Methods

2022-03-03

Methods

General description

We built an R (R Core Team 2020) based framework to simplify two aspects of systematic literature review: record acquisition and classification. The code used to generate the results is available at https://github.com/AD-Papers-Material/BART_SystReviewClassifier, while an updated and ready to use version of the framework is distributed as an R package at https://github.com/bakaburg1/BaySREn. The framework includes several modules that communicate through intermediate outputs stored in standard formats, making it possible for users to extend the framework or easily integrate it with other tools in their pipeline. See Supplemental Material S1 for an in-depth description of the framework and how to use it.
The tasks carried out by the framework are grouped into "sessions," which comprise obtaining scientific citation data (records) using a search query and then labelling them as relevant ("positive" in the rest of the text) or not ("negative") for the topic of interest with the help of a machine learning engine (Fig. 1). The initial search query should be built using domain knowledge, trying to achieve a high relevant/non-relevant record ratio.
The framework can then generate a new data-driven query from this labelled set to perform a new session to find records possibly missed by the first query.

Record acquisition and initial labelling

We built a set of tools to allow users to automatically search and download citation data from three major scientific databases ("sources"): Pubmed (https://pubmed.ncbi.nlm.nih.gov/), Web Of Science (WOS, https://apps.webofknowledge.com/) and the Institute of Electrical and Electronics Engineers (IEEE, https://ieeexplore.ieee.org/Xplore/home.jsp). The framework handles authorisation management for non-open databases like WOS and IEEE. It is also possible to import previously downloaded records in the framework; this is particularly useful for acquiring records from SCOPUS (https://www.scopus.com/) and EMBASE databases (https://www.embase.com/), for which a comprehensive API interface was not easy to build. An extra manual search was also necessary for Pubmed since the API and the web interface have different rule expansion algorithms and return slightly different results ("NCBI Insights : Updated Pubmed e-Utilities Coming in April 2022!" n.d.). A short guide on how to set up the framework for each database supported is available in Supplemental Material S3.
The collected records are merged into a single database, resolving duplicates and different formatting between sources. The records are ordered according to the frequency of the positive query terms (e.g., not preceded by a *NOT* modifier) in the title and abstract ("simple query ordering").
The researcher is then asked to label a subset of records to create the "initial training set" needed to start the automatic classification. We recommend manually labelling the first 250 records (see "hyperparameter optimisation" later). Simple query ordering increases the positivity rate in the initial training set (Wallace et al. 2010), leading to higher sensitivity during automatic classification (Chawla, Japkowicz, and Kotcz 2004).

Text feature extraction

The framework models the relevance of a record based on the following fields in the citation data: title, abstract, authors, keywords, MeSH terms (Lipscomb 2000). A range of Natural Language Processing (NLP) techniques

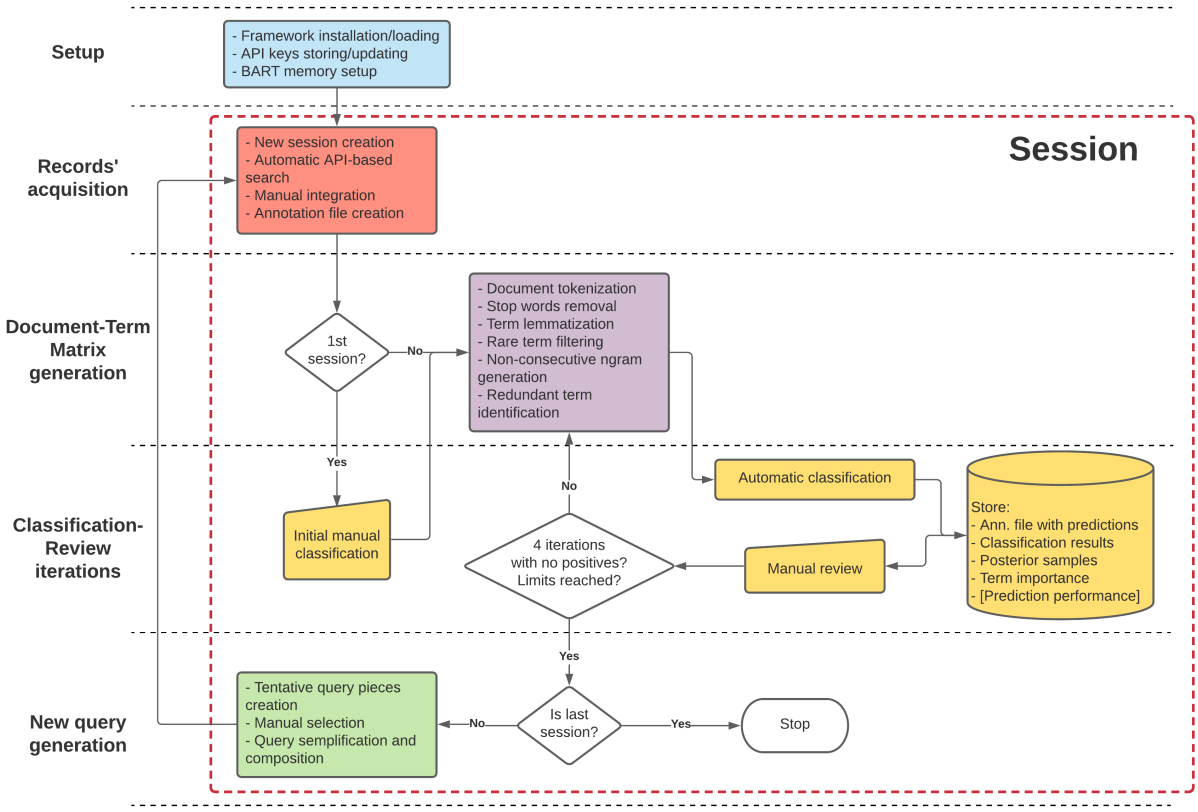

Figure 1. Framework's visual representation.

(Baeza-Yates, Ribeiro-Neto, and others 1999; Marshall and Wallace 2019; Ananiadou and McNaught 2006) are employed to convert the textual information in these fields into features for machine learning through a bag-of-words approach (Marshall and Wallace 2019). Processing of free text fields (title, abstract) includes: tokenisation (i.e., extracting the terms), removal of common stopwords (i.e., sentence components having no semantic value), part-of-speech filtering (only nouns, adjectives, verbs and untagged terms are retained), and lemmatisation of terms (i.e., reduction to their base grammatical form). Text processing for authors, keywords and MeSH terms identifies logical units (e.g., authors' full names, composite keywords) and extracts them.
Terms appearing in less than 5% of the labelled documents are removed from negative records. All terms in the positive set are kept to increase sensitivity at the cost of specificity.
Some terms tend to co-appear in records (non-consecutive ngrams, nc-ngrams), often carrying a particular meaning when they do co-occur. To detect nc-ngrams, we generated a word network representation (Rousseau 2015) with edges occurring between terms with a cosine similarity in terms of document co-occurrence > 0.5. We extracted the maximal cliques in the network (Eppstein, Löffler, and Strash 2010) representing highly correlated groups of terms; these groups are added to the dataset as individual features. Only nc-ngrams comprising a maximum of ten terms are kept.
A second network is built using a co-occurrence threshold of 0.9. In this case, the cliques represent terms that always appear together and can therefore be considered redundant (i.e., they do not need to be considered separately). These terms are merged to increase computational efficiency and reduce overfitting.
The output is a Document-Term Matrix (DTM), with N_d rows representing the records (D_i), N_t terms column for the t_{field} terms (divided by record field) and $0, 1$ values whether $t_{field} \in D_i$. We also enriched the DTM with features referencing the number of terms in each field to help the model scale term importance based on the field length.

Label prediction

We used a Bayesian Additive Regression Trees (BART) machine learning "classification model" (Chipman et al. 2010) (in the implementation of Kapelner and Bleich 2013) to predict the probability of a record being relevant, given the information coded into the enriched DTM and the initial training set. We set up the BART model to use 2,000 MCMC iterations (after 250 burn-in iterations) and 50 trees; we used a k value of 2 to regularise extreme prediction and let the model use missing fields in the DTM as features (Kapelner and Bleich 2015). Positive records are oversampled ten times to increase sensitivity (Batista, Prati, and Monard 2004).
The output is the expected value posterior predictive distribution for each record (PPD in brief) describing the probability of it being relevant (i.e., a positive match). An ensemble of ten models was fitted to improve prediction stability by averaging the PPD between models (Zhou 2021; Dietterich 2000).

To assign the labels, we employed an "active learning" approach (Settles 2009; Miwa et al. 2014), where a human reviews a specific subset of predictions made by the machine, which is then retrained on the manually reviewed dataset. This process is carried out iteratively to reduce prediction uncertainty.
Label assignment is done through identification of an "uncertainty zone," the construction of which is possible thanks to the Bayesian nature of BART, which provides full PPDs instead of point-wise predictions for each record.
To describe the process formally, we define:

$$\pi_i = \frac{1}{M} \sum_{j=1}^{M} Pr(L_i = 1 | DTM, m_j) \qquad \text{(Eq. 1)}$$

as the PPD of a record D_i being relevant (i.e, having a positive label, $L_i = 1$), averaging the PPDs of the ensemble of $M = 10$ models m, and:

$$\pi_{i,l} = \{\pi_i : Pr(\pi_i) = 1\%\}$$
$$\pi_{i,u} = \{\pi_i : Pr(\pi_i) = 99\%\} \quad \text{(Eq. 2)}$$

respectively as the lower and upper boundaries of the 98% quantile interval of π_i (98% Predictive Interval, 98% PrI).
Then we identify the "uncertainty zone" as:

$$U_\pi = [\max \vec{\pi}_u^-, \min \vec{\pi}_l^+] \quad \text{(Eq. 3)}$$

with $\vec{\pi}_u^-$ being the vector of $\pi_{i,u}$ with a negative label and $\vec{\pi}_l^+$ the vector of $\pi_{i,l}$ with a positive label. That is, U_π defines a range of values between the smallest $\pi_{i,l}$ in the set of already labelled positive records L_p and the largest $pi_{i,u}$ related to the negative ones L_n, noting that the two limits can appear in any order.
Consequently, a record D_i will be labelled as positive if:

$$\pi_{i,l} > \max_{\pi \in U_\pi} \pi \quad \text{(Eq. 4)}$$

that is, the record lower 98% PrI boundary should be higher than every value in the uncertainty zone. In other words, for a record to be labelled positive, its PPD should be within the range of the mixture of PPD of the records previously labelled positive and should not cross the distributions of the negative records.
Conversely, a record is labelled as negative if:

$$\pi_{i,u} < \min_{\pi \in U_\pi} \pi \quad \text{(Eq. 5)}$$

The remaining records are labelled as "uncertain."

Manual review is then necessary for: 1) uncertain records, 2) positive records (to avoid false positives), and 3) records whose predicted label differs from the existing manual one. The last case helps identify human errors or inconsistent labelling criteria.

The automatic classification and manual review steps alternate in a loop (CR iterations) until no new positive matches are found in four consecutive iterations.

Relevant term extraction

As a measure of feature importance, we computed the "inclusion rate," that is, the proportion of times a term is used in a posterior tree over the sum of total inclusions of all variables (Kapelner and Bleich 2013). We extracted the terms, the portion of the citation data in which they were used, the average "inclusion rate" among the ensemble models (over 10,000 inclusions) and its ratio over the standard deviation of this inclusion (inclusion stability, IS). For each term, we ran a Poisson regression to get the linear association with a positive label and reported it as Relative Risk (RR) with the number of standard errors as significance index (Statistic); the comparison between the inclusion rate in the BART models and the linear association allows to spot relevant non-linear effects (i.e., the feature is relevant only in association with others). In the Results, we only listed the first 15 terms with IS > 1.5 (in order of inclusion rate), while the first fifty terms, regardless of inclusion stability, are listed in Supplemental Material S2.

New search query generation

We developed an algorithm that generates a new search query to find further relevant publications missed in the initial search, possibly at a reasonable cost to specificity (i.e., a higher number of negative results).
The algorithm is composed of the following steps:

- A partition tree (Therneau and Atkinson 2019) is built between the DTM and 800 samples from the PPD; if a term is present multiple times in the DTM (e.g., both in the title and abstract), it is counted just once, and field term count features are removed. This step generates a list of rules composed by AND/NOT "conditions" made of terms/authors/keywords/MeSH tokens, which together identify a group of records.
- For each rule, negative conditions (i.e., NOT statements) are added iteratively, starting from the most specific one, until no conditions are found that would not also remove positive records.
- The extended set of rules is sorted by positive-negative record difference in descending order. The cumulative number of unique positive records is computed and used to group the rules. Rules inside each group are ordered by specificity.
- The researcher is then asked to review the rule groups and select one or more rules from each group or edit overly specific rules (e.g., citing a non-relevant concept casually associated with a paper, like a numeric value or indicator). It is possible to exclude a group of rules altogether, especially those with the poorest sensitivity/specificity ratio.
- The selected rules are joined together by OR statements, defining a subset of records with a sensibly higher proportion of positive records than the original set
- Redundant (i.e., rules whose positive records are already included in more specific ones) and non-relevant rules (i.e., conditions that when removed do not impact sensitivity and specificity) are removed.
- Finally, the rules are re-elaborated in a query that can be used to perform a new citation search.

Because the algorithm is data-driven, it creates queries that effectively select positive records from the input dataset but may be not specific enough when applied to actual research databases. Therefore we added an extra subquery in _AND_that specifies the general topics of our search and narrows the search domain. The new query was used to initiate a second search session.

Performance evaluation

We trained a simple Bayesian logistic regression (surrogate model) on the reviewed records to evaluate the consistency of the classification model (see Discussion for the theoretical justification). The surrogate model uses as predictor the lower boundary of the 98% PrI of the PPD of the records with weakly regularising, robust priors for the intercept (Student T with $\nu = 3, \mu = 0, \sigma = 2.5$) and the linear coefficient (Student T with $\nu = 3, \mu = 0, \sigma = 1.5$).
The quality of the model was evaluated through the Bayesian R^2 (Gelman et al. 2019), of which we reported the posterior median and 90% Credible Interval [90% CrI]. The R^2 also provides an evaluation of the consistency of the original classification model. Given that this model is conditional only on the BART predictions and not on the DTM, it is characterised by more uncertainty, providing plausible worst-case scenarios.
The surrogate model is then used to generate the predictive cumulative distribution of the number of total positive records in the whole dataset. This distribution allows estimating the expected total posterior "Sensitivity" and "Efficiency" of the classification model in the whole (unreviewed) dataset. Efficiency is summarised by the "Work saved over random" (WSoR) statistic: one minus the ratio between the number of records manually reviewed and those that would be required to find the same number of positives if classification were performed choosing records randomly; this last quantity is estimated through a negative hypergeometric distribution (Chae 1993) over the predicted number of positive records.
For the number of predicted positive records, sensitivity and efficiency, we reported the "truncated 90% PrI" [trunc. 90% PrI], i.e., the uncertainty interval bounded by the number of observed total positive records (i.e., there cannot be fewer predicted positive records than observed).

Hyperparameter evaluation

Our classification algorithm has a limited number of hyperparameters:

- Size of the initial training set: 50, 100, 250, 500 records;
- Number of models in the ensemble: 1, 5, 10, 20, 40, 60 repetitions;
- Oversampling rate of positive records: (1x (i.e., no oversampling), 10x, 20x;

- PrI quantiles for building the uncertainty zone: 80%, 90%, 98%;
- Source of randomness between models in the ensemble: MCMC sampling only (Robert and Casella 2004), MCMC plus data bootstrapping (Breiman 1996) of the training set.

To evaluate the hyperparameter effect of performance, we set up a "grid search" (Claesen and De Moor 2015; Yang and Shami 2020) on a prelabelled "validation set" derived from the first 1,200 records of the first session dataset. Each hyperparameter combination was tested until four CR iterations were completed with no positive records or until the whole dataset was labelled.
For each combination, a performance score was computed as the product of "Efficiency" (1 minus the ratio of records that required reviewing over the total number of records) and "Sensitivity" (number of positive records found over the total number of positive records). We then used a partition tree (Therneau and Atkinson 2019) to identify homogeneous "performance clusters" of scores given hyperparameter values. For the rest of the study, we used the best hyperparameter set in terms of sensitivity followed by efficiency from the cluster with the highest average score.

Software and data

The framework is built with R v4.0.4 (R Core Team 2021). The R packages required by the framework are listed at https://github.com/bakaburg1/BaySREn/blob/main/DESCRIPTION. All relevant data necessary to replicate the results is available at https://doi.org/10.5281/zenodo.6323360.

Ananiadou, Sophia, and John McNaught. 2006. *Text Mining for Biology and Biomedicine*. Citeseer.

Baeza-Yates, Ricardo, Berthier Ribeiro-Neto, and others. 1999. *Modern Information Retrieval*. Vol. 463. ACM press New York.

Batista, Gustavo EAPA, Ronaldo C Prati, and Maria Carolina Monard. 2004. "A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data." *ACM SIGKDD Explorations Newsletter* 6 (1): 20–29.

Breiman, Leo. 1996. "Bagging Predictors." *Machine Learning* 24 (2): 123–40.

Chae, Kyung-Chul. 1993. "Presenting the Negative Hypergeometric Distribution to the Introductory Statistics Courses." *International Journal of Mathematical Education in Science and Technology* 24 (4): 523–26.

Chawla, Nitesh V, Nathalie Japkowicz, and Aleksander Kotcz. 2004. "Special Issue on Learning from Imbalanced Data Sets." *ACM SIGKDD Explorations Newsletter* 6 (1): 1–6.

Chipman, Hugh A, Edward I George, Robert E McCulloch, and others. 2010. "BART: Bayesian Additive Regression Trees." *The Annals of Applied Statistics* 4 (1): 266–98.

Claesen, Marc, and Bart De Moor. 2015. "Hyperparameter Search in Machine Learning." *arXiv Preprint arXiv:1502.02127*.

Dietterich, Thomas G. 2000. "Ensemble Methods in Machine Learning." In *International Workshop on Multiple Classifier Systems*, 1–15. Springer.

Eppstein, David, Maarten Löffler, and Darren Strash. 2010. "Listing All Maximal Cliques in Sparse Graphs in Near-Optimal Time." In *International Symposium on Algorithms and Computation*, 403–14. Springer.

Gelman, Andrew, Ben Goodrich, Jonah Gabry, and Aki Vehtari. 2019. "R-Squared for Bayesian Regression Models." *The American Statistician*.

Kapelner, Adam, and Justin Bleich. 2013. "bartMachine: Machine Learning with Bayesian Additive Regression Trees." *arXiv Preprint arXiv:1312.2171*.

———. 2015. "Prediction with Missing Data via Bayesian Additive Regression Trees." *Canadian Journal of Statistics* 43 (2): 224–39.

Lipscomb, Carolyn E. 2000. "Medical Subject Headings (MeSH)." *Bulletin of the Medical Library Association* 88 (3): 265.

Marshall, Iain J, and Byron C Wallace. 2019. "Toward Systematic Review Automation: A Practical Guide to Using Machine Learning Tools in Research Synthesis." In *Systematic Reviews*, 8:1–10. 1. Springer.

Miwa, Makoto, James Thomas, Alison O'Mara-Eves, and Sophia Ananiadou. 2014. "Reducing Systematic Review Workload Through Certainty-Based Screening." *Journal of Biomedical Informatics* 51: 242–53.

"NCBI Insights : Updated Pubmed e-Utilities Coming in April 2022!" n.d. U.S. National Library of Medicine. https://ncbiinsights.ncbi.nlm.nih.gov/2021/10/05/updated-pubmed-api/.

R Core Team. 2020. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

———. 2021. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Robert, Christian P, and George Casella. 2004. *Monte Carlo Statistical Methods*. Vol. 2. Springer.

Rousseau, Francois. 2015. "Graph-of-Words: Mining and Retrieving Text with Networks of Features." PhD thesis, Ph. D. dissertation.

Settles, Burr. 2009. "Active Learning Literature Survey."

Therneau, Terry, and Beth Atkinson. 2019. *Rpart: Recursive Partitioning and Regression Trees*. https://CRAN.R-project.org/package=rpart.

Wallace, Byron C, Kevin Small, Carla E Brodley, and Thomas A Trikalinos. 2010. "Active Learning for Biomedical Citation Screening." In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 173–82.

Yang, Li, and Abdallah Shami. 2020. "On Hyperparameter Optimization of Machine Learning Algorithms: Theory and Practice." *Neurocomputing* 415: 295–316.

Zhou, Zhi-Hua. 2021. "Ensemble Learning." In *Machine Learning*, 181–210. Springer.