ≈ “Damamakis”*

Conventional Data Matching

- **Simplistic substring matching**
 - *First five letters of last name + first letter of first name, etc...*
- **Dictionary Meta-data**
 - *Store, update, and manage known errors a priori*
- **Coding Methods**
 - *Soundex (1917)*
 - *Encodes strings into leading character followed by a three digit number*
 - *Rogers → (R, g=2, r=6, s=2) → R262*
 - *Rodgers → (R, d=3, g=2, r=6) → R326*
 - *NYSIIS (1970)*
 - *Similar to Soundex except that all vowels become “A”s and a pure alpha code is generated*
 - *Jaro-Winkler (1999), Metaphone (1990), Double Metaphone (2000)*
 - *DM: ...it uses a much more complex ruleset for coding than its predecessor; for example, it tests for approximately 100 different contexts of the use of the letter C alone.*
- **String Distance Methods (1969)**
 - *Computes the cost of converting one string to another via basic operations*
 - *Examples include Levenshtein Distance, String Edit Distance, Hamming Distance, Block Distance, q-gram Distance*

All of the above are Low Resolution solutions with Local Visibility that do not tolerate a wide range of data errors

Deficiencies of Soundex and NYSIIS

	Last Name	Soundex Code	NYSIIS Code
False Negatives	Skotnica	S235	SCATNAC
	Kotnica	K352	CATNAC
	Thuhangt	T523	TANGT
	Hangthu	H523	HANGT
	Oritz	O632	ORAT
Obitz	O132	OBAT	
False Positives	Morrow	M600	MAR
	Mayro	M600	MAR
	Maloney	M450	MALANY
	Malinow	M450	MALAN

Deficiencies of Edit-Distance

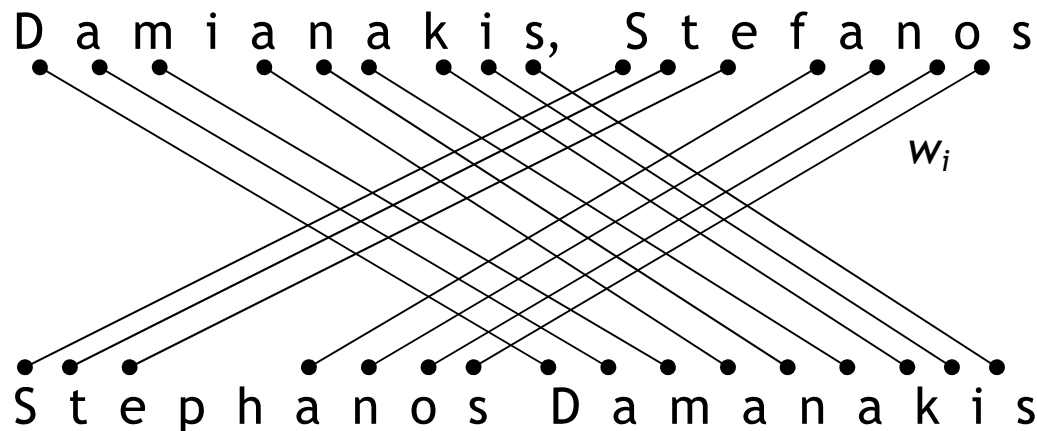
- **Computationally expensive:**
 - *Query string (length n)*
 - *Edit window (length m)*
 - *Computation increases $n \times m$*
- **For efficiency, must set small queries and edit windows**
- **Limits ability to find typical errors in fielded data, e.g. transpositions, mis-fielded data**

Mathematically Modeling Human Similarity

- **Model human notion of similarity**
- **Replace string compare functionality with advanced bipartite graph matching**
 - *High efficiency*
 - *Global visibility*
- **Compensates for errors in services request and data source**
- **High accuracy with both structured and semi-structured data**

Advanced Bipartite Graph Matching

- Uses Bipartite Graphs to compute similarity metrics
- Mathematically models a human notion of similar
- Captures a truer, richer notion of similarity than conventional methods



Mathematically Modeling Human Similarity...

- **Handles All Common (and Uncommon) Issues – eg:**
 - *Product number errors (“AB2530” for “AW2530”)*
 - *Letters out of order (“Patsries” for “Pastries”)*
 - *Words out of order (“Truffle Tower” for “Tower Truffle”)*
 - *Fused words (“Windtunnel” for “Wind tunnel”)*
 - *Split words (“Wind tunnel” for “Windtunnel”)*
 - *Missing letters (“Windtunel” for “Windtunnel”)*
 - *Extraneous letters (“Chocolatwe” for “Chocolate”)*
 - *Multiple errors (“Trufle Tripl Towr” for “Triple Truffle Tower”)*
 - *Incomplete words (“hocolate” for “Chocolate”)*
 - *Extraneous information (“rflkj Chocolate dhhg” for “Chocolate”)*
 - *Incorrect or missing punctuation (“Lemon-log” for “Lemon log”)*
 - *Incorrect fielding in fielded data sets*
- **And much more – including multilingual support (40+)**

What Makes Connecting Silo Data a Difficult Problem?

- **Data is never perfect – it will never be perfect**
 - *data changes over time in ways that change*
 - *CPUs and DBMSes are based on an “exact” world*
- **Human expertise is impossible to extract and program explicitly**

The Limitations of Conventional Record Matching

- **An developer/statistician must explicitly define the matching criteria:**
 - *Specify the rules that constitute a record match (a priori)*
 - *Specify the field weights for each of the rules*
 - *If FIRST > 0.70 and LAST > 0.85 and SEX = MALE and STREET > 0.50 and BDAY = 1.00 and SSN = 1.00*
 - *If FIRST > 0.93 and SEX = FEMALE and SNN > 0.95 and BDAY = 1.00*
 - ...
- **This approach has many drawbacks**
 - *Ad hoc – lot's of guesswork*
 - *Rules must be known a priori*
 - *How can we know every possible match rule in advance?*
 - *Guessing/setting the weights adds significant complexity and is a major source of error*
 - *Why is the FIRST weight 0.70 and not 0.68?*
 - *Only a limited set of fields are used*
 - *Adding even one field requires re-programming and new weights*

Adding More Complexity to Conventional Record Matching

- Fellegi-Sunter Probabilistic Record Matching (1968)

The relative frequency, or uniqueness, of the matching attributes must be taken into consideration

– Ivan Fellegi and Alan Sunter “A Theory for Record Linkage” *Journal of the American Statistical Association*, Vol 64, 1968

• Honor	Wassoombo	06/22/1977
H.	Wassoombo	06/21/1977
• Robert	Smith	06/22/1977
R.	Smith	06/21/1977

- Incisive, intuitive idea – but it’s only part of the solution

Mathematically Modeling Overcomes Conventional Limitations

- **Learns to weight different data fields automatically**
- **Automatically discovers rules/patterns**
- **Automatically takes advantage of ALL available data to improve accuracy (even sparsely populated fields)**
- **Learns directly from non-technical users who are the domain experts**
 - *Nothing gets lost in translation*
 - *Can be designed to learn incrementally*
 - *Can be customized to individual users*

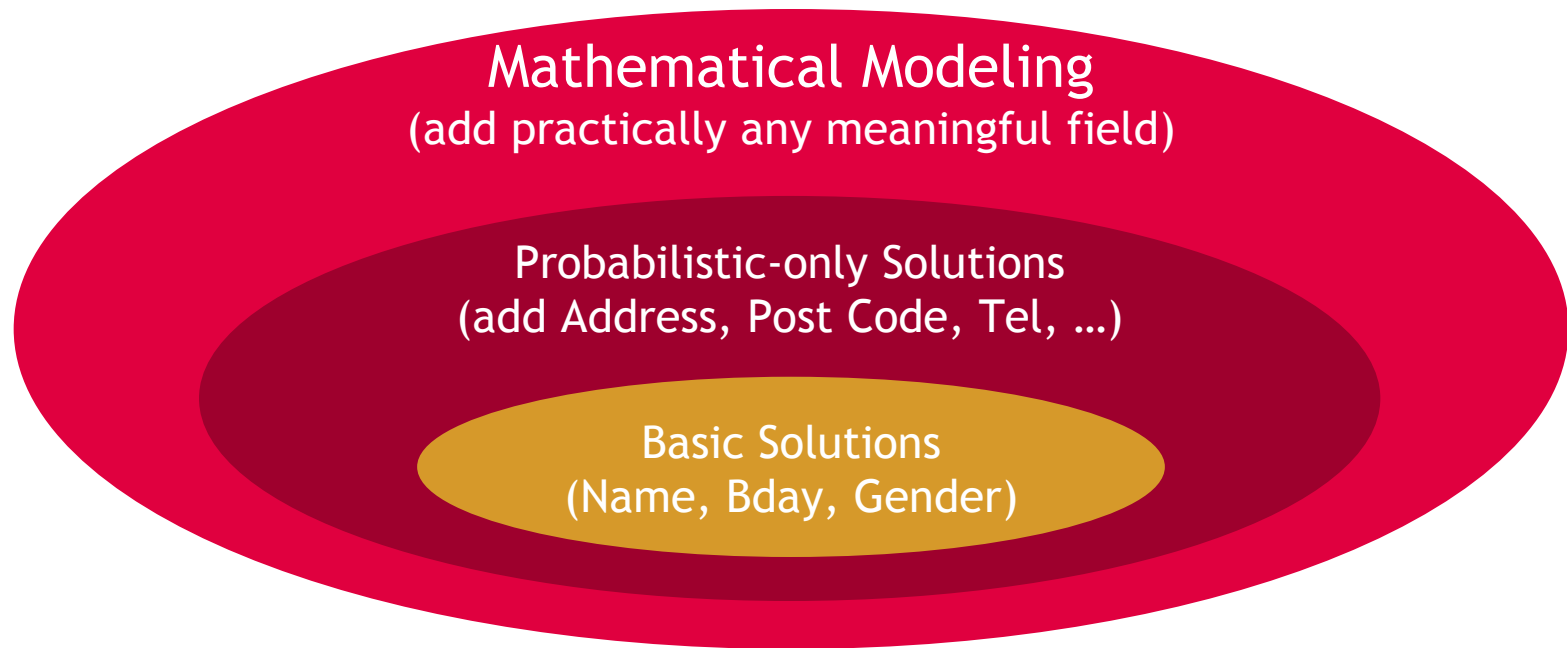
Note: Matching and Learning can be implemented independently or sequentially

Learn Automatically from Non-Technical Users

- **Displays two records to a human domain expert and asks:**
Do these two records represent the same entity?
 - **YES or NO?**
- **Repeat across a training set**
 - *Hundreds to a few thousand records, depending on the concept that must be learned and required accuracy*
- **Does not need to be told why – the algorithm deals with why by using its internal state (variables)**
- **No weights or variables need to be set by operator**
- **Ultimately, the algorithm learns what the human is doing and is even able to identify errors that that same human makes**

Leverage More Information

- The math model analyzes any additional attribute that help a human expert make a decision
 - *E.g.: blood type, parents, siblings, dept, etc.*
- Additional fields increase record matching accuracy



Record Matching Comparison

Record Matching Model...	Mathematical Modeling	Conventional
Building	<ul style="list-style-type: none">• Auto-generated for each specific business need<ul style="list-style-type: none">– <i>Tailored to data, market, decision rules, and business requirements</i>• Learns from business experts based on examples	<ul style="list-style-type: none">• “Canned” models developed for “typical” data and business requirements• Customization requires programmers to interview business experts and guess about rules and weights
Updating <ul style="list-style-type: none">• special cases• new data• corrections• etc...	<ul style="list-style-type: none">• Add examples	<ul style="list-style-type: none">• Tweak and hope nothing breaks• May require starting from scratch

The Power of Mathematics: Innovation

Problem	Conventional Solutions	Mathematical Innovations	Advantages of Mathematics
Data Matching (compare data elements)	Soundex, NYSIIS, Edit Distance, etc	Mathematically Model Human Similarity (bipartite graphs)	<ul style="list-style-type: none"> • Superior Accuracy • Symmetric error-tolerance • No Guessing (of rules and parameters) • Computational Efficiency & Scalability • Data Independence <ul style="list-style-type: none"> • people, assets, products, companies, claims, transactions, etc.
Record Matching (compare/link database records)	Custom, Manual Matching Rule Sets with optional statistical parameters	Mathematically Model Human Decisions (machine learning)	<ul style="list-style-type: none"> • Engineering Efficiency <ul style="list-style-type: none"> • easy to maintain and refine • Multi-lingual • Real-time • Sparse data support built-in • Embeddable • Quick and easy deployment • DBMS independent

Advantages of Mathematical Modeling

- **No need to clean data**
 - *Software excels with imperfect data*
- **No matching rules to build and maintain**
 - *No need to:*
 - *Check for different errors in the various fields*
 - *Guess and enumerate all possible record matching rules*
- **Pre-deployment**
 - *Less time and effort to deploy*
- **Post-deployment**
 - *Virtually no ongoing management to maintain matching accuracy*

Bottom Line

- **Any Services solution must deal with the imperfections inherent in “real-world” data...**
 - *otherwise silos will never be truly connected*
 - *and true value will not be realized by the enterprise*

Questions?



**For more information
please visit:**

www.Netrics.com

or email me at

**Stefanos.Damianakis @
Netrics.com**