Peng, Yaohao; Mation, Lucas Ferreira

Working Paper

O desafio do pareamento de grandes bases de dados: Mapeamento de métodos de record linkage probabilístico e diagnóstico de sua viabilidade empírica

Texto para Discussão, No. 2420

Provided in Cooperation with:
Institute of Applied Economic Research (ipea), Brasília

Suggested Citation: Peng, Yaohao; Mation, Lucas Ferreira (2018) : O desafio do pareamento de grandes bases de dados: Mapeamento de métodos de record linkage probabilístico e diagnóstico de sua viabilidade empírica, Texto para Discussão, No. 2420, Instituto de Pesquisa Econômica Aplicada (IPEA), Brasília

This Version is available at:
https://hdl.handle.net/10419/211367

TEXTO PARA DISCUSSÃO

2420

O DESAFIO DO PAREAMENTO DE GRANDES BASES DE DADOS: MAPEAMENTO DE MÉTODOS DE *RECORD LINKAGE* PROBABILÍSTICO E DIAGNÓSTICO DE SUA VIABILIDADE EMPÍRICA

Peng Yaohao
Lucas Ferreira Mation

ipea

2420
TEXTO PARA DISCUSSÃO
Rio de Janeiro, outubro de 2018

O DESAFIO DO PAREAMENTO DE GRANDES BASES DE DADOS: MAPEAMENTO DE MÉTODOS DE *RECORD LINKAGE* PROBABILÍSTICO E DIAGNÓSTICO DE SUA VIABILIDADE EMPÍRICA

Peng Yaohao[1]
Lucas Ferreira Mation[2]

1. Pesquisador do Programa de Pesquisa para o Desenvolvimento Nacional (PNPD) na Assessoria Técnica (Astec) do Ipea. *E-mail*: <peng.yaohao@ipea.gov.br>.
2. Técnico de planejamento e pesquisa na Astec/Ipea. *E-mail*: <lucas.mation@ipea.gov.br>.

Texto para Discussão

Publicação seriada que divulga resultados de estudos e pesquisas em desenvolvimento pelo Ipea com o objetivo de fomentar o debate e oferecer subsídios à formulação e avaliação de políticas públicas.

Texto para discussão / Instituto de Pesquisa Econômica Aplicada.- Brasília : Rio de Janeiro : Ipea , 1990-

ISSN 1415-4765

1.Brasil. 2.Aspectos Econômicos. 3.Aspectos Sociais. I. Instituto de Pesquisa Econômica Aplicada.

CDD 330.908

JEL: C52; C55; C65; C80; C88.

SUMÁRIO

SINOPSE

Este trabalho verificou o desempenho preditivo de algoritmos de pareamento de registros (*record linkage* – RL) probabilístico para a integração de bases de dados reais de grande porte, avaliando os efeitos da definição das variáveis de *blocking* (blocagem ou indexação), de funções de distanciamento de *strings* (sequência de caracteres) e de algoritmos de pareamento fonético em relação à qualidade das previsões e à complexidade computacional. Realizou-se um levantamento bibliográfico dos principais métodos de RL determinístico e probabilístico, bem como de avanços recentes aliando técnicas de *machine learning* (aprendizado de máquinas) e principais pacotes e implementações disponíveis em linguagem *open-source* (código aberto) R.[1] Os resultados podem fornecer heurísticas para problemas de integração de registros administrativos em escala nacional e são de potencial valia para a formulação e a avaliação de políticas públicas.

Palavras-chave: pareamento de registros; *blocking*; registros administrativos; *Big Data*; R.

ABSTRACT

This paper verified the predictive performance of probabilistic record linkage algorithms for the integration big sized real databases, evaluating the effects of the blocking key definition, as well as string metric functions and phonetic code pairing algorithms with respect to the prediction's quality and computational complexity. A bibliographical survey of the main deterministic and probabilistic record linkage methods was carried out, as well as of recent advances combining machine learning techniques and main packages and implementations available in open-source R language. The results can provide heuristics for problems of administrative records integration at national level and have potential value for the formulation and evaluation of public policies.

Keywords: pairs linking; blocking; administrative records; Big Data; R.

1. R é uma linguagem de programação estruturada de acesso gratuito com foco em análises estatísticas e manipulação de dados.

1 INTRODUÇÃO

O Brasil é o quinto maior país do mundo em extensão territorial e em população absoluta, de forma que bases de dados de registros administrativos deste país são significativamente volumosas. Frequentemente, um único indivíduo está presente em várias das diversas bases de dados – por exemplo, uma mesma pessoa pode usufruir de serviços médicos, ser beneficiária de programas de assistência social, de distribuição de renda ou moradia, ser uma funcionária de carteira assinada, estar envolvida em um processo judicial, entre outras situações. Em razão da abundância de dados no cenário brasileiro, é desejável a agregação das inúmeras fontes existentes para uma base única na qual constem os registros dos indivíduos nas variadas esferas da vida social, o que pode ser de extrema valia para estudos relacionados à avaliação de políticas públicas e à análise das estruturas socioeconômicas, permitindo um mapeamento mais eficiente da realidade da nação e melhores fundamentos para a tomada de decisão nas esferas federal, estadual e municipal. Em se tratando de um país de extensão continental como o Brasil, a qualidade dos dados utilizados nesses estudos constitui-se em elemento primordial para a importância das análises realizadas.

Conforme aponta o estudo de Ferreira (2008), os registros administrativos do Brasil constituem uma vasta fonte de dados, mas ainda subexplorada. O autor elucida que o uso desses registros representa diversas vantagens em relação a técnicas tradicionais, como *survey* e entrevista, dado que a ausência de um entrevistador pode minimizar seu viés, as informações falsas e a omissão de dados, bem como permitir a obtenção de uma gama mais diversificada de informações, fornecendo uma visão mais completa das esferas da realidade social. Particularmente para registros individuais ou pequenas áreas, os registros administrativos podem inclusive prover maior detalhamento que pesquisas oficiais restritas a amostras em períodos específicos. Exemplos incluem o sistema do Departamento de Informática do Sistema Único de Saúde (DATASUS) do Ministério da Saúde, cadastros de programas de assistência social – como o Bolsa Família – e cadastros de aposentados e pensionistas do Instituto Nacional do Seguro Social (INSS) etc.

Essas diversas bases de dados, no entanto, podem não conversar bem entre si, o que se manifesta, por exemplo, na potencial heterogeneidade do armazenamento e da manutenção das informações de diferentes bases, de modo a dificultar a integração entre elas e a identificação do mesmo indivíduo em seus registros – um cidadão pode

ser beneficiário de um programa de assistência social, mas não ser assinalado como tal nos registros administrativos; em vez disso, pode constar nos registros o nome do chefe da família ou dos pais do beneficiário em questão. Ademais, eventuais erros de digitação ou equívocos de informação, como o indivíduo fornecer data de nascimento errada, tornam o cruzamento de registros em bases distintas uma tarefa desafiadora. Dada a grande população brasileira, mesmo uma ínfima margem de erro nesse sentido pode acarretar desvios de grande magnitude durante o processamento dos dados para estudos e pesquisas.

Assim, encontrar uma maneira de realizar o pareamento de registros (também conhecido como *record linkage* – RL) de forma eficiente, tanto em termos de qualidade do cruzamento quanto de viabilidade temporal, é uma questão pertinente. Este *Texto para Discussão* busca registrar as técnicas desenvolvidas pela literatura científica especializada em relação a essa tarefa, apresentando boas práticas em nível internacional e a viabilidade destas em aplicações reais, notadamente em bases de dados de grande extensão. São detalhadas as principais classes do RL e os seus modelos clássicos, bem como avanços recentes na produção científica correlata e ferramentas computacionais de prático manejo e escalabilidade que podem ser exploradas para a integração dos registros administrativos brasileiros.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Definição

O problema do RL consiste basicamente em se realizar pareamento de registros de várias bases de dados distintas, combinando informações de cada uma delas relativas a uma única unidade de observação – por exemplo, uma mesma pessoa pode ter registros em plataformas diferentes, como cadastros de programas de assistência social, serviços médicos, vínculos empregatícios etc. Por vezes, é pertinente realizar a unificação e integração das inúmeras bases de dados, concatenando as informações relativas ao mesmo indivíduo, de modo que não deixem de ser pareadas (falso negativo), ao mesmo tempo que informações de pessoas diferentes não sejam combinadas como se fossem provenientes da mesma pessoa (falso positivo).

O RL é relevante para diversos contextos em variados campos de estudo e áreas do conhecimento, tais como:

- monitoramento de serviços médicos como histórico de internações, medicamentos consumidos e intervenções cirúrgicas com registros por pacientes (Clifton *et al.*, 2004; Button *et al.*, 2011);
- mapeamento de estrelas e constelações na astronomia (Budavári e Loredo, 2015);
- cruzamento de dados de censo histórico e registros administrativos (Winkler, 2006; Jutte, Roos e Brownell, 2011; Richards *et al.*, 2014);
- desenvolvimento eficiente de políticas públicas, com base no cruzamento de dados relativos a contribuições tributárias e beneficiários de programas de assistência governamental (Kelman, Bass e Holman, 2002);
- identificação de fraudes e/ou atividades criminosas (Wang, Chen e Atabakhsh, 2004; Phua *et al.*, 2012); e
- detecção de plágio ou de páginas na *web* com conteúdo similar (Murugesan *et al.*, 2010).

Para um compêndio da aplicação de métodos de RL no tratamento de dados de extensão nacional, ver Gill (2001).

Não raramente, as diversas bases de dados não possuem um identificador único que permita que os batimentos sejam realizados de maneira imediata, requerendo por vezes técnicas mais apuradas para eliminar potenciais não pares e abarcar o máximo possível de pares verdadeiros. Especificamente, é comum a ocorrência de associações não bijetoras – ou seja, um indivíduo registrado em uma base com várias associações em outras, como o histórico de imigração ou a atualização de escolaridade de alguém. A grande diversidade de fontes de informação culmina na existência de múltiplas bases de dados, cada qual com suas potenciais especificidades, o que aumenta o desafio do RL.

O RL possui vários nomes alternativos, a depender da área de aplicação e da literatura específica, como *merge/purge problem e entity resolution*. De maneira frequente, ele é associado a um problema específico denominado deduplicação, termo que remete ao pareamento que busca suprimir registros repetidos.

É fácil notar que, com a presença de um identificador único para cada indivíduo em diversas bases de dados, o pareamento pode ser facilmente realizado ao se cruzar esse identificador em todas as bases nas quais é disponível, bastando fazer um *merge* (junção de duas ou mais bases de dados em uma única). Todavia, o desafio surge quando esse identificador não existe ou possui qualidade comprometida por erros de digitação, inconsistência nas diferentes bases de dados ou mesmo por questões de sigilo. Dessa maneira, faz-se necessário o desenvolvimento de técnicas de RL que permitam contornar esses entraves.

Essas técnicas se dividem em duas grandes classes: determinística e probabilística (Richards *et al.*, 2014). Ambas são consideradas abordagens iterativas. Após o pareamento inicial ser realizado, em geral toma-se uma amostra aleatória dos resultados preditos a fim de garantir que o algoritmo esteja funcionando conforme o esperado e identificar eventuais anomalias ou insuficiências mediante a avaliação comparada com métricas conhecidas.

2.2 RL determinístico

A abordagem determinística avalia se os pares de registros concordam ou discordam em um determinado conjunto de indexadores, de modo que a concordância precisa ser total para que dois registros sejam considerados como um *match* (um par). O pareamento determinístico pode ser feito em uma ou várias etapas. Em uma etapa única, comparam-se todos os registros de uma vez no conjunto dos indexadores, de forma que um *match* exige que os dois registros sejam completamente iguais em todos os indexadores; mais ainda, demanda que nenhum outro *match* tenha os mesmos valores deste batimento específico, fazendo com que todos os pares de registros sejam unicamente definidos para que não haja indivíduos duplicados. Já em um *framework* de várias etapas, por sua vez, os registros são combinados em uma série de estágios progressivamente menos restritivos em que os pares de registros que não atendem a uma primeira rodada de critérios de correspondência são passados para uma segunda rodada para comparações adicionais. Se um par de registros obedece aos critérios de qualquer uma das fases, considera-se que estes registros formam um *match*. A taxonomia para as duas estratégias de RL determinístico é constituída pelas classes "determinístico exato", para o pareamento por uma etapa, pois exige correspondência exata em todos os identificadores; e "determinístico aproximado", porque a correspondência exata não

precisa ocorrer em todos os indexadores possíveis, sendo suficiente que seja idêntica em uma das rodadas de avaliação.

Em situações em que identificadores estão disponíveis, mas são sigilosos ou sensíveis por alguma razão, o RL determinístico pode ser aplicado após uma etapa de criptografia que garanta a privacidade da informação. Técnicas como os filtros de Bloom são comumente empregadas com o propósito de assegurar a confidencialidade dos indivíduos, conforme elucidado por Schneier (2007) e Quantin *et al.* (1998).

A abordagem determinística produz bons resultados quando existe um conjunto de atributos que permitem uma chave de ligação formada por atributos precisos, estáveis ao longo do tempo e presentes em todas as bases de dados envolvidas. O uso da estratégia aproximada (várias etapas) provê flexibilidade ao pareamento, porém a elaboração do conjunto de regras que define os critérios usados para as rodadas e o número delas pode ser desafiadora, uma vez que está condicionada às características das bases de dados que se quer lincar, além da ausência de consenso acerca da elaboração dessas regras e de heurísticas empiricamente superiores ou inferiores em termos de qualidade preditiva.

O problema geral do RL exige a comparação dos registros de duas bases A e B de tamanhos η_A e η_B, respectivamente, sendo necessárias, assim, $\eta_A \cdot \eta_B$ combinações dois a dois. Dessa forma, em uma aplicação real que envolva o cruzamento de duas bases com 100 mil registros cada, seriam necessárias 10^{10} = 10 bilhões de comparações, de modo que essa abordagem torna-se inviável em bases de dados grandes. Ademais, a abordagem determinística ignora o fato de o poder discriminatório de alguns indexadores ou certos valores específicos não ser distribuído uniformemente entre estes – por exemplo, a chance de o mês de nascimento de dois indivíduos diferentes coincidir por acaso é de 1/12, enquanto a chance de duas pessoas distintas se chamarem José é significantemente menor, mesmo sendo este um nome próprio bastante recorrente. Assim, foram desenvolvidas estratégias probabilísticas para avaliar: *i)* o poder discriminatório de cada identificador; e *ii)* a probabilidade de dois registros serem verdadeiros sobre a concordância ou discordância a respeito dos vários identificadores. Alguns resultados empíricos mostram que os resultados do RL determinístico são piores que os do probabilístico (Christen e Goiser, 2007).

2.3 RL probabilístico

Tendo em mente as limitações da estratégia determinística, foram desenvolvidas estratégias probabilísticas para atacar o problema do RL, levando-se em consideração fatores negligenciados, como o poder discriminatório "marginal" de cada indexador, de modo que dois registros de bases diferentes serão considerados como um *match* ou não conforme métricas enunciadas em termos de probabilidade e a concordância ou discordância sobre os indexadores utilizados.

Conforme ilustram Sayers *et al.* (2015) em um exemplo rápido, a abordagem probabilística permite que o vetor de similaridade não seja binário – ou seja: 1, se os registros forem idênticos, e 0, caso contrário; em vez disso, as informações podem ser parcialmente iguais, de acordo com alguma métrica de distância. Uma vez que atributos recorrentes em distintas bases de dados, como nome, sobrenome, endereço residencial, data de nascimento etc., podem conter imprecisões ou inconsistências decorrentes de erros de digitação, abreviações ou valores *missing* (faltantes), a chance de se perderem *matches* verdadeiros com a abordagem determinística é muito grande, pois considera-se que "Paulo Silva" e "Paulio Silba" são pessoas diferentes, assim como "Paulo Silva" e "Ferdinando Schweinsteiger" são igualmente diferentes, no sentido de não serem idênticos. Uma inspeção visual ingênua, porém, é capaz de diagnosticar que o primeiro par de nomes possui muito mais chances de fazer referência à mesma pessoa que o segundo. Em uma estratégia determinística, os dois pares seriam descartados como *matches*, enquanto pela ótica probabilística nenhum deles é totalmente igual, tampouco são completamente desiguais, mas parcialmente iguais em intensidades diferentes, de maneira que a similaridade relativa é muito maior no primeiro par que no segundo. Logo, um par será considerado um *match* caso os indexadores sejam predominantemente iguais entre si, e um *não match* caso sejam predominantemente diferentes entre si.

O primeiro estudo de RL probabilístico remete ao trabalho de Newcombe (1967), o qual analisou a viabilidade da aplicação dessa estratégia mediante inspeção visual da frequência relativa de acertos em pares conhecidos *a priori* como verdadeiros. Posteriormente, Nathan (1967) e Tepping (1968) fornecem os fundamentos teóricos dessa abordagem. Mas o principal trabalho da temática é de Fellegi e Sunter (1969), que propuseram um modelo relativamente simples, com desempenho satisfatório, que permanece até os dias de hoje como o principal modelo para o RL probabilístico, em

especial para bases de dados de grande porte, dada a baixa exigência computacional para a grande parte das técnicas alternativas propostas na literatura.

Quando se pretende cruzar bases de dados de grande porte (como o Cadastro Único e o Programa Bolsa Família), considerar o pareamento para todo o produto cartesiano das variáveis disponíveis é computacionalmente inviável, porque o número de observações em cada base pode ultrapassar a casa das dezenas, ou mesmo das centenas de milhões. Sob essas circunstâncias, a literatura especializada sugere o uso de técnicas que reduzam o espaço de comparação apenas para registros que atendam a determinados critérios básicos, as quais recebem o nome de *blocking* (blocagem ou indexação). A ideia do *blocking* é segmentar as bases originais em subconjuntos menores com uma ou mais características idênticas – por exemplo, assumir que *matches* verdadeiros possuem o mesmo Cadastro de Pessoa Física (CPF), sexo ou nome da mãe. Os pares de registros que não obedecem aos critérios de correspondência especificados no *blocking* são prontamente classificados como *não matches*, dispensando análises adicionais. Esse artifício permite reduzir substancialmente o número de comparações a serem realizadas, e são primordiais para fazer do RL em grandes bases um problema factível. Considerando o *trade-off* (dilema de escolha) entre eficiência computacional e qualidade da classificação, tornar o problema resolvível em uma escala de tempo razoável tende a ter prevalência sobre potenciais incrementos na acurácia.

No modelo de Fellegi e Sunter (1969), cada par de potencial *match* filtrado do *blocking* é comparado com todo identificador disponível, fornecendo um padrão de similaridade (ou dissimilaridade). O peso designado para a similaridade em cada identificador é mensurado como uma razão de verossimilhanças entre a probabilidade de o identificador ser igual, dado que o par é um *match* verdadeiro, e a de ser igual por acaso, mesmo o par não sendo um *match*.

Para um conjunto de identificadores $\boldsymbol{\gamma} = \{\gamma_1, \gamma_2, \ldots, \gamma_p\}$ e um conjunto de η pares a serem comparados após o *blocking*, a probabilidade $m_j = P(\gamma_{ij_A} = \gamma_{ij_B}|$ o par *j* é um *match*), *i* = 1, 2, ..., *p*; *j* = 1, 2, ..., *n* é chamada de probabilidade-m (*match*), e a probabilidade $m_j = P(\gamma_{ij_A} = \gamma_{ij_B}|$ o par não é um *match*), *i* = 1, 2, ..., *p*; *j* = 1, 2, ..., *n* é chamada de probabilidade-u (*unmatch*). Em contextos mais gerais, a primeira é equivalente à sensitividade e a segunda, ao complementar da especificidade.

A probabilidade-m pode ser estimada retirando-se uma amostra aleatória de pares de registros do espaço de comparação e computando-se a frequência relativa de pares de registros iguais em um *i*-ésimo identificador arbitrário quando são classificados manualmente, a partir de uma inspeção visual, como *matches* verdadeiros. Já a probabilidade-u pode ser estimada observando-se a frequência relativa de pares de registros iguais no *i*-ésimo identificador meramente por acaso, de onde assume-se *a priori* uma distribuição de probabilidades para o identificador em questão (por exemplo, é possível admitir simploriamente a distribuição uniforme, de maneira que a probabilidade para o sexo seja de 1/2, de 1/12 para o mês de nascimento etc.).

As probabilidades-u podem ser definidas especificamente para dado identificador com base na frequência de cada valor observado e na probabilidade de concordância por acaso – é razoável supor que a probabilidade de concordância por acaso do sobrenome "Silva" seja significantemente maior que a do sobrenome "Salustiano"; portanto, o peso da concordância neste último deve ser maior que naquele. De todo modo, independentemente de se ponderar pela frequência relativa ou se assumir uma distribuição uniforme, como em geral não se sabe se um par de registros é um *match* ou não, as probabilidades-m e u precisam ser estimadas a partir de uma base conhecida e bem consolidada.

Sob a premissa de independência condicional, cada par de registros $j_A \in \{1, \ldots, n\}$ e $j_B \in \{1, \ldots, n\}, j_A \neq j_B$ receberá um peso de concordância (*matching weight*), expresso pela razão entre as probabilidades-m e u. Analogamente, o peso de discordância será a razão entre os complementares dessas probabilidades. Por conveniência matemática, esses pesos são frequentemente dados em termos da soma de um logaritmo, a saber:

$$wm_j = \sum_{i=1}^{p} \log_2 \left(\frac{P(\gamma_{ij_A} = \gamma_{ij_B} | \text{o par } j \text{ é um } match)}{P(\gamma_{ij_A} = \gamma_{ij_B} | \text{o par } j \text{ não é um match})} \right)$$

$$wu_j = \sum_{i=1}^{p} \log_2 \left(\frac{P(\gamma_{ij_A} = \gamma_{ij_B} | \text{o par } j \text{ é um } match)}{P(\gamma_{ij_A} = \gamma_{ij_B} | \text{o par } j \text{ não é um } match)} \right)$$

Dada a similaridade entre os identificadores γ_j, é possível expressar a probabilidade de o *j*-ésimo par ser um *match* verdadeiro – após manipulações das probabilidades condicionais

via teorema de Bayes $(P(A|B) = \frac{P(B|A)P(A)}{P(B)})$, obtém-se a expressão da probabilidade *a posteriori* de um par de registros *j* ser um *match*, condicionada ao pareamento dos identificadores γ_j, dada por:

$$P(match_j) = \frac{\frac{P(match_j | \gamma_{j_A} = \gamma_{j_B})}{P(\text{não } match_j | \gamma_{j_A} = \gamma_{j_B})}}{1 + \frac{P(match_j | \gamma_{j_A} = \gamma_{j_B})}{P(\text{não } match_j | \gamma_{j_A} = \gamma_{j_B})}}$$

Quanto mais $P(match_j)$ está próximo de 1, mais provável é que os registros sejam pertencentes à mesma entidade. O modelo de Fellegi e Sunter (1969) considera três zonas de classificação: *matches*, *não matches* e pares incertos. A atribuição dos pares de registros a cada uma dessas zonas depende do valor do *score* (peso) de concordância wm_j: são definidos dois níveis de *cutoff* (limite) – superior (cut_{sup}) e inferior (cut_{inf}) –, a fim de que pares com valores de $wm_j > cut_{sup}$ sejam classificados como *matches* e os com valores de $wm_j < cut_{inf}$ como *não matches*. Para os demais que ficam na zona cinzenta – como $cut_{inf} < wm_j < cut_{sup}$ –, os autores recomendam uma etapa adicional de inspeção manual. Ainda assim, esse modelo permanece bastante popular atualmente, sendo a principal abordagem utilizada em situações reais, mesmo envolvendo bases de dados grandes. Fair (2016), por exemplo, apresenta uma aplicação de tal modelo para bases de dados de saúde do Canadá pareando registros de uma base com mais de 500 mil deles.

Dado que a própria existência de uma zona de indefinição é extremamente incômoda, há estudos que sugerem mecanismos para estabelecer um *cutoff* único, que divide os pareamentos apenas entre *matches* e *não matches*. Cook *et al.* (2001) sugerem utilizar como *cutoff* a expressão:

$$\log_2\left(\frac{E}{(AB - E)}\right) - \log_2\left(\frac{P}{(1 - P)}\right)$$

Nela, *A* e *B* são o número de observações nas duas bases de dados a serem pareadas, *E* é o número esperado de *matches* entre as duas bases e $0 < P < 1$ é a probabilidade desejada de se encontrar um *match* verdadeiro. Para algoritmos com apenas um *cutoff*, calcula-se o *score* wm_j para cada par de registros, o qual será classificado como um

match se for maior ou igual ao limiar estabelecido e como um *não match* caso contrário. A critério do pesquisador, uma inspeção manual pode ser realizada para pares de registros cujo wm_j esteja dentro de um intervalo de confiança suficientemente próximo do valor-limite calculado que separa as duas classes.

Grannis *et al.* (2003), por sua vez, propõem uma abordagem baseada no algoritmo *expectation-maximization* (EM) para determinar um único *cutoff* que dispensaria a etapa de revisão manual. Os autores indicam que o algoritmo proposto auferiu resultados preditivos superiores em relação à abordagem determinística. O algoritmo EM proposto é uma das abordagens mais populares implementadas nos principais programas de RL probabilístico, como o *RecordLinkage* do R.

A depender da natureza da pesquisa, o *cutoff*, para considerar pares como *matches* ou *não matches*, pode ser ajustado a fim de que esses pares sejam mais "otimistas" ou "conservadores" em relação a encontrar *matches* verdadeiros. Em abordagem otimista, o valor-limite seria menor e, portanto, mais pares de registros tenderiam a ser classificados como *matches*; analogamente, uma abordagem mais conservadora faria com que o limite fosse maior, sendo, então, mais exigente com a classificação positiva como *match*. Essa escolha tende a impactar os índices de *performance* do algoritmo e evidencia um "cobertor curto" entre evitar conjuntamente falsos positivos e falsos negativos. Observa-se uma matriz de confusão típica no contexto do RL.

QUADRO 1
Modelo de matriz de confusão

Observado/predito	*Match*	*Não-Match*
Match	Matches detectados corretamente (Verdadeiros positivos - VP)	Não-matches detectados incorretamente (Falsos negativos - FN) (**Erro tipo II**)
Não-Match	Matches detectados incorretamente (Falsos positivos - FP) (**Erro tipo I**)	Não-matches detectados corretamente (Verdadeiros negativos - VN)

Elaboração dos autores.
Obs.: Figura cujos leiaute e textos não puderam ser padronizados e revisados em virtude das condições técnicas dos originais (nota do Editorial).

Nota-se que uma abordagem otimista permite que mais pares sejam classificados como *matches*. A tendência, portanto, é que vários pares que são de fato *não matches* acabarão sendo considerados como *matches*, enquanto dificilmente um *match* real será classificado como *não match*. Isso acaba fazendo com que a sensitividade seja alta e a

especificidade seja baixa. Para a abordagem conservadora, a relação se inverte: vários *matches* reais seriam classificados como *não matches*, aumentando a especificidade e diminuindo a sensitividade.

O modelo descrito possui diversas limitações, uma das mais proeminentes sendo o fato de estar condicionado à consistência dos identificadores usados para o *blocking*; mesmo quando são consistentes, ele descarta dois registros como um potencial *match* com identificadores diferentes, o que pode ser oriundo de um erro de digitação. Dusetzina *et al.* (2014), por exemplo, sugerem estender o *blocking* para pares de registros que não necessariamente são exatamente iguais, incluindo uma ponderação do valor de wm_j por uma métrica de distância de *strings* (sequência de caracteres). Entre as mais usuais, destacam-se as métricas de distância de Levenshtein, o índice de Jaro-Winkler, o coeficiente de Jaccard e o índice de Sørensen-Dice. Pita (2016) discute as diferenças entre essas métricas e realiza uma comparação entre elas. O critério para a similaridade exigida pelo *blocking* para esses índices também é arbitrário e de livre escolha para o pesquisador: Pita *et al.* (2017a) estabeleceram o corte de 91% de similaridade para o índice Sørensen-Dice (a ser abordado mais adiante). Essas métricas podem ser inclusive utilizadas para o cálculo de γ_j, o que levaria em consideração escalas de concordância parcial, já que γ_j não seria mais um vetor com valores binários.

A qualidade dos pareamentos realizados pode ser avaliada via histogramas – um diagrama com ampla concentração de valores com *scores* muito próximos de 0 ou de 1 é um sinal de boa classificação, pois simbolizaria uma grande proporção de números com baixo *score* (relativos aos pares que muito provavelmente não são *match*) e de valores com alto *score* (relativos aos pares que muito provavelmente são *match*). Em contrapartida, uma grande concentração de pares na zona cinzenta (faixa centrada em $P(match_j) = 0.5$) seria um indício de uma classificação defeituosa.

Uma amostra dos pares de registros avaliados pelo algoritmo de RL também pode ser retirada para se observar seu desempenho em relação às métricas usuais (acurácia, sensitividade etc.). Esse é um passo importante para diagnosticar potenciais dificuldades que o algoritmo está enfrentando e padrões recorrentes em classificações errôneas. Dusetzina *et al.* (2014) citam como exemplos o pareamento de irmãos gêmeos (pessoas diferentes que compartilham data de nascimento, sobrenome e nome dos pais), mulheres casadas (a mesma pessoa com nome de solteira e casada em diferentes

bases, além de endereços distintos) e membros de uma família (cônjuges e/ou filhos) que usam um documento do principal responsável, notoriamente em casos de famílias beneficiárias de programas de assistência social. Analisar a força e as fraquezas de um algoritmo de pareamento, além de permitir melhorar o seu desempenho, tende a ser muito importante para reutilizações dele no cruzamento de bases semelhantes ou iguais em contextos futuros.

Em síntese, as principais etapas do RL probabilístico podem ser resumidas conforme a seguir descrito.

1) Estimar as probabilidades-m e u para cada variável de identificador selecionada como *blocking* utilizando as frequências observadas de concordância e discordância (*match* idêntico ou alguma outra métrica de distanciamento de *strings*) em todos os pares de registros candidatos. O procedimento mais usual é utilizar o algoritmo EM para estimar as probabilidades enunciadas por Fellegi e Sunter (1969); esse procedimento é discutido com detalhes em Winkler (1988).
2) Definir uma métrica de similaridade para mensurar a concordância total ou relativa entre as variáveis a serem comparadas.
3) Usando as probabilidades estimadas no passo anterior, calcular os pesos de concordância wm_j e de discordância wu_j do *j*-ésimo par de registros com base na métrica de similaridade escolhida para cada variável.
4) Comparar o peso de concordância com os valores-limite de *cutoff* definidos com as informações do passo 1; um valor único de *cutoff* discrimina os pares entre *matches* e *não matches*, enquanto o uso de dois *cutoffs* deixa uma zona de inconclusão que requer uma inspeção manual posterior.

Caso as bases de dados a serem pareadas não sejam de qualidade – a exemplo de identificadores inconsistentes, número elevado de *missings* ou erros de digitação –, uma etapa preliminar de limpeza desses dados pode ser aplicada para aumentar a probabilidade de se encontrarem *matches* de registros, ampliando consequentemente a acurácia geral do algoritmo de pareamento, mediante a utilização de técnicas como padronização de datas e endereços, remoção de pontuação, codificação fonética, correspondência de nome próprio com apelidos comuns, entre outras (Newcombe *et al.*, 1983; Wajda e Roos, 1987; Quantin *et al.*, 1998; Gill, 2001; Churches *et al.*, 2002). Em contrapartida, Randall *et al.* (2013) apontam que a limpeza dos dados por si só representa um ônus significativo e uma melhoria tímida na qualidade do RL. Mais que isso, uma limpeza excessiva nos dados em bases

razoavelmente consistentes pode reduzir a variabilidade deles de modo a potencializar a propensão a classificações errôneas de *não matches* como *matches* (falso positivo/erro tipo I), o que acaba por comprometer o desempenho geral do pareamento. Por conseguinte, essa etapa adicional deve ser avaliada com cautela e executada caso a má qualidade dos dados seja notável o suficiente para que justifique o seu uso.

2.4 Indicadores de *performance* das classificações

Em geral, as classes descritas na matriz de confusão (quadro 1) são combinadas em indicadores para se avaliar a qualidade das classificações realizadas. As métricas mais comuns para a análise da qualidade dos pareamentos realizados são:

$$\text{Acurácia} = \frac{VP + VN}{VP + FP + VN + FN}$$

Reduction ratio = 1 - acurácia

$$\text{Precisão (}\textit{pairs quality}\text{)} = \frac{VP}{VP + FP}$$

$$\text{Sensitividade (}\textit{pairs completeness}\text{)} = \frac{VP}{VP + FN}$$

$$\text{Especificidade} = \frac{VN}{VN + FP}$$

$$\text{F1 } score = \frac{1}{\frac{1}{\text{Precisão}} + \frac{1}{\text{Sensitividade}}} = 2 \cdot \frac{\text{Precisão} \cdot \text{Sensitividade}}{\text{Precisão} + \text{Sensitividade}}$$

Conforme elucidam Vatsalan, Christen e Verykios (2013), a acurácia pode ser um indicador enganoso, uma vez que o contexto do RL geralmente envolve bases de dados não balanceadas, o que a tornaria muito elevada, mesmo classificando todos os pares como *não matches*. Por exemplo, em um caso com cem registros, entre os quais apenas cinco são *não matches*, uma previsão de que "não há correspondências" teria uma elevada acurácia de 95%, apesar de não ter previsto nenhum par verdadeiro.

A precisão, por sua vez, permite ver a proporção de verdadeiros positivos em relação aos positivos previstos, de modo que este indicador penalizará a ocorrência de

falsos positivos, mas não de falsos negativos – ou seja, é o indicador mais recomendado para casos em que o custo por um não par erroneamente classificado como par é maior que o de se deixar passar uma correspondência verdadeira. A especificidade funciona de modo similar para a classe de verdadeiros negativos.

Em contrapartida, a sensitividade penaliza a ocorrência de falsos negativos, sendo mais indicada para circunstâncias em que o interesse maior é não perder pares verdadeiros, mesmo pagando-se o preço de classificar informações de indivíduos distintos como pares.

Por fim, o *F1 score* é a média harmônica entre a precisão e a sensitividade, e fornece uma medida conservativa (por exemplo: tende para o valor menor entre os dois) entre os erros tipos I e II, de modo que tanto falsos positivos quanto falsos negativos façam com que o valor dessa medida diminua. Dessa forma, um valor elevado para o *F1 score* representa um ajuste satisfatório para a classe positiva (*matches*) e também para sua classe complementar (*não matches*). No contexto do RL, o *F1 score* é conhecido inclusive como índice Sørensen-Dice.

2.5 Desenvolvimentos recentes: *machine learning* e RL

Wilson (2011) demonstra que o modelo de Fellegi e Sunter (1969) é equivalente ao algoritmo Naïve Bayes, conhecido na literatura de *machine learning* (aprendizado de máquinas) por possuir uma das melhores *performances* em termos de processamento computacional. Há vastas evidências, porém, de que esse modelo provê uma qualidade de classificação/previsão pior que a maioria dos modelos consagrados em *machine learning*: especificamente no contexto do RL, Richards *et al.* (2014) comparam Naïve Bayes aos modelos de *machine learning k-nearest neighbours* (KNN – *k*-nésimo vizinho mais próximo) e *support vector machine* (máquina de vetores de suporte – SVM), os quais se mostraram bem superiores em termos preditivos.

Algoritmos de *machine learning* tipicamente demandam, no entanto, um tempo de processamento bem superior, de modo que o desempenho preditivo adicional pode acabar não compensando o tempo extra necessário para a execução do algoritmo. Para efeito de comparação, o modelo de Fellegi e Sunter (1969) – equivalente ao algoritmo

Naïve Bayes – possui complexidade $\mathcal{O}(\eta_1 \cdot \eta_2 \cdot p)$,[1] em que η_1 e η_2 são os números de observações nas bases de treinamento e p é o número de variáveis a serem pareadas, enquanto o algoritmo SVM depende da multiplicação de uma matriz *kernel*, responsável pelo mapeamento de não linearidades, e, portanto, possui complexidade $\mathcal{O}(\eta^3)$. O algoritmo de árvores de decisão, no entanto, tem complexidade $\mathcal{O}(p \cdot \eta \cdot \log(\eta))$ e poderia representar uma opção mais atrativa caso o ganho em velocidade não seja mitigado pela piora do desempenho preditivo. É bem sabido, entretanto, que árvores de decisão sofrem de alto nível de *overfitting* (sobreajuste), exigindo cautela em relação ao uso dessa técnica.

O método de resolução via algoritmo EM foi estendido para métodos não supervisionados em diversos estudos. Bhattacharya e Getoor (2005), por exemplo, usam alocação latente de Dirichlet (latent Dirichlet allocation – LDA) para realizar RL em bases reais de artigos científicos, pareando citações de autores comuns em diversos documentos. A abordagem não considera os registros como pares independentes; em vez disso, as decisões de pareamento são tomadas coletivamente, levando-se em conta as potenciais relações entre os documentos e as referências. Com base nisso, a amostragem de Gibbs foi utilizada para definir grupos de documentos que geralmente são analisados em conjunto.

Feigenbaum (2016) utiliza uma abordagem probit a fim de designar os *scores* de concordância para determinar os *matches* em potencial, fazendo comparações com diversas técnicas de *machine learning*: os resultados mostram que o modelo probit forneceu os melhores índices de eficiência e acurácia, superando os modelos logit, OLS (ordinary least squares, ou mínimos quadrados ordinários – MQO), SVM e *random forest* (floresta de decisão aleatória), além das abordagens por Soundex e da correspondência exata – esta apresentou os piores índices de *performance*, evidenciando a pertinência de se considerarem métricas de concordância parcial.

Fu *et al.* (2014) avaliam a similaridade dos pares de registros comparando um SVM com o método baseado em valores-limite (*thresholds*) e mostram que a abordagem fundamentada no *machine learning* oferece um aumento substancial na qualidade do pareamento. Estudos que aliam métodos de *machine learning* ao contexto do RL

1. Um algoritmo $\mathcal{A}(\eta)$ possui complexidade $\mathcal{O}(f(n))$ caso existam uma constante real $k \geq 0$ e um inteiro $\eta 0 \geq 0$ tal que $\mathcal{A}(\eta) \leq k \cdot f(n)$, $\forall\, \eta \geq \eta_0$.

incluem ainda Michelson e Knoblock (2006), os quais apresentam um algoritmo de aprendizagem para definir melhores predicados para serem usados como *blocking* – os resultados mostram que o algoritmo proposto conseguiu *reduction ratio* (índice de redução) superior a 98% em todas as bases de dados verificadas –, e Wang e Wang (2016), que agruparam dois SVMs iterativos para realizar a classificação. Para um experimento realizado com o pareamento de duas bases com 38.707 e 99.315 observações respectivamente, com dez grupos de predicados para *blocking*, o algoritmo proposto forneceu precisão, *recall* e *F1 score* superiores a 98% em todos os conjuntos testados.

3 O PROBLEMA DO RL NO BRASIL

O projeto Coorte de 100 Milhões (Pita *et al*., 2017b) foi desenvolvido pelo Centro de Integração de Dados e Conhecimentos para Saúde (Cidacs), da Fundação Oswaldo Cruz (Fiocruz), junto a pesquisadores da Universidade Federal da Bahia, com o intuito de estabelecer um cruzamento entre dados do sistema público de saúde brasileiro e registros administrativos, a fim de avaliar e monitorar a eficiência e os impactos de políticas públicas, em especial as de assistência social e distribuição da renda, na incidência de doenças. Cerca de 76 milhões de registros do Cadastro Único foram cruzados com pagamentos do Programa Bolsa Família e informações do Sistema Único de Saúde (SUS) – Sistema de Informações Hospitalares (SIH), Sistema Nacional de Atendimento Médico (Sinam) e Sistema de Informações sobre Mortalidade (SIM). O trabalho foi executado em Spark para acelerar o tempo de processamento computacional e tornar o algoritmo escalável para o elevado número de comparações que precisariam ser realizadas.

O RL também é aplicado para monitorar a eficácia do programa de Benefício de Prestação Continuada (BPC) da Lei Orgânica da Assistência Social (LOAS) na escola, o qual possui como objetivo garantir o acesso e a permanência no colégio das pessoas de 0 a 18 anos com deficiência, beneficiárias do BPC. A Empresa de Tecnologia e Informações da Previdência Social (Dataprev), órgão vinculado ao Ministério da Previdência Social, realiza periodicamente um pareamento entre os dados do Educacenso e do cadastro administrativo do BPC a fim de identificar o percentual de beneficiários com deficiência que estão na escola, para, assim, avaliar a eficiência do programa em relação à inclusão social. O pareamento foi feito utilizando-se como variáveis de *blocking* o nome do

beneficiário, sua data de nascimento e o nome da mãe, com o uso de correspondência exata e correspondência fonética.[2]

Os algoritmos de codificação fonética mais utilizados – como Soundex, New York State Immunization Information System (NYSIIS) e Metaphone – foram originalmente formulados para o inglês, levando-se em consideração as práticas de pronúncia, as contrações e os dígrafos desse idioma (mesmo entre o inglês britânico e o americano há diferenças relevantes). Dessa forma, conforme é de se esperar, algoritmos dessa natureza geram resultados ruins caso sejam aplicados diretamente para nomes brasileiros.

QUADRO 2
Distribuição fonética Soundex

Codificação SOUNDEX	
Código fonético	**Letras**
1	B,F,P,V
2	C, G, J, K, Q,S, X, Z—
3	D, T
4	L
5	M, N
6	R

Fonte: Pita (2016, p. 27).
Elaboração dos autores.
Obs.: Figura cujos leiaute e textos não puderam ser padronizados e revisados em virtude das condições técnicas dos originais (nota do Editorial).

Adaptações têm sido feitas para os nomes em português brasileiro como alternativas às versões feitas para o inglês, conforme exemplos a seguir.

1) BuscaBR:[3] proposto originalmente por Fred Jorge em 2007 e posteriormente remodelado por Marcos Rodrigues em 2010 e Gabriel Sobrinho em 2011.

2. Para mais informações, ver Nota Técnica nº 51/2013, publicada pelo Ministério da Educação (MEC), juntamente com a Secretaria de Educação Continuada, Alfabetização, Diversidade e Inclusão (SECADI) e a Diretoria de Políticas de Educação Especial (DPEE). Disponível em: <http://portal.mec.gov.br/index.php?option=com_docman&view=download&alias=13285-nt51-pare-bpc-2012-pdf&Itemid=30192>.
3. Disponível em: <https://github.com/JayrAlencar/buscaBR.js/; http://extremoconhecimento.blogspot.com.br/2013/02/implementando-algoritmo-buscabr-em-mysql.html>.

QUADRO 3
Distribuição fonética BuscaBR

Codificação BuscaBR			
Fonema	**Letras**	**Fonema**	**Letras**
I	I	M	N, RM, GM, MD, SM e terminação em AO
B	BR	N	NH
F	PH	P	P
G	GR, MG, NG, RG	S	Ç, X, TS, C, Z, RS
J	GE, GI, RJ, MJ, NJ	T	LT, TR, CT, RT, ST
K	Q, CA, CO, CU, C	V	W
L	LH	L	R

Fonte: Pita (2016, p. 28).
Elaboração dos autores.
Obs.: Figura cujos leiaute e textos não puderam ser padronizados e revisados em virtude das condições técnicas dos originais (nota do Editorial).

2) pt_metaphone: desenvolvido em 2008 pela prefeitura de Várzea Paulista.[4]

3) MTFN-MeTaPHoNe: adaptação do pt_metaphone desenvolvida por Ronie Uliana em 2009,[5] adequando-o para ficar mais "frouxo", de modo a focar apenas os fonemas que mais se destacam nas palavras, com o objetivo de abarcar, com o mesmo código fonético, uma variedade maior de nomes próprios. Com o custo de se aumentar o número de potenciais *matches* falsos positivos, o total de falsos negativos tende a diminuir, dado que teoricamente seria mais difícil que *não matches* tenham o mesmo código.

4) Metaphone-pt_BR: adaptação do pt_metaphone desenvolvida por Jordão e Rosa (2012).[6] Esse algoritmo realiza substituições com base em correspondências não apenas de letras ou dígrafos, mas também em expressões regulares, de modo que o número de substituições é muito maior e consegue abarcar diversas variações de uma mesma combinação de letras que produzem pronúncias diferentes, a depender da palavra.

4. Disponível em: <http://informatica.varzeapaulista.sp.gov.br/metaphone/>.
5. Disponível em: <https://github.com/ruliana/MTFN>.
6. Disponível em: <https://code.launchpad.net/metaphoneptbr>.

QUADRO 4
Distribuição fonética Metaphone-pt_BR

Codificação Metaphone_pt-BR			
Fonema	**Letras**	**Fonema**	**Letras**
R	CR, *R*,	Z	EX*
K	Q, CA, CO, CU, C, K, CHR	T	TH, T, T*
S	CE, CI, Ç, SS, SCE, SCI	F	PH
G	GA, GO, GU, GH	L	L*
M	M, N*,	2	R, R$, RR
J	GE, GI, GHE, GHI	3	NH
X	SCH, SH, CH, EXE, EXI,	1	R

Fonte: Pita (2016, p. 28).
Elaboração dos autores.
Obs.: Figura cujos leiaute e textos não puderam ser padronizados e revisados em virtude das condições técnicas dos originais (nota do Editorial).

Há evidências empíricas de que a utilização de algoritmos adaptados para o português ameniza problemas com erros de digitação para fonemas parecidos – por exemplo, "Souza" e "Sousa" (Pita, 2016). Porém, dado que uma mesma letra (ou até uma mesma combinação de letras) produz pronúncias diferentes, há inconsistências na transliteração feita por esses algoritmos. A título de exemplo, seguem listadas algumas situações envolvendo campos de nomes que poderiam ser problemáticas para a realização dos pareamentos:

- abreviações de nomes intermediários, com potencial de inclusão de pontos, como em "Adriano Almeida de Sousa" e "Adriano A. de Sousa";
- supressão de palavras de ligação (de, da etc.), como em "Maria das Neves Guedes e Cruz" e "Maria Neves Guedes Cruz";
- supressão de nomes intermediários, como em "Paloma Miranda Dutra Barros de Oliveira" para "Paloma Dutra de Oliveira";
- alterações no sobrenome de mulheres que adotaram o sobrenome do marido, como em "Larissa Silva Leite" para "Larissa Silva Leite Godoy";
- mudanças em algumas partes do nome: "Dulcinea Lino dos Santos" *versus* "Dulcineia dos Santos";
- "si" no início da palavra retorna S, mas no meio retorna Z, enquanto "ssi" retorna S, tornando o algoritmo vulnerável ao erro de digitação, de "Jacinto" para "Jasinto" ou de "Tomassini" para "Tomasini", por exemplo;

- "Alexandro" e "Aleksandro" retornam o mesmo código no algoritmo Metaphone-pt_BR (ALKSNDR), porém "Alex" retorna ALX e "Aleks" retorna ALKS;
- letras mudas como "Vitor" (VT2) e "Victor" (VKT2);
- erros de transposição, como "Demerval" (DMRVL) e "Dermeval" (DRMVL);
- consoantes repetidas, como "Felipe" (FLP) e "Fellipe" (FLLP) ou "Jeferson" (JFRSM) e "Jefferson" (JFFRSM); e
- o efeito da letra h no Metaphone parece ser obscuro, a exemplo de "Cristian" (KRSTM)/"Christian" (KSTM) e "Talita" (TLT)/"Thalita" (TLT)/"Talihta" (TL0T)/"Talitah" (TLT0).

4 *PERFORMANCE* X VELOCIDADE: MÉTODOS DE *BLOCKING*

Conforme discutido anteriormente, bases de dados de grande porte exigem uma filtragem prévia dos potenciais pares de registros que formam um *match* a fim de reduzir o número de comparações a serem realizadas. Em bases com milhões ou dezenas de milhões de observações, esse número estaria na casa dos trilhões – claramente inviável para problemas reais. Dessa forma, a escolha do método de *blocking* é um passo importante para a execução de um algoritmo de RL que consiga equilibrar o meio-termo entre a diminuição do número de comparações a serem feitas – como a quantidade de pares de registros filtrados como potenciais *matches* – e a parcimônia em segmentar o conjunto de dados original excessivamente, de modo que *matches* reais acabem sendo divididos em blocos distintos, e, portanto, sequer serão avaliados como um possível par. Esse *trade-off* entre a complexidade computacional exigida e o número de pares candidatos é discutido em Vatsalan, Christen, e Verykios (2013).

O método de *blocking* tradicional consiste em simplesmente agrupar todos os pares de registros que possuem o mesmo identificador (ou *key* – chave), de forma que apenas os que atendam a esse critério são comparados como potenciais *matches*. Esse mecanismo garante que as bases originais serão segmentadas em partições mutuamente excludentes. Sua intuição e implementação são bastante simples, porém possuem como grande limitação a incapacidade de lidar com erros tipográficos ou identificadores inconsistentes, ou seja, que não estejam presentes ou que tenham alta proporção de *missings* em alguma das bases de dados a serem pareadas. A fim de contornar esse problema, a literatura sugere o uso de funções de codificação como o Soundex e o Metaphone, de métricas de distanciamento de *strings* (para incluir concordâncias

parciais ou quase idênticas), ou, como recurso, a utilização de mais de um método de *blocking*. No entanto, os algoritmos de codificação fonética são, em quase sua totalidade, desenvolvidos para o inglês, e adaptações radicais precisam ser feitas para que se obtenham resultados satisfatórios para nomes de outro idioma, conforme ilustrado por estudos como os de Schnell, Bachteler e Bender (2004).

Para amplificar as chances de se identificarem *matches* verdadeiros que não são pareados pela etapa de *blocking*, em geral são feitos múltiplos passos para que os potenciais pares que não satisfaçam os critérios do primeiro *blocking* sejam automaticamente classificados como *não matches*. O uso de várias etapas de *blocking* tem se mostrado como uma alternativa válida para a captura de mais *matches* verdadeiros e, ao mesmo tempo, para a manutenção do número de comparações a serem feitas em uma escala razoavelmente viável, desde que as variáveis escolhidas como *keys* sejam consistentes e com poucos *missings*.

Outro procedimento que tem se mostrado eficiente é o uso de "predicados" como chaves de *blocking*. No contexto do RL, predicados são entendidos como conjunções lógicas concatenadas com disjunções, ou seja, uma expressão lógica do tipo $\{v_1$ E v_2 OU (v_3) OU $(v_4$ E $v_5)\}$. Conforme elucidado por Hernández e Stolfo (1998) e Koudas, Sarawagi e Srivastava (2006), a abordagem do uso de predicados pode ajudar a atenuar o efeito nocivo de erros de digitação nas variáveis de *blocking* – por exemplo, para um par de registros com nomes Ítalo Arraes e Ítalo Arrais, um *blocking* tradicional por sobrenome eliminaria esse par como um potencial *match*; no entanto, caso o par tenha o mesmo CPF ou data de nascimento, a chance de ser de fato um *match* verdadeiro é muito grande. Assim, utilizando-se um predicado como "sobrenome OU data de nascimento", o par se qualificaria como potencial *match* em vez de ser prontamente eliminado do espaço de comparações posteriores. Outro problema que poderia ser amenizado com os predicados é a definição de blocos demasiadamente grandes pela existência de atributos pouco discriminatórios, como usar "sexo" como *blocking key* – isto separaria as bases em dois blocos, que poderiam ainda ser grandes demais para serem comparadas pelo produto cartesiano; nomes comuns como João ou Maria poderiam também definir blocos de tamanho inviável para comparação. O uso de predicados inspira cautela na medida em que tende a exercer influência decisiva no total de *matches* gerados e no tempo de processamento, a depender do quão restrita é a expressão usada. Predicados muito restritivos refinam a busca para vários blocos de tamanho pequeno,

mas tendem a pegar poucos pares, enquanto os mais gerais podem identificar mais *matches*, porém a um custo de tempo bem maior e de baixa frequência relativa.

As técnicas citadas podem ser combinadas em aplicações reais. Fu *et al.* (2014), por exemplo, utilizaram três passos de *blocking* por predicado para o pareamento de seis bases do censo histórico no Reino Unido entre 1851 e 1901, quais sejam: "sobrenome OU (sexo E distrito do censo)", "nome OU endereço" e "sobrenome OU código industrial". Diferentes métricas de distanciamento foram aplicadas para variáveis distintas (sexo foi avaliado por correspondência exata, enquanto nome e sobrenome foram analisados por *q-grams* e distância de Jaccard). Os resultados mostraram que esse procedimento diminuiu significativamente os *matches* com *link* múltiplo – isto é, um registro da base A ser pareada com mais de um registro (diferentes entre si) da base B.

Christen (2012) apresenta um compêndio de técnicas de *blocking* mais usadas pela literatura, explicando brevemente o funcionamento de cada uma e comparando o desempenho delas em relação à qualidade de classificação e ao tempo de processamento. As técnicas a seguir foram as avaliadas.

1) *Blocking* tradicional, abordado anteriormente.
2) *Blocking* por vizinhança ordenada, no qual as bases de dados são organizadas em relação a uma chave, em que uma janela móvel de tamanho fixo é movida na lista de pares de registros ordenada, e as comparações são feitas apenas para os pares dentro dessa janela. Três extensões dessa abordagem são consideradas (ordenamento padrão, ordenamento invertido e ordenamento adaptativo).
3) *Blocking* por *q-grams*, no qual uma *string* é dividida em *substrings* sequenciais de tamanho *q*, a fim de permitir que um mesmo registro possa ser inserido em mais de um bloco.
4) *Blocking* por *suffix-array*, abordagem similar ao *q-grams* na qual os blocos são definidos a partir de uma lista ordenada com os sufixos dos identificadores. Uma extensão adaptativa ("robusta") também foi considerada.
5) Clusterização por cobertura (*canopy clustering*), abordagem similar ao *q-grams* na qual são gerados *clusters* (coberturas) baseados em alguma métrica de similaridade, de modo que um mesmo registro possa estar em vários *clusters* simultaneamente, e registros dentro de cada um deles serão comparados entre si. Foram consideradas as extensões baseadas em *thresholds* e vizinhança mais próxima (*nearest neighbour*).

6) *Blocking* baseado em *string map*, no qual as chaves são mapeadas para um espaço euclidiano de dimensão superior em que as similaridades relativas entre os registros sejam mantidas, e os potenciais *matches* são determinados via clusterização nesse espaço de dimensão maior. Para essa abordagem também foram consideradas as extensões baseadas em *thresholds* e vizinhança mais próxima (*nearest neighbour*).

Os resultados apresentados por Christen (2012) indicam que o método de *blocking* tradicional obteve os melhores resultados entre as abordagens comparadas, apresentando razoáveis índices de *performance* e baixo tempo médio de processamento. As abordagens de *blocking* por vizinhança ordenada adaptativa e clusterização por cobertura obtiveram resultados semelhantes ou ligeiramente superiores ao *blocking* tradicional, porém com tempos de execução muito superiores. Por fim, especificamente, a técnica mais lenta (*q-grams*) apresentou um tempo médio de execução mais de 2 milhões de vezes maior que o *blocking* tradicional.

5 PACOTES NO R

Os pacotes desenvolvidos no software livre R contendo implementações de funções relativas ao problema do RL se encontram a seguir.

1) *RecordLinkage*: implementação do modelo de Fellegi e Sunter (1969) pelo algoritmo EM, que é mais rápido que a maioria dos algoritmos de *machine learning*. Para o cômputo dos pesos de concordância de cada par de registros, possui implementadas as variantes de Contiero *et al.* (2005) – peso padronizado entre 0 e 1 – e Winkler (1988) – peso como soma dos logaritmos, portanto com domínio $(-\infty,\infty)$.[7] A função *trainSupv* permite alterar o classificador supervisionado para diversas técnicas de *machine learning*: SVM, árvores de decisão, *bagging*, *bumping* (similar ao *bagging*, usando a decisão do melhor modelo no treinamento em vez de combinar como voto majoritário), rede neural de uma camada e AdaBoost. O pacote ainda possui a opção de personalizar as funções de distanciamento de *strings* e de pareamento fonético. Não aceita o formato "*Date*" para o pareamento, de modo que variáveis de data devem ser convertidas para *character* (formato de caracteres).

7. Para uma discussão comparativa entre essas implementações, bem como com outros algoritmos de *machine learning* aplicados ao contexto do RL, ver Sariyar, Borg e Pommerening (2009).

2) *fastLink*: a descrição do algoritmo implementado está em Enamorado, Fifield e Imai (2017). Consiste em um modelo de Fellegi e Sunter (1969) que permite dados missing e a inserção de informação auxiliar – cálculo das estimativas *a priori* pela função *calcMoversPriors*, que podem ser usadas como os valores iniciais do argumento *priors.obj* da função *fastLink* (Fellegi-Sunter original) ou do argumento *prior.lambda* da função *emlinkMARmov* (com dados *missing*). Apesar de ser construída para processamento paralelizado (busca automaticamente núcleos disponíveis), a função transforma internamente os dados em uma matriz de frequências com o número de concordâncias para cada combinação possível de variáveis. Por exemplo, para pareamentos entre bases com p variáveis, a matriz conteria o número de vezes que cada uma das 3^p combinações das três classes (concordância/discordância/*missing*) foi verificada entre as bases. A saber, um teste piloto realizado com duas bases com 50 mil observações cada exigiu um tempo de processamento superior a cinco dias.

3) *blink*: abordagem bayesiana para dados categóricos e em *string*; permite definição de funções de distância/similaridade pelo usuário. A implementação do modelo está em Steorts *et al.* (2015). Usa um algoritmo probabilístico (amostragem de Gibbs, um caso de Markov Chain Monte Carlo) como alternativa ao EM. Possui a limitação de exigir que o pareamento seja realizado entre um vetor de *strings* e um de variáveis categóricas.

4) *corlink*: tem funções para RL, imputação de *missings* para identificadores do *blocking* e modelagem de padrões de correlação entre os pares de registros. Também baseado no algoritmo EM. Possui variações ao modelo de Fellegi e Sunter (1969) – argumento *alg* da função *linkd* –, porém exige que o *input* da função seja uma matriz binária de frequência com o número de vezes que cada padrão de concordância se verificou, o que exige preparação prévia dos dados, processo que seria muito custoso computacionalmente caso levasse em consideração concordância parcial de *strings*. Isto exigiria comparação prévia de todo o produto cartesiano, procedimento inviável para grandes bases de dados.

5) *ludic*: possui o modelo de Fellegi e Sunter (1969) – funções *em_winkler* e *matchingScore_C* – implementado em C++, porém também exige que os pareamentos sejam realizados com matrizes binárias (para discordância total e para concordância total), com a mesma limitação dos pacotes *fastLink* e *corlink*.

À luz do exposto, constatou-se que apenas o pacote *RecordLinkage* apresenta funcionalidades versáteis, além de restrições de formato de *inputs* que exigem tratamento prévio relativamente menor, o que permite sua utilização para aplicações reais. Por

conseguinte, este trabalho utilizou somente esse pacote para os testes empíricos de *performance* preditiva e processamento computacional.

6 APLICAÇÃO EMPÍRICA

Para verificação empírica, foram realizados pareamentos com a base de dados contendo registros do Registro Nacional de Condutores Habilitados (Renach) para inscritos no programa Minha Casa Minha Vida nos municípios de Rio de Janeiro, Guarulhos, Campinas e Sorocaba (188.150 observações). Essa base foi cruzada com a do Cadastro Único, a qual foi filtrada para os indivíduos nascidos em algum dos quatro municípios citados, resultando em uma base com 1.113.877 observações após tratamentos. Foram mantidos apenas registros com CPF diferente de *missing* para controlar o número real de *matches* verdadeiros, a fim de avaliar a eficiência dos métodos testados, tanto em relação à qualidade dos pareamentos gerados quanto ao custo computacional das operações processadas.

Testes empíricos foram realizados com o pacote *RecordLinkage* (versão 0.4-10), utilizando-se o software R (versão 3.4.2). Os códigos foram executados em um computador com quatro processadores Intel Xeon E7-4830 v2, com 2.20 *giga-hertz* (GHz) cada, e 512 *gigabytes* (GB) de memória RAM.

Testaram-se diversas combinações de variáveis e predicados lógicos para o *blocking*, bem como o pareamento de bases de diferentes tamanhos para a avaliação da estabilidade da *performance* preditiva para essas bases, além do crescimento relativo da exigência computacional. Ademais, almejou-se verificar o desempenho relativo dos pacotes disponíveis no R especializados em RL, visando à identificação de rotinas mais eficientes e ao descarte de funções menos eficientes para replicações futuras em aplicações práticas com grandes bases de dados, especialmente no cruzamento de registros administrativos em escala nacional.

6.1 Pareamento determinístico

Utilizando-se o código de CPF como controle, as duas bases de dados foram primeiramente pareadas deterministicamente para se verificar o número de pares de registros reais e usar esse número para a avaliação da qualidade dos pareamentos

probabilísticos. Com a conferência do CPF, constatou-se a existência de 18.510 pares de indivíduos não duplicados. Nem todos os registros com o mesmo CPF apresentaram, no entanto, concordância exata nas variáveis levantadas, o que motiva o uso de métodos probabilísticos; aquelas que evidenciaram registros diferentes entre as bases para pares de dados verdadeiros, bem como os respectivos pesos de concordância pelo método probabilístico e número de ocorrências de cada caso, estão dispostas na tabela 1.

TABELA 1
Dispersão de discordância entre as bases para pares verdadeiros

Variáveis distintas	Peso de concordância	Contagem
Todas as variáveis iguais	1.000000	11394
Nome do pai	0.796742	2537
Nome da mãe	0.793093	2338
Nome da pessoa	0.789403	796
Nome da mãe/Nome do pai	0.589835	472
Nome da pessoa/Nome da mãe	0.582496	307
Nome da pessoa/Nome do pai	0.586145	218
Sexo	0.989682	152
Nome da pessoa/Nome da mãe/Nome do pai	0.379238	81
Sexo/Nome da mãe	0.782775	52
Data de nascimento	0.843375	50
Sexo/Nome do pai	0.786424	45
Sexo/Nome da pessoa	0.779085	17
Data de nascimento/Nome do pai	0.640117	15
Data de nascimento/Nome da mãe	0.636468	8
Sexo/Nome da mãe/Nome do pai	0.579517	7
Data de nascimento/Nome da pessoa	0.632778	6
Nome da pessoa/Sexo/Nome da mãe	0.572178	5
Nome da pessoa/Data de nascimento/Nome da mãe	0.425871	3
Sexo/Data de nascimento	0.833057	2
Data de nascimento/Nome da mãe/Nome do pai	0.433210	2
Nome da pessoa/Data de nascimento/Nome do pai	0.429520	2
Nome da pessoa/Sexo/Nome da mãe/Nome do pai	0.368920	1

Elaboração dos autores.
Obs.: Figura cujos leiaute e textos não puderam ser padronizados e revisados em virtude das condições técnicas dos originais (nota do Editorial).

Erros na variável de data, em especial, foram analisados à parte, visando à identificação de potenciais padrões de dígitos que se confundem com frequência. Matrizes contendo a frequência de dígitos discrepantes entre as duas bases de dados (Cadastro Único nas linhas e Renach nas colunas) para o ano, o mês e o dia do nascimento da pessoa estão dispostas nas tabelas 2, 3 e 4.

TABELA 2
Frequência de dígitos distintos para pares verdadeiros, ano de nascimento

CadÚnico \ RENACH	0	1	2	3	4	5	6	7	8	9
0	0	0	0	3	0	0	0	0	2	0
1	0	0	0	0	2	0	0	2	0	0
2	0	2	0	0	2	0	0	0	0	0
3	0	0	0	0	0	2	0	2	0	0
4	0	0	0	0	0	2	0	2	0	0
5	0	0	0	0	0	0	2	0	0	0
6	0	2	0	2	2	0	0	4	3	2
7	0	0	0	0	2	0	0	0	3	2
8	0	0	0	0	0	0	0	2	0	0
9	2	0	2	3	0	2	2	0	2	0

Elaboração dos autores.
Obs.: Figura cujos leiaute e textos não puderam ser padronizados e revisados em virtude das condições técnicas dos originais (nota do Editorial).

TABELA 3
Frequência de dígitos distintos para pares verdadeiros, mês de nascimento

CadÚnico \ RENACH	0	1	2	3	4	5	6	7	8	9
0	0	0	1	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	1	1
2	1	2	0	1	2	0	0	0	1	0
3	0	0	1	0	1	2	1	0	0	0
4	0	1	0	1	0	2	0	2	0	0
5	0	0	1	0	1	0	2	0	0	0
6	0	0	1	0	0	0	0	4	3	0
7	0	0	0	0	0	0	1	0	0	0
8	1	0	0	0	1	0	1	0	0	1
9	0	1	0	0	0	2	0	0	0	0

Elaboração dos autores.
Obs.: Figura cujos leiaute e textos não puderam ser padronizados e revisados em virtude das condições técnicas dos originais (nota do Editorial).

TABELA 4
Frequência de dígitos distintos para pares verdadeiros, dia de nascimento

CadÚnico \ RENACH	0	1	2	3	4	5	6	7	8	9
0	0	1	1	0	1	1	1	0	2	0
1	1	0	1	0	2	1	1	2	0	0
2	1	2	0	0	0	1	1	0	0	0
3	1	0	0	0	0	0	1	2	1	1
4	1	0	0	0	0	0	0	2	0	0
5	0	0	0	1	0	0	2	0	0	1
6	1	0	0	0	2	0	0	0	0	0
7	0	1	0	1	2	1	0	0	0	2
8	1	1	1	0	0	1	0	0	0	0
9	2	1	0	0	0	2	2	1	2	0

Elaboração dos autores.
Obs.: Figura cujos leiaute e textos não puderam ser padronizados e revisados em virtude das condições técnicas dos originais (nota do Editorial).

6.2 Pareamento probabilístico

Para os pareamentos probabilísticos, verificou-se a *performance* do algoritmo de RL para diversas combinações de *blocking keys*; o valor de *cutoff* foi estabelecido em $P(match_j) = 0.8$.[8] Ambas as bases foram tratadas de modo a restarem as seguintes variáveis para comparação:

- nome completo da pessoa (NOME_PESSOA);
- sexo (SEXO);
- data de nascimento (DATA);
- nome completo da mãe da pessoa (NOME_MÃE); e
- nome completo do pai da pessoa (NOME_PAI).

8. Para a implementação de Winkler (1988), utilizou-se o *cutoff* 0.8 na escala normalizada $\frac{wm_j - wm_{mínimo}}{wm_{máximo} - wm_{mínimo}}$, definida em função dos pesos de concordância para cada *j*-ésimo par de registros comparados.

Este trabalho considerou nove combinações de variáveis de *blocking* e funções de distanciamento de *strings*.

1) Para os casos 1, 2 e 3: *blocking key* = x_1.
2) Para os casos 4, 5 e 6: *blocking key* = (NOME_PESSOA E SEXO) OU (NOME_PESSOA E DATA) OU (NOME_PESSOA E NOME_MÃE) OU (NOME_PESSOA E NOME_PAI) OU (SEXO E DATA) OU (SEXO E NOME_MÃE) OU (SEXO E NOME_PAI) OU (DATA E NOME_MÃE) OU (DATA E NOME_PAI) OU (NOME_MÃE E NOME_PAI).
3) Para os casos 7, 8 e 9: *blocking key* = (NOME_PESSOA E SEXO E DATA) OU (NOME_PESSOA E SEXO E NOME_MÃE) OU (NOME_PESSOA E SEXO E NOME_PAI) OU (NOME_PESSOA E DATA E NOME_MÃE) OU (NOME_PESSOA E DATA E NOME_PAI) OU (NOME_PESSOA E NOME_MÃE E NOME_PAI) OU (SEXO E DATA E NOME_MÃE) OU (SEXO E DATA E NOME_PAI) OU (SEXO E NOME_MÃE E NOME_PAI) OU (DATA E NOE_MÃE E NOME_PAI).
4) Para os casos 1, 4 e 7, não foi aplicada função de distanciamento de *strings*.
5) Para os casos 2, 5 e 8, foi aplicada a função de distanciamento de *strings* de Jaro-Winkler.
6) Para os casos 3, 6 e 9, foi aplicada a função de distanciamento de *strings* de Levenshtein.

A função de distanciamento de Jaro-Winkler entre duas *strings* s_1 e s_2 é dada por:

$$d_{s_1,s_2} = \frac{1}{3}\left(\frac{m}{a} + \frac{m}{b} + \frac{m - \frac{t}{2}}{m}\right)$$

Onde a e b são os tamanhos das *strings* s_1 e s_2, respectivamente; m é o número de caracteres em comum entre s_1 e s_2; e t é o número de transposições necessárias para que s_1 e s_2 se tornem idênticas.

Já a distância de Levenshtein (também conhecida como distância de edição) entre *strings* s_1 e s_2 fornece o número de operações – inserção/substituição/deleção de um caracter – necessárias para que *string* s_1 possa ser transformada em s_2.

A análise de todos os nove casos previamente definidos foi reiterada nos casos 10 a 18 para as variáveis de nome (da pessoa, da mãe e do pai), convertidos pelo algoritmo Metaphone ajustado para a pronúncia do português brasileiro, cuja implementação foi adicionada ao parâmetro *language* da função *metaphone* do pacote *phonics* do R.[9] Os *blocking keys* e funções de distanciamento de *strings* do caso 1 são os mesmos do caso 10, assim como os do 2 em relação ao 11, e assim por diante, até o 9, cuja configuração é a mesma do 18.

Caso a caso, o número de pares de registros que satisfizeram a condição de *blocking* e foram, consequentemente, comparados para o pareamento segue como a seguir:

- nos casos 1, 2 e 3, 181.675 pares de registros;
- nos casos 4, 5 e 6, 7.230.156 pares de registros;
- nos casos 7, 8 e 9, 29.842 pares de registros;
- nos casos 10, 11 e 12, 12.274.351 pares de registros;
- nos casos 13, 14 e 15, 62.632.078 pares de registros; e
- nos casos 16, 17 e 18, 80.715 pares de registros.

É possível notar que o uso de um predicado lógico dois a dois aumenta o número de comparações, dada a existência de várias combinações de concordância para que um par se qualifique para ser comparado, enquanto um predicado três a três força a concordância exata em mais atributos, diminuindo os pares comparados. Os casos 10 a 12, apesar de usarem apenas o nome da pessoa como chave, possuem grande número de pares qualificados em razão da possibilidade de múltiplas palavras retornarem a mesma transliteração pelo Metaphone_pt-BR.

Seguem apresentados nas tabelas 5 a 8 os resultados preditos e reais dos pareamentos para os dezoito casos testados, assim como métricas de desempenho usuais na literatura especializada. Foram testados os algoritmos descritos em Contiero *et al.* (2005) e Winkler (1988), e seus desempenhos preditivos e computacionais foram comparados e discutidos. Os mesmos resultados estão dispostos nos gráficos 1 e 2.

9. Disponível em: <https://github.com/ipea/phonics>.

TABELA 5
Pares preditos e encontrados, implementação de Contiero *et al.* (2005)

	PP	VP	FP	VN	FN	Tempo de processamento (minutos)
Caso 1	11627	11547	80	163085	6963	1.8644
Caso 2	65014	15301	49713	113452	3209	2.6795
Caso 3	15517	15121	396	162769	3389	2.5627
Caso 4	11627	11547	80	7211566	6963	24.1934
Caso 5	461817	16877	444940	6766706	1633	44.4006
Caso 6	31536	16476	15060	7196586	2034	34.0740
Caso 7	11627	11547	80	11252	6963	27.7398
Caso 8	27748	16466	11282	50	2044	26.4989
Caso 9	23461	16366	7095	4237	2144	27.0987
Caso 10	14138	12177	1961	12253880	6333	12.7378
Caso 11	764947	14205	750742	11505099	4305	15.9710
Caso 12	17304	13763	3541	12252300	4747	20.0414
Caso 13	14138	12177	1961	62611607	6333	67.2484
Caso 14	2693742	14927	2678815	59934753	3583	119.5100
Caso 15	25388	14181	11207	62602361	4329	81.4983
Caso 16	14138	12177	1961	60244	6333	16.3469
Caso 17	70674	14392	56282	5923	4118	18.9998
Caso 18	21823	14123	7700	54505	4387	14.6160

PP = Pares preditos; VP = Pares verdadeiros encontrados; FP = Falsos positivos
VN = Não-pares verdadeiros encontrados; FN = Falsos negativos

Elaboração dos autores.
Obs.: Figura cujos leiaute e textos não puderam ser padronizados e revisados em virtude das condições técnicas dos originais (nota do Editorial).

TABELA 6
Métricas de desempenho, implementação de Contiero *et al.* (2005)

	Acurácia	Reduction ratio	Precisão	Sensitividade	Especificidade	F1 Score
Caso 1	0.9612	0.0388	0.9931	0.6238	0.9995	0.7663
Caso 2	0.7087	0.2913	0.2353	0.8266	0.6953	0.3664
Caso 3	0.9792	0.0208	0.9745	0.8169	0.9976	0.8888
Caso 4	0.9990	0.0010	0.9931	0.6238	0.9999	0.7663
Caso 5	0.9382	0.0618	0.0365	0.9118	0.9383	0.0703
Caso 6	0.9976	0.0024	0.5225	0.8901	0.9979	0.6584
Caso 7	0.7640	0.2360	0.9931	0.6238	0.9929	0.7663
Caso 8	0.5534	0.4466	0.5934	0.8896	0.0044	0.7119
Caso 9	0.6904	0.3096	0.6976	0.8842	0.3739	0.7799
Caso 10	0.9993	0.0007	0.8613	0.6579	0.9998	0.7460
Caso 11	0.9385	0.0615	0.0186	0.7674	0.9387	0.0363
Caso 12	0.9993	0.0007	0.7954	0.7435	0.9997	0.7686
Caso 13	0.9999	0.0001	0.8613	0.6579	0.9999	0.7460
Caso 14	0.9572	0.0428	0.0055	0.8064	0.9572	0.0110
Caso 15	0.9998	0.0002	0.5586	0.7661	0.9998	0.6461
Caso 16	0.8972	0.1028	0.8613	0.6579	0.9685	0.7460
Caso 17	0.2517	0.7483	0.2036	0.7775	0.0952	0.3227
Caso 18	0.8503	0.1497	0.6472	0.7630	0.8762	0.7003

Elaboração dos autores.
Obs.: Figura cujos leiaute e textos não puderam ser padronizados e revisados em virtude das condições técnicas dos originais (nota do Editorial).

TABELA 7
Pares preditos e encontrados, implementação de Winkler (1988)

	PP	VP	FP	VN	FN	Tempo de processamento (minutos)
Caso 1	11627	11547	80	163085	6963	2.6945
Caso 2	13366	13257	109	163056	5253	3.7234
Caso 3	12692	12592	100	163065	5918	3.3041
Caso 4	12403	2695	9708	7201938	15815	61.4194
Caso 5	12516	2712	9804	7201842	15798	81.9945
Caso 6	12482	2724	9758	7201888	15786	69.9468
Caso 7	11425	11346	79	11253	7164	27.1057
Caso 8	13684	13566	118	11214	4944	26.2444
Caso 9	12761	12658	103	11229	5852	26.7662
Caso 10	14138	12177	1961	12253880	6333	40.6917
Caso 11	14819	12707	2112	12253729	5803	36.8939
Caso 12	14138	12177	1961	12253880	6333	40.5210
Caso 13	24461802	12251	24449551	38164017	6259	145.9929
Caso 14	29732490	15634	29716856	32896712	2876	230.1171
Caso 15	24461802	12251	24449551	38164017	6259	148.9286
Caso 16	13929	12024	1905	60300	6486	17.4241
Caso 17	14710	12635	2075	60130	5875	19.4240
Caso 18	13929	12024	1905	60300	6486	15.1025

PP = Pares preditos; VP = Pares verdadeiros encontrados; FP = Falsos positivos
VN = Não-pares verdadeiros encontrados; FN = Falsos negativos

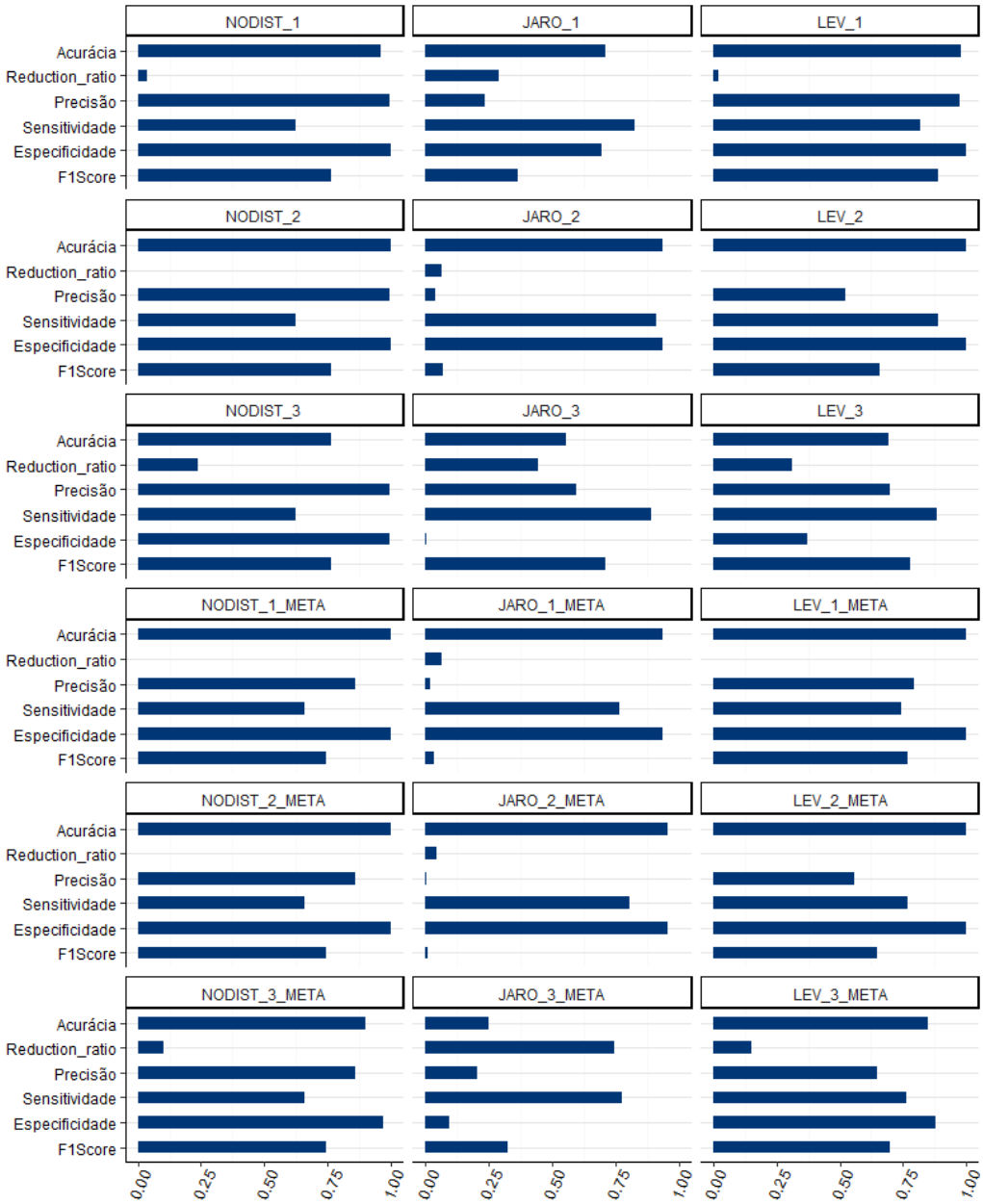
Elaboração dos autores.
Obs.: Figura cujos leiaute e textos não puderam ser padronizados e revisados em virtude das condições técnicas dos originais (nota do Editorial).

TABELA 8
Métricas de desempenho, implementação de Winkler (1988)

	Acurácia	Reduction ratio	Precisão	Sensitividade	Especificidade	F1 Score
Caso 1	0.9612	0.0388	0.9931	0.6238	0.9995	0.7663
Caso 2	0.9705	0.0295	0.9918	0.7162	0.9993	0.8318
Caso 3	0.9669	0.0331	0.9921	0.6803	0.9994	0.8071
Caso 4	0.9965	0.0035	0.2173	0.1456	0.9987	0.1744
Caso 5	0.9965	0.0035	0.2167	0.1465	0.9986	0.1748
Caso 6	0.9965	0.0035	0.2182	0.1472	0.9986	0.1758
Caso 7	0.7573	0.2427	0.9931	0.6130	0.9930	0.7580
Caso 8	0.8304	0.1696	0.9914	0.7329	0.9896	0.8428
Caso 9	0.8004	0.1996	0.9919	0.6838	0.9909	0.8096
Caso 10	0.9993	0.0007	0.8613	0.6579	0.9998	0.7460
Caso 11	0.9994	0.0006	0.8575	0.6865	0.9998	0.7625
Caso 12	0.9993	0.0007	0.8613	0.6579	0.9998	0.7460
Caso 13	0.6095	0.3905	0.0005	0.6619	0.6095	0.0010
Caso 14	0.5255	0.4745	0.0005	0.8446	0.5254	0.0011
Caso 15	0.6095	0.3905	0.0005	0.6619	0.6095	0.0010
Caso 16	0.8960	0.1040	0.8632	0.6496	0.9694	0.7413
Caso 17	0.9015	0.0985	0.8589	0.6826	0.9666	0.7607
Caso 18	0.8960	0.1040	0.8632	0.6496	0.9694	0.7413

Elaboração dos autores.
Obs.: Figura cujos leiaute e textos não puderam ser padronizados e revisados em virtude das condições técnicas dos originais (nota do Editorial).

GRÁFICO 1
Métricas de desempenho, implementação de Contiero *et al.* (2005)



Elaboração dos autores.
Obs.: Figura cujos leiaute e textos não puderam ser padronizados e revisados em virtude das condições técnicas dos originais (nota do Editorial).

GRÁFICO 2
Métricas de desempenho, implementação de Winkler (1988)

Elaboração dos autores.
Obs.: Figura cujos leiaute e textos não puderam ser padronizados e revisados em virtude das condições técnicas dos originais (nota do Editorial).

Observou-se pelas tabelas 5 e 7 que a inserção da função de distanciamento de *strings* aumenta significativamente o tempo de processamento; em especial, a função de Jaro-Winkler mostrou ser mais custosa em termos computacionais em comparação com a função de Levenshtein. Baseando-se na comparação da classificação dos pares preditos com os resultados reais observados (obtidos após o cruzamento das bases pelo CPF), foram geradas as métricas de desempenho mais utilizadas na literatura especializada para se avaliarem os resultados sobre a qualidade dos pareamentos apresentados nas tabelas 6 e 8.

Constatou-se que a acurácia dos casos testados foi em geral bastante alta (apenas 9 do total de 36 implementações apresentaram acurácia menor que 80%); em consequência, o *reduction ratio*, que é dado simplesmente pelo complementar da acurácia, mostrou valores baixos. No entanto, é sabido que ela é uma das métricas mais "fracas" para se avaliar o desempenho preditivo, uma vez que não reflete o balanceamento entre a identificação de pares verdadeiros e a proporção de classificações errôneas ou do número de pares não identificados. A depender do contexto, pode ser preferível capturar o número máximo de pares verdadeiros possível, mesmo se isso significar uma grande incidência de falsos positivos; assim como pode ser mais adequado evitar previsões equivocadas, tendo em contrapartida um alto número de pares reais classificados como não pares (falsos negativos). Dessa maneira, métricas que refletem esse *trade-off* são desejáveis no sentido de auxiliarem na escolha do modelo condicionado ao tipo de erro que se revela ser mais custoso para a aplicação específica em questão.

A precisão mede a proporção relativa dos pareamentos corretos em relação ao total de previsões realizadas, ou seja, mensura a proporção do erro tipo I nos resultados previstos. Ao contrário da acurácia, é possível notar casos em que a precisão apresenta valores indesejáveis – com destaque para os casos 5, 11 e 14 da implementação de Contiero *et al.* (2005) e para o 13, o 14 e o 15 da implementação de Winkler (1988), todos com precisão inferior a 4%, o que implica mais de 96% de falsos positivos do total de pares previstos. Comparando-se esses casos com o 7, por exemplo, nota-se que o número de pareamentos reais se elevou, contudo o de falsos positivos aumentou em proporção ainda maior, "contaminando" um índice de desempenho que leva esse fator em consideração. Em um cenário hipotético em que o intuito do pareamento fosse a identificação de receptores fraudulentos de programas de distribuição de renda, o alto número de falsos positivos poderia trazer conclusões errôneas sobre a eficiência do programa ou mesmo

culminar em eventuais litígios judiciais caso se optasse pelo cancelamento do benefício para cidadãos equivocadamente classificados como fraudulentos.

A sensitividade, em contrapartida, mede a proporção dos pareamentos corretos em relação ao total de pares verdadeiros existentes, permitindo observar-se a magnitude do erro tipo II (falsos negativos). Ela apresentou, no geral, valores inferiores à precisão, sendo especialmente baixa para os casos com *blocking* de predicados da implementação de Winkler (1988), sugestionados pelo alto número de falsos negativos em comparação com o baixo número de pares reais encontrados. Com esse indicador, foi possível notar a insuficiência da acurácia em refletir a qualidade dos pareamentos, que nos casos já citados foi influenciada fortemente pelo alto índice de verdadeiros negativos, o que é um resultado natural, dada a quantidade elevada dessa classe em relação ao número de pares de registros comparados; em especial, esse efeito é destacado para os casos 10 a 15, nos quais o número de pares comparados está na ordem das dezenas de milhões.

A especificidade funciona exatamente como a sensitividade, porém em relação aos não pares preditos e verdadeiros. No geral, essa métrica apresentou valores bastante elevados, influenciados em parte pelo fato de o número de não pares reais superar em muito o de pares reais – ou seja, o fato de as classes não serem balanceadas favorece a decisão de se classificar como não par, de modo que a quantidade de falsos negativos acaba sendo diluída pela baixa proporção em relação ao total de não pares. Mesmo para casos com mais de 15 mil falsos negativos, o alto índice de verdadeiros negativos faz com que essa métrica exiba bons valores, tornando-a, como consequência, relativamente menos informativa, sobretudo ao se lidar com grandes bases de dados, visto que em geral há mais *não matches* que *matches*.

Por fim, o *F1 score* (também chamado de *F-score*) é uma média harmônica entre a precisão e a sensitividade, e representa uma maneira sucinta de se avaliar o *trade-off* dos erros tipos I e II. A média harmônica possui a característica de ser conservadora, no sentido de ser mais sensível para valores inferiores, e portanto representa uma estatística descritiva mais pessimista que as médias aritmética $\left(\frac{x_1 + x_2}{2}\right)$ e geométrica $(\sqrt{x_1 \cdot x_2})$. Assim, o *F1 score* reflete o equilíbrio relativo entre os falsos positivos e os falsos negativos, priorizando o erro que mais predomina nas previsões realizadas, de modo que, ainda que a sensitividade seja alta, uma baixa precisão (ou vice-versa) fará com que esse índice seja baixo. Nota-se que o *F1 score* foi baixo para diversos

casos, em especial para aqueles em que o número de falsos positivos foi elevado, o que comprometeu a precisão e contaminou o *F1 score* indiretamente. De fato, os casos 13, 14 e 15 da implementação de Winkler (1988) apresentaram mais de 24 milhões de falsos positivos cada, fazendo com que a precisão e o *F1 score* ficassem abaixo de 1%.

Em termos gerais, o algoritmo de Contiero *et al.* (2005) mostrou uma exigência computacional menor, evidenciada pelos tempos de execução para todas as combinações de *blocking key* e funções de distanciamento de *strings*. No tocante à qualidade dos pareamentos, essa implementação parece ser altamente sensível à utilização da função de distância de Jaro-Winkler, dada a significativa piora dos índices de desempenho para os casos em que essa função foi utilizada (2, 5, 8, 11, 14 e 17), demonstrada pelo alto número de falsos positivos. Nos outros casos, porém, essa versão apresentou resultados melhores em relação ao algoritmo de Winkler (1988), o qual sistematicamente exibiu índices ruins para todos os casos com *blocking* por predicado, encontrando número de pares verdadeiros bastante abaixo dos demais casos.

7 CONCLUSÃO

Este estudo realizou um levantamento da literatura científica relativa ao problema do RL, descrevendo sua taxonomia básica e detalhando o principal modelo da vertente probabilística (Fellegi e Sunter, 1969), a qual continua sendo o principal método empregado em pesquisas recentes de ponta relativas a esse tema. Ademais, foram listadas as principais métricas de desempenho utilizadas nessa literatura para a avaliação dos resultados, bem como técnicas alternativas de *blocking* e desenvolvimentos recentes agregando técnicas de *machine learning* no contexto do RL. Por fim, foram feitos testes empíricos com bases de dados de grande porte para se verificar o desempenho preditivo de diferentes implementações, chaves de *blocking* e funções de distanciamento de *strings*, assim como o crescimento da exigência computacional. O intuito foi de identificar potenciais heurísticas a serem adotadas (ou evitadas) para a aplicabilidade eficiente do RL probabilístico para bases de dados ainda maiores – idealmente, visando à integração dos registros administrativos brasileiros, bases que comumente possuem número de observações na casa das centenas de milhões.

Os resultados ratificam a complexidade e o desafio do problema do RL, especialmente o pareamento de bases de dados com grande número de observações, visto pelo acentuado crescimento da complexidade computacional à medida que as bases pareadas sejam maiores. Em termos de desenvolvimentos futuros, é pertinente considerar extensões em processamento distribuído e *clustered file system* (sistema de arquivo em *cluster*), a fim de acelerar o tempo de execução dos algoritmos, principalmente tendo em mente a exigência adicional em memória RAM para grandes bases de dados. *Frameworks* como Apache Hadoop e Apache Spark podem ser explorados para a garantia da viabilidade temporal para o pareamento de bases maiores.

Em linhas gerais, a implementação de Contiero *et al.* (2005) apresentou melhores resultados que o algoritmo de Winkler (1988), tanto em termos computacionais quanto no tocante às métricas de desempenho – observando-se a mais restritiva, *F1 score*, é possível constatar que esse índice favoreceu a segunda abordagem apenas nos casos em que a função de Jaro-Winkler foi utilizada. Dessa forma, é razoável inferir uma certa incompatibilidade entre essa função de distanciamento e a primeira implementação (testes futuros acerca dessa conjectura são recomendáveis). Em contrapartida, a versão de Winkler (1988) exibiu resultados significativamente piores para os casos em que o *blocking key* era composto por uma união de interseções (predicado lógico), de modo que é aconselhável não utilizar esse algoritmo para aplicações reais de pareamentos em que há disponibilidade de grande número de variáveis que podem ser combinadas para o *blocking*; em vez disso, essa abordagem parece funcionar melhor para *blocking* com apenas uma chave.

Os desempenhos preditivos vistos sugerem que a escolha das variáveis de *blocking* e da função de distanciamento de *strings* exerce grande influência na qualidade dos pareamentos, e parecem impactar de maneira heterogênea as diversas métricas de desempenho, especificamente em relação aos erros tipos I e II. Com base nos testes realizados, há evidências de que a inserção da função de distanciamento de *strings* aumenta o número de pareamentos corretos encontrados, porém também eleva o de falsos positivos (erro tipo I); em contrapartida, o número de falsos positivos é bem menor para os casos sem o uso desse argumento (1 e 4), em troca de uma quantidade maior de falsos negativos (erro tipo II), cuja proporção acaba não sofrendo grandes alterações em vista do forte desbalanceamento entre as classes *match* e *não match*. A depender do contexto da aplicação do pareamento de registros – ou seja, o custo relativo

de se ter um falso positivo em relação ao de se ter um falso negativo –, essa constatação pode ser utilizada como subsídio para a definição dos argumentos associados ao método a ser utilizado.

Este estudo utilizou um pareamento determinístico *a priori* a fim de avaliar o desempenho dos métodos probabilísticos. Ele está fundamentado, portanto, em bases de dados grandes, porém com tamanhos factíveis para um pareamento determinístico, fator que constitui uma limitação diante da mensuração do desafio de se parearem bases ainda maiores, que sejam condizentes com os registros administrativos em escala nacional. Além disso, a qualidade das bases selecionadas (Cadastro Único e Renach) pode não ser representativa para replicações futuras em bases alternativas – por exemplo, caso a qualidade dos dados seja bastante diferente da qualidade das bases utilizadas neste estudo, espera-se que as conclusões obtidas difiram em alguma escala. Nesse sentido, verificações futuras com distintas bases de dados são encorajadas, bem como replicações do exercício empírico proposto para outras implementações, técnicas de *machine learning*, combinações de *blocking keys* e funções de distanciamento de *strings* que não foram contempladas aqui.

REFERÊNCIAS

BHATTACHARYA, I.; GETOOR, L. A latent dirichlet allocation model for entity resolution. *In*: SIAM INTERNATIONAL CONFERENCE ON DATA MINING, 6., 2005.

BUDAVÁRI, T.; LOREDO, T. J. Probabilistic record linkage in astronomy: directional cross-identification and beyond. **Annual Review of Statistics and Its Application**, v. 2, p. 113-139, 2015.

BUTTON, L. A. *et al.* Hospitalized incidence and case fatality for upper gastrointestinal bleeding from 1999 to 2007: a record linkage study. **Alimentary Pharmacology e Therapeutics**, v. 33, n. 1, p. 64-76, 2011.

CHRISTEN, P. A survey of indexing techniques for scalable record linkage and deduplication. **IEEE Transactions on Knowledge and Data Engineering**, v. 24, n. 9, p. 1537-1555, 2012.

CHRISTEN, P.; GOISER, K. Quality and complexity measures for data linkage and deduplication. *In*: GUILLET, F.; HAMILTON, H. J. (Ed.). **Quality Measures in Data Mining**. v. 43. New York: Springer Berlin Heidelberg, 2007. p. 127-151.

CHURCHES, T. *et al.* Preparation of name and address data for record linkage using hidden markov models. **BMC Medical Informatics and Decision Making**, v. 2, n. 1, 2002.

CLIFTON, C. *et al.* Privacy-preserving data integration and sharing. *In*: ACM SIGMOD WORKSHOP ON RESEARCH ISSUES IN DATA MINING AND KNOWLEDGE DISCOVERY, 9., 2004, Paris. **Anais**... Paris: ACM, 2004.

CONTIERO, P. *et al.* The epilink record linkage software. **Methods of Information in Medicine**, v. 44, n. 1, p. 66-71, 2005.

COOK, L. J. *et al.* Probabilistic record linkage: relationships between file sizes, identifiers, and match weights. **Methods Archive**, v. 40, n. 3, p. 196-203, 2001.

DUSETZINA, S. B. *et al.* **Linking data for health services research**: a framework and instructional guide. Rockville: Agency for Healthcare Research; Quality, 2014.

ENAMORADO, T.; FIFIELD, B.; IMAI, K. **Using a probabilistic model to assist merging of large-scale administrative records**. New Jersey: Princeton University, 2017.

FAIR, M. Generalized record linkage system-statistics Canada's record linkage software. **Austrian Journal of Statistics**, v. 33, n. 1-2, p. 37-53, 2016.

FEIGENBAUM, J. J. **A machine learning approach to census record linking**. Cambridge: Harvard University, 2016. Disponível em: <https://scholar.harvard.edu/files/jfeigenbaum/files/feigenbaum-censuslink.pdf>.

FELLEGI, I. P.; SUNTER, A. B. A theory for record linkage. **Journal of the American Statistical Association**, v. 64, n. 328, p. 1183-1210, 1969.

FERREIRA, F. P. M. Registros administrativos como fonte de dados estatísticos. **Informática Pública**, ano 10, n. 1, p. 81-93, 2008.

FU, Z. *et al.* Automatic record linkage of individuals and households in historical census data. **International Journal of Humanities and Arts Computing**, v. 8, n. 2, p. 204-25, 2014.

GILL, L. **Methods for automatic record matching and linkage and their use in national statistics**. United Kingdom: Office for National Statistics, 2001.

GRANNIS, S. J. J. *et al.* Analysis of a probabilistic record linkage technique without human review. *In*: AMIA ANNUAL SYMPOSIUM, 2003, Washington. **Anais**... Washington: AMIA, 2003.

HERNÁNDEZ, M. A.; STOLFO, S. J. Real-world data is dirty: data cleansing and the merge/purge problem. **Data Mining and Knowledge Discovery**, v. 2, n. 1, p. 9-37, 1998.

JORDÃO, C. C.; ROSA, J. L. G. Metaphone-Pt_BR: the phonetic importance on search and correction of textual information. *In*: INTERNATIONAL CONFERENCE, 13., 2012, New Delhi, India. **Anais**... New Delhi: CICLing, 2012.

JUTTE, D. P.; ROOS, L. L.; BROWNELL, M. D. Administrative record linkage as a tool for public health research. **Annual Review of Public Health**, v. 32, p. 91-108, 2011.

KELMAN, C. W. A.; BASS, J.; HOLMAN, C. D. J. Research use of linked health data: a best practice protocol. **Australian and New Zealand Journal of Public Health**, v. 26, n. 3, p. 251-55, 2002.

KOUDAS, N.; SARAWAGI, S.; SRIVASTAVA, D. Record linkage: similarity measures and algorithms. *In*: ACM SIGMOD INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA, 25., 2006, Chicago. **Anais**... Chicago: ACM, 2006.

MICHELSON, M.; KNOBLOCK, C. A. **Learning blocking schemes for record linkage**. Marina del Rey: University of Southern California, 2006. p. 440-445.

MURUGESAN, M. *et al.* Efficient privacy-preserving similar document detection. **The VLDB Journal**: the International Journal on Very Large Data Bases, v. 19, n. 4, p. 457-475, 2010.

NATHAN, G. Outcome probabilities for a record matching process with complete invariant information. **Journal of the American Statistical Association**, v. 62, n. 318, p. 454-469, 1967.

NEWCOMBE, H. B. Record linking: the design of efficient systems for linking records into individual and family histories. **American Journal of Human Genetics**, v. 19, n. 3, p. 335-359, 1967.

NEWCOMBE, H. B. *et al.* Reliability of computerized versus manual death searches in a study of the health of eldorado uranium workers. **Computers in Biology and Medicine**, v. 13, n. 3, p. 157-169, 1983.

PHUA, C. *et al.* Resilient identity crime detection. **IEEE Transactions on Knowledge and Data Engineering**, v. 24, n. 3, p. 533-546, 2012.

PITA, R. D. da R. **Correlação probabilística implementada em spark para big data em saúde**. 2016. Tese (Doutorado) – Instituto de Matemática, Universidade Federal da Bahia, Salvador, 2016.

PITA, R. D. da R. *et al.* A machine learning trainable model to assess the accuracy of probabilistic record linkage. Trabalho apresentado em 19th International Conference on Big Data Analytics and Knowledge Discovery. Lyon, France, Springer, 2017a.

_____. Design and evaluation of probabilistic record linkage methods supporting the Brazilian 100-million cohort initiative. **International Journal for Population Data Science**, v. 1, n. 1, 2017b.

QUANTIN, C. *et al.* How to ensure data security of an epidemiological follow-up: quality assessment of an anonymous record linkage procedure. **International Journal of Medical Informatics**, v. 49, n. 1, p. 117-122, 1998.

RANDALL, S. M. *et al.* The effect of data cleaning on record linkage quality. **BMC Medical Informatics and Decision Making**, v. 13, n. 1, p. 64, 2013.

RICHARDS, L. *et al.* Comparing classifiers in historical census linkage. Trabalho apresentado em 2014 IEEE International Conference on Data Mining Workshop. Shenzhen, China, 2014.

SARIYAR, M.; BORG, A.; POMMERENING, K. Evaluation of record linkage methods for iterative insertions. **Methods of Information in Medicine**, v. 48, n. 5, p. 429-437, 2009.

SAYERS, A. *et al.* Probabilistic record linkage. **International Journal of Epidemiology**, v. 45, n. 3, p. 954-964, 2015.

SCHNEIER, B. **Applied cryptography**: protocols, algorithms, and source code in C. 2 ed. New Jersey: John Wiley e Sons, 2007.

SCHNELL, R.; BACHTELER, T.; BENDER, S. A toolbox for record linkage. **Austrian Journal of Statistics**, v. 33, n. 1-2, p. 125-33, 2004.

STEORTS, R. C. *et al.* Entity resolution with empirically motivated priors. **Bayesian Analysis**, v. 10, n. 4, p. 849-75, 2015.

TEPPING, B. J. A model for optimum linkage of records. **Journal of the American Statistical Association**, v. 63, n. 324, p. 1321-1332, 1968.

VATSALAN, D.; CHRISTEN, P.; VERYKIOS, V. S. A taxonomy of privacy-preserving record linkage techniques. **Information Systems**, v. 38, n. 6, p. 946-969, 2013.

WAJDA, A.; ROOS, L. L. Simplifying record linkage: software and strategy. **Computers in Biology and Medicine**, v. 17, n. 4, p. 239-248, 1987.

WANG, F.; WANG, H. Record linkage using the combination of twice iterative svm training and controllable manual review. Trabalho apresentado em IEEE 14th International Conference on Dependable, Autonomic and Secure Computing; 14th Intl Conf on Pervasive Intelligence and Computing; e 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress. Auckland, New Zealand, 2016.

WANG, G.; CHEN, H.; ATABAKHSH, H. Automatically detecting deceptive criminal identities. **Communications of the ACM**, v. 47, n. 3, p. 70-76, 2004.

WILSON, D. R. Beyond probabilistic record linkage: using neural networks and complex features to improve genealogical record linkage. Trabalho apresentado em Neural Networks (Ijcnn), The 2011 International Joint Conference. San Jose, California, 2011.

WINKLER, W. E. Using the EM algorithm for weight computation in the Fellegi-Sunter model of record linkage. *In*: SECTION ON SURVEY RESEARCH METHODS, AMERICAN STATISTICAL ASSOCIATION, 1988, Washington. **Anais**... Washington: ASA, 1988.

_____. **Overview of record linkage and current research directions**. Washington: Bureau of the Census, 2006. (Research Report Series).

Ipea – Instituto de Pesquisa Econômica Aplicada

Assessoria de Imprensa e Comunicação

EDITORIAL

Coordenação
Cláudio Passos de Oliveira

Supervisão
Andrea Bossle de Abreu

Revisão
Carlos Eduardo Gonçalves de Melo
Elaine Oliveira Couto
Lis Silva Hall
Mariana Silva de Lima
Rava Caldeira de Andrada Vieira
Vivian Barros Volotão Santos
Bruna Oliveira Ranquine da Rocha (estagiária)
Lorena de Sant'Anna Fontoura Vale (estagiária)

Editoração
Aline Cristine Torres da Silva Martins
Carlos Henrique Santos Vianna
Mayana Mendes de Mattos (estagiária)
Vinícius Arruda de Souza (estagiário)

Capa
Danielle de Oliveira Ayres
Flaviane Dias de Sant'ana

Projeto Gráfico
Renato Rodrigues Bueno

The manuscripts in languages other than Portuguese published herein have not been proofread.

Livraria Ipea

SBS – Quadra 1 - Bloco J - Ed. BNDES, Térreo.
70076-900 – Brasília – DF
Fone: (61) 2026-5336

Correio eletrônico: livraria@ipea.gov.br

Composto em adobe garamond pro 12/16 (texto)
Frutiger 67 bold condensed (títulos, gráficos e tabelas)
Rio de Janeiro-RJ

Missão do Ipea
Aprimorar as políticas públicas essenciais ao desenvolvimento brasileiro por meio da produção e disseminação de conhecimentos e da assessoria ao Estado nas suas decisões estratégicas.

ipea Instituto de Pesquisa Econômica Aplicada

MINISTÉRIO DO PLANEJAMENTO, DESENVOLVIMENTO E GESTÃO

ISSN 1415-4765