Deterministic, Probabilistic, or Fuzzy?: A Primer on the Search Algorithms that Drive MPI Quality

by Sorin Gudea

MPI quality relies on the sophisticated algorithms that drive patient lookup. As health information exchange gains momentum, accurate and complete record matching will be more critical than ever.

Record matching has been around since the dawn of the database, but much has changed since. Heathcare organizations have merged to form enterprises that manage millions of patient encounters, and unaffiliated facilities are beginning to form regional health information exchanges such as RHIOs. The best method for identifying and compiling an individual's complete health information is taking on a whole new significance. Maintaining clean databases—always a priority for HIM professionals—is becoming more vital than ever.

When patient lookup requires locating data aggregated from multiple sources, an organization's master patient (or person) index (MPI) relies on search algorithms to match records and retrieve data. Algorithms can be thought of as the formulas that determine what constitutes a match between the user's query and a database entry. The most sophisticated algorithms are capable of identifying a patient based on vague or incomplete information and returning a list of both exact and probable matches. Good algorithms can thus prevent duplicate records by identifying with reasonable accuracy existing records created by a keying error during data entry.

HIM staff play a critical role in maintaining accurate patient information. And as healthcare organizations are changing, the roles of the HIM professionals are expanding.[1] They are being asked to provide more data—and even better data—for both clinical and administrative decisions. Effective management of MPI is at the core of ensuring data quality throughout the enterprise.

The Burden of Duplicates

In a Database of 1 Million Records, Even Slight Administrative Error Can Cost Organizations Millions

As databases became mainstream, organizations began to realize the need for data standardization in order to make sense of their enterprise data. Data warehousing aggregated data from an organization's separate systems into a centralized database in order to make more sophisticated data analyses possible. However, aggregating data from multiple sources raises its own issues. Data formats, data domains, and file formats may be different. Data may be stored in different places, in systems that are not compatible.

For these and other reasons, large organizations are prone to duplicate records. For example, when a patient receives care at more than one facility in the system—whether by random occurrence or as part of the medical management process—and no data, or incomplete data, are exchanged between the facilities, a patient may receive a new medical record number. A duplicate record is born.

In general, the term duplicate is used to describe multiple records within one facility that refer to the same patient. The term overlap is used to describe records that refer to one patient across two or more facilities. The term overlay refers to a situation in which multiple patients share one ID.[1]

Duplicate records are responsible for significant inefficiencies, as clinical data are associated with multiple medical records. Financial issues occur as well.

Duplicate patient records may increase administration costs by an additional $100–400 for each occurrence, and it may cost $6–10 per record to clean the duplicates.[2]

A database of 1,000,000 patient records with a conservative duplicate rate of 1 percent could create $1 million to $4 million of additional administrative costs. Anecdotal evidence indicates that systems suffer from duplicate rates as high as 15 percent.

When the number of duplicate records is that high, automated duplicate matching is the most efficient way to clean an MPI. Record matching performed by people is slow and prone to errors in comparison to computer matching.[3] The sophistication of today's algorithms allows for a match threshold that can exceed the performance of the human operators.

Notes

1. AHIMA MPI Task Force. "Building an Enterprise Master Person Index." *Journal of AHIMA* 75, no. 1 (2004): 56A–D.
2. Carnese, D. J., and B.H. Just. "Solving the Identity Management Problem." Paper presented at the Proceedings of Windows on Healthcare IV, Microsoft Healthcare Users Group, October 13, 1998.
3. Winkler, William E. "Record Linkage Software and Methods for Merging Administrative Lists." Bureau of the Census, Statistical Research Division, 2001. Available online at www.census.gov/srd/papers/pdf/rr2001-03.pdf.

Three Types of Search Algorithms

While search algorithms are designed in many different ways, three common approaches are prevalent. For HIM professionals concerned with keeping MPI quality high, the key differences in the three approaches lie in the types of results they return.

Deterministic

The deterministic approach is the method traditionally associated with data search. It seeks an exact match between the search string—the data used in the search process—and the possible data candidates in the database. A deterministic search for a patient with the last name Jones will return a list of all patient records containing the last name Jones. If the database contains a large number of patients with this name, the result of the search will be of debatable value.

Increasing the amount of data fed to the search algorithm by including additional data elements will improve the search results. For example, a search for patients with last name Jones, first name Mary, and with a birth date of July 4, 1984, will result in fewer patient records being returned to the user. The more data used for the search, the fewer patient records returned and the better the match.

However, a deterministic algorithm seeks only exact matches. Unless there is a perfect match between Jones in the search string and Jones in the database, no record is found. If the patient's last name is mistakenly recorded as Jons during data entry, a deterministic algorithm will ignore the record. Given the potential for data-entry error, this situation is a cause of concern.

Nondeterministic algorithms incorporate greater flexibility in their searches, capturing records that match the search string with a greater range and less precision.

Fuzzy

Fuzzy searches incorporate a degree of approximation. In addition to finding exact matches, fuzzy searches also identify near matches.[2] Consider the following misspellings of always:

alwasy alwyas alwats alway

The variations suffer from transposed, incorrect, and dropped characters. A fuzzy search algorithm parsing these spellings for a match on always will identify them as approximate matches to varying degrees. A number of sophisticated algorithms are at the core of fuzzy searching.[3,4,5]

How well do fuzzy algorithms work? Let's take a look at the following paragraph, widely circulated on the Internet:

> Aoccdrnig to a rscheearch at Cmabrigde Uinervtisy, it deosn't mttaer in waht oredr the ltteers in a wrod are, the olny iprmoetnt tihng is taht the frist and lsat ltteer be at the rghit pclae. The rset can be a toatl mses and you can sitll raed it wouthit porbelm. Tihs is bcuseae the huamn mnid deos not raed ervey lteter by istlef, but the wrod as a wlohe.[6]

Despite the jumbled letters, most readers understand the text in a single reading. The example illustrates the syntactic dimension that fuzzy algorithms address: word misspellings, spelling variants, character or digit transpositions, et cetera. Fuzzy algorithms perform approximate string comparisons between the search parameters and the target data.[7]

Incidentally, the human brain is very good at dealing with this type of problem. The brain handles words with multiple semantic dimensions and multiple meanings such as get without difficulty. Search algorithms typically refer to a data dictionary or thesaurus to take synonyms into consideration.

Probabilistic

Probabilistic searching brings an additional refinement to nondeterministic search methodology by taking into account the relative frequency with which data occurs within a given field.

For example, the US census of 1990 reported a population of 248,709,873.[8] Of these, there were 2,502,021 people named Smith, or 1.006 percent of the US population. Fewer than 2,487 people had the last name Aarhus, approximately 0.001 percent of the population. There were only five people with the last name Gudea. This means that the search algorithm is more likely to find a match on Smith than on Aarhus or Gudea. However, a reported possible match on Aarhus or Gudea has a much higher probability of being correct. A possible match on an uncommon name carries more relevance than a possible match on a common name.

In any given database, similar frequency counts can be performed against any field in the database.[9,10] A weight table can store the frequency of occurrence of the last name, or any other string in the database. The search algorithm is then tuned to account for these relative weights and rank probability for matches accordingly.

Record Matching in HIE

Healthcare's Most Probable: Probabilistic

Health information exchange (HIE) among unaffiliated healthcare organizations raises the bar on MPI quality and record matching. The success of data exchange networks will rest solidly on algorithms that return the best possible matches.

Deterministic matching might be a viable search method if a national unique patient identifier were established. National ID numbers would simplify the exchange of an individual's health information by creating a single, unequivocal health identifier for each citizen. There is support for this option—especially among those who see it streamlining a national health information infrastructure—but there is as much or more resistance within healthcare, in Congress, and in the general population. It is likely that the state of patient identity will remain complex enough to require more sophistication than deterministic search methods offer.

Probabilistic algorithms thus appear to be healthcare's most likely solution, given their ability to identify and rank both exact and probable matches. Large healthcare organizations are successfully using the method to manage their enterprise MPIs, and health information exchange demonstration projects are employing the algorithms, also. Connecting for Health's record locator service demonstration project uses probablistic matching, and it is likely that the proposals for the Office of

the National Coordinator for Health Information Technology RHIO demonstration project will do the same.

Privacy and Probable Matches
Probable matches raise privacy issues when healthcare organizations share records. If clinic A queries the network for records belonging to Mary Jones, it may learn that clinics B and C hold probable—though not exact—matches. Should clinic A be able to view those matches?
The exchange's privacy policy, for example, could require that the search alert clinic A to the existence of probable matches but request additional data to better identify the match before proceeding. Any solution, however, must strike an appropriate balance between accessibility and privacy. This is just one more challenge in regional data exchange with a role for HIM expertise.

Upcoming Materials from AHIMA
Look for more on record matching in data sharing networks, including roles for HIM professionals, from AHIMA's Patient Identification in RHIOs work group. In early 2006, the group will publish the results of its work, which will include a practice brief on operational RHIOs and record linkage methods, sample job descriptions for a record linkage professional in an HIE setting, a comparative analysis of patient identification methodologies by vendor, and a glossary of terms related to patient identification in HIE settings. The material will appear in the Journal and online in the FORE Library: HIM Body of Knowledge.

—Editors

Which Algorithm Is Best?

Deterministic search may seem to be the most preferable approach for MPI management, as it relies on precise matches. Yet, as it has been demonstrated, this strict match/nonmatch approach can be too limiting. The nondeteministic algorithms introduce a continuum of matches, from more to less precise.

Consider the following street address information:

1234 ½	No. Maine Rd.	Apt #6
1234 1/ 2	N. Maine Road	Apt. 6
1234 ½	N Maine Rd	Apt 6
1234 1/ 2	N Maine Rd	#6
1234½	Maine Rd	No 6
1234 1/2	N. Main Road	Apt. 6
1234½	Maine Road	Apt #6

As a quick scan indicates, the addresses are very similar. They seem so similar that one can argue with varying degrees of confidence that in fact they express the same address.

The small variations in street suffix and prefix, geographical designator, and apartment number may be the result of various data entry mishaps. The variations have resulted in the creation of duplicate records within the database, all referring to the same person. This example illustrates the pitfalls of a free-form address field.[11] It also illustrates the limits of deterministic algorithms. Despite the apparent similarities of the addresses, a deterministic search will return at most just one of the seven records. The accuracy of deterministic algorithms is gauged to be in the range of 20 to 40 percent.[12]

Depending on the implementation, it can be asserted with a certain degree of confidence that the record matches found by fuzzy matching are true matches. Such algorithms can be designed to take into account transposition of digits in dates and numbers so that near matches contribute positively to the match score. A fuzzy algorithm may combine techniques such as phonetic name matching and approximate string comparison to account for typographical errors, character transpositions, dropped characters, inserted or doubled characters, spelling variations, and misspellings. Phonetic searching techniques will result in the algorithm reaching an accuracy of 50 to 80 percent.[13]

With an accuracy rate gauged to be 90 percent or higher, probabilistic matching has the greatest potential to maintain MPI integrity.[14] The examples centered on the US census show the value of relying on probabilistic techniques and weight tables to increase the relevance of the data identified by the search algorithm.

Although algorithms are capable of searching on highly complex terms, users must determine an appropriate search threshold. Complex algorithms require more time to complete a search because they are comparing more variables. Search performance is also affected by hardware, software, and database size. The more comparisons a given algorithm must perform the longer the response time and therefore perceived decrease in performance. It is up to the user to determine the optimal balance between precision and response time.

Notes

1. Wing, Paul, and Margaret H. Langelier. "The Future of HIM: Employer Insights into the Coming Decade of Rapid Change." *Journal of AHIMA* 75, no. 6 (June 2004): 28–32.
2. Girill, T. R., and Clement H. Luk. "Fuzzy Matching as a Retrieval-Enabling Technique for Digital Libraries." Available online at www.asis.org/midyear-96/girillpaper.html.
3. Winkler, William E. "Frequency-Based Matching in Fellegi-Sunter Model of Record Linkage." Bureau of the Census, Statistical Research Division, 2000. Available online at www.census.gov/srd/papers/pdf/rr2000-06.pdf.
4. Yancey, William E. "Frequency-Dependent Probability Measures for Record Linkage." Bureau of the Census, Statistical Research Division, 2000. Available online at www.census.gov/srd/papers/pdf/rr2000-07.pdf.
5. Lait, A. J., and Brian Randell. "An Assessment of Name Matching Algorithms." Department of Computing Science, University of Newcastle upon Tyne, UK, 1993. Available online at www.cs.ncl.ac.uk/~brian.randell/home.informal/Genealogy/NameMatching.pdf.
6. Davis, Matt. Web page. Available online at www.mrc-cbu.cam.ac.uk/personal/matt.davis/Cmabrigde.
7. Porter, Edward H., and William E. Winkler. "Approximate String Comparison and Its Effect on an Advanced Record Linkage System." Bureau of the Census, Statistical Research Division, 1997. Available online at www.census.gov/srd/papers/pdf/rr97-2.pdf.
8. US Census Bureau. "Frequently Occurring First Names and Surnames from the 1990 Census." Available online at www.census.gov/genealogy/www/freqnames.html.
9. Winkler. "Frequency-Based Matching."
10. Yancey. "Frequency-Dependent Probability Measures."
11. Christen, Peter, Tim Churches, and Alan Willmore. "A Probabilistic Fuzzy Geocoding System Based on a National Address File." Paper presented at the 17th Australian Computer Society Australian Joint Conference on Artificial Intelligence, Cairns, December 2004.
12. AHIMA MPI Task Force. "Building an Enterprise Master Person Index." J*ournal of AHIMA* 75, no. 1 (2004): 56A–D.
13. Ibid.
14. Ibid.

Sorin Gudea *(swgudea@email.phoenix.edu) teaches technology courses in the College of Information Systems and Technology at University of Phoenix, in the Southern California Campus. He is a PhD candidate in information science at Claremont Graduate University, in Claremont, California.*

Article citation:
Gudea, Sorin. "Deterministic, Probabilistic, or Fuzzy?: A Primer on the Search Algorithms that Drive MPI Quality." *Journal of AHIMA* 76, no.8