Text Retrieval Priors for Bayesian Logistic Regression

Eugene Yang
IR Lab, Georgetown University
Washington, DC, USA
eugene@ir.cs.georgetown.edu

David D. Lewis
Cyxtera Technologies
Dallas, TX, USA
sigir2019paper@davelewis.com

Ophir Frieder
IR Lab, Georgetown University
Washington, DC, USA
ophir@ir.cs.georgetown.edu

ABSTRACT

Discriminative learning algorithms such as logistic regression excel when training data are plentiful, but falter when it is meager. An extreme case is text retrieval (zero training data), where discriminative learning is impossible and heuristics such as BM25, which combine domain knowledge (a topical keyword query) with generative learning (Naive Bayes), are dominant. Building on past work, we show that BM25-inspired Gaussian priors for Bayesian logistic regression based on topical keywords provide better effectiveness than the usual L2 (zero mode, uniform variance) Gaussian prior. On two high recall retrieval datasets, the resulting models transition smoothly from BM25 level effectiveness to discriminative effectiveness as training data volume increases, dominating L2 regularization even when substantial training data is available.

CCS CONCEPTS

• **Information systems → Clustering and classification**; • **Computing methodologies → Supervised learning by classification**; **Regularization**; • **Theory of computation →** *Bayesian analysis.*

KEYWORDS

text classification, regularization, ad hoc retrieval, Bayesian priors, Bayesian logistic regression

ACM Reference Format:
Eugene Yang, David D. Lewis, and Ophir Frieder. 2019. Text Retrieval Priors for Bayesian Logistic Regression. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19), July 21–25, 2019, Paris, France.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3331184.3331299

1 INTRODUCTION

Discriminative learning methods such as regularized logistic regression are widely used in applications, such as text categorization, where large amounts of labeled training data are available. When little or no labeled data is available, as in ad hoc retrieval and relevance feedback, heuristics such as BM25, based on domain knowledge (queries) and generative learning (Naive Bayes, for example), are dominant [11]. This dichotomy reflects results showing that generative approaches dominate at small training set sizes, while discriminative ones dominate for large training sets [8].

In high recall retrieval (HRR) tasks such as systematic review in medicine [13] and electronic discovery in the law [2], however, algorithms must deal with training sets of varying size. HRR projects often begin with keyword queries, and build training sets by iterative active learning [1]. A range of training set sizes from zero to handfuls to thousands of examples are encountered during a typical project. A single, simple algorithmic approach that spans all training set sizes is desirable.

Our contribution is a deceptively simple synthesis: logistic regression that regularizes toward the coefficient values of a good text retrieval query (BM25 in our case) rather than toward values of zero as is usual. We describe how this approach builds on previous work in Bayesian logistic regression for text data, and draws on two theoretical interpretations of IDF weighting. We test our approach on two HRR datasets and find that our proposed methods dominate both standard L2 regularization and statistical text retrieval baselines at all training set sizes.

2 BACKGROUND

Regularization—the penalization of solutions that deviate from prior expectations—is a key technique for avoiding fitting to accidental properties of data in supervised learning. A common approach is the so-called L2 penalty, which is proportional to the squares of the coefficients (thus the square of the L2 norm). L2 penalties not only improve generalization, but they aid convergence of optimization algorithms used for fitting to training data [14].

L2 penalties can be given a Bayesian interpretation [4]. Assume a conditional probability model $y = f(\mathbf{x}; \mathbf{w})$ (logistic regression, for instance) parameterized by a $d+1$-dimensional vector of coefficients $\mathbf{w}$ (one per feature, plus an intercept). In the Bayesian framework, we encode our expectations about likely coefficient values as a prior probability distribution.

Suppose that the prior is a product of independent univariate gaussian distributions $N(b_j, {\sigma_j}^2)$, where j ranges over coefficients. Let $D = (\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)$ be a data set where each y value is assumed to be generated by applying $f(\mathbf{x}; \mathbf{w})$ independently to the corresponding $\mathbf{x}$. Bayes Rule then gives this posterior probability distribution:

$$
\begin{aligned}
p(\mathbf{w}|D) &= \frac{p(D|\mathbf{w})p(\mathbf{w})}{p(D)} \\
&= \frac{\left(\prod_{i=1}^{n} p(y_i|\mathbf{w}; \mathbf{x_i})\right)\left(\prod_{j=1}^{d+1} \frac{1}{\sqrt{2\pi {\sigma_j}^2}} e^{-\frac{(w_j-b_j)^2}{2{\sigma_j}^2}}\right)}{p(D)}
\end{aligned}
$$

SIGIR '19, July 21–25, 2019, Paris, France

ACM ISBN 978-1-4503-6172-9/19/07…$15.00
https://doi.org/10.1145/3331184.3331299

where $p(D|\mathbf{w})$ is the conditional probability of seeing the y values given the corresponding $\mathbf{x}$'s, and $p(D)$ is the unconditional probability.

When d is large, as in information retrieval, we typically ask algorithms to produce the MAP (maximum a posteriori) estimate for $\mathbf{w}^*$ and use that as our predictive model. This is equivalent to finding the maximum of this penalized loss:

$$\mathbf{w}^* = \operatorname*{argmax}_{\mathbf{w}} \left\{ \left(-\sum_{i=1}^{n} \ln p(y_i|\mathbf{w};\mathbf{x_i}) \right) + \sum_{j=1}^{d} \lambda_j (w_j - b_j)^2 \right\}$$

where b_j is the prior mode for coefficient j, and λ_j is inversely proportional to the standard deviation of the prior on coefficient j. The usual L2 penalization scheme corresponds to an assumption that all coefficients will be small (prior mode 0), and that belief in that smallness is the same for all coefficients (uniform standard deviation, and thus penalty).

Some studies have relaxed these assumptions. Dayanik, et al [3] used IDF (inverse document frequency values) weights computed from a category corpus to set the mode or standard deviation of Bayesian priors on coefficients of a logistic regression model. In more recent work, we used keyword queries with no category corpus to set modes (only) of Bayesian priors for logistic regression [15].

Several authors have proposed first training a generative model such as Naive Bayes or Rocchio, and then using the resulting coefficients as a prior or parameter space constraint on training a logistic regression model [6, 17]. This method has questionable statistical foundations when applied to a single training set, but is more justifiable when applied to multiple training sets in transfer learning [9].

3 IDF-BASED REGULARIZATION

Modern term weighting schemes such as BM25 [11] were developed to deal with the ultimate low-data scenario: ranked retrieval with no training data. We suggest this provides a simpler alternative than past approaches for incorporating domain knowledge in supervised learning for text analytics problems.

The notion of IDF weighting is key. Justifications for IDF weighting of query terms fall into two major classes [10], and give a new perspective on Dayanik's methods for constructing priors [3]. In probabilistic IR models, IDF weights appear as the coefficients of a generative learning model (Naive Bayes) trained on a blind negative labeling of an entire data set. This assumes that all keywords are positively associated with relevance, and suggests a prior where the modal value of a coefficient is proportional to its IDF.

In contrast, information-theoretic interpretations of IDF view rare terms as being more informative about relevance. This view suggests that less training data should be required to produce a large coefficient for high IDF terms than low IDF terms. In other words, priors for high IDF terms should have a higher variance, regardless of their mode. This translates to a smaller penalty on coefficient magnitude.

Based on these perspectives, we tested the following four schemes for determining priors:

- UPQM: Uniform penalty (i.e., uniform standard deviation of priors) for all coefficients. Prior mode equal to $1 + \log QTF$ where QTF is the number of occurrences of the term in the keyword query. When each term occurs only once, this is identical to Dayanik's **Mode** method [3].
- UPQIM: Uniform penalty. Prior mode equal to a BM25 query. Similar to Dayanik's **Mode/TFIDF** method, but requiring only a single query, not a corpus of queries or category descriptions.
- IIPQM: Inverse IDF penalty: penalty is inversely proportional to IDF. QTF modes.
- IIPQIM: Inverse IDF penalty and BM25 modes.

Using the same notation, we refer to conventional L2 regularization, with a uniform penalty toward zero modes, as UPZM.

Our four methods leave three prior values unspecified: the prior mode and penalty for the intercept coefficient of the logistic regression model, and the base prior penalty for term coefficients. The intercept affects only calibration of the model, not ranking, so for these experiments we used a fixed zero mode prior.

Choosing a base penalty value, however, is needed both as a uniform penalty for UP methods, and to be divided by IDF values in IIP methods. Dayanik, et al chose their base penalty value by using 10-fold cross-validation [3]. This required a minimum of 10 training examples, and arguably was unstable well above that size. Since priors provide their main benefit for small training sets, we eschewed cross-validation and instead explored a range of base penalty values (2^{-24} to 2^{16}) to determine if a plausible default value exists.

Our method is easy to implement, since most existing logistic regression code bases support L2 penalties in their optimization code (albeit only in UPZM form). We modified the existing `sklearn.linear_model.SGDClassifier` package from scikit-learn to support nonzero modes and variable penalties in logistic regression [15]. We provide the modified version on GitHub[1]. The changes to support the penalties used in this paper required only about 30 lines of new or edited code (assuming IDFs are computed externally).

4 METHODS

4.1 Data Sets

We used two test collections drawn from high recall retrieval research: the Jeb Bush Collection and RCV1-v2.

The Jeb Bush Collection consists of email to and from a state governor in the United States [12]. It consists of 290,099 files, of which 274,124 are unique based on MD5 hash. The TREC 2015 and 2016 Total Recall Tracks defined a total of 44 topics on the Jeb Bush data and distributed short titles and labeled data for each [5, 12]. The 2016 labels were ternary, so we treated both "relevant" and "important" as positive and "non-relevant" as negative. To ensure enough positive examples for accurate estimation of effectiveness, our experiments used only the 33 topics with at least 160 positive documents. We used the topic title as our keyword query. This provided from one to five keywords per topic.

RCV1-v2 is a widely used text categorization dataset [7]. It consists of 804,414 newswire stories categorized by professional editors. Of the 823 categories, we chose the 82 categories that had at least

[1] https://github.com/eugene-yang/priorsgd

Table 1: Mean testset R-precision for logistic regression variants with base penalty strength of 1.0 and various training set sizes. Percentages are relative improvements of knowledge-based priors over a uniform prior. With no training data (size 0), UPQM and IIPQM act as QTF queries, UPQIM and IIPQIM act as BM25 queries, and UPZM ranks randomly.

	Jeb Bush Collection (10 replicates)					RCV1-v2 (1 replicate)				
Size	UPZM	UPQM	UPQIM	IIPQM	IIPQIM	UPZM	UPQM	UPQIM	IIPQM	IIPQIM
0	0.5	34.6 (6395%)	38.6 (7133%)	34.6 (6395%)	38.6 (7133%)	5.1	37.1 (630%)	37.9 (645%)	37.1 (630%)	37.9 (645%)
1	13.1	37.9 (190%)	39.3 (200%)	39.3 (200%)	40.3 (208%)	17.6	38.4 (118%)	38.5 (118%)	38.7 (120%)	39.0 (121%)
2	13.1	29.6 (127%)	39.2 (200%)	39.7 (204%)	41.0 (214%)	16.0	34.9 (118%)	38.6 (142%)	39.2 (145%)	39.1 (145%)
4	17.3	33.2 (92%)	40.5 (135%)	41.8 (142%)	43.3 (151%)	22.1	36.1 (64%)	39.1 (77%)	40.2 (82%)	39.7 (80%)
8	22.9	36.9 (62%)	42.2 (84%)	46.4 (103%)	44.9 (96%)	30.1	41.4 (38%)	40.0 (33%)	43.5 (45%)	40.4 (34%)
16	29.8	42.1 (41%)	43.8 (47%)	52.5 (76%)	47.7 (60%)	39.7	47.2 (19%)	41.5 (5%)	49.5 (25%)	41.7 (5%)
32	38.3	47.6 (24%)	46.5 (22%)	58.2 (52%)	50.8 (33%)	49.0	53.3 (9%)	43.7 (-11%)	56.0 (14%)	44.0 (-10%)
64	47.3	53.5 (13%)	49.6 (5%)	62.6 (32%)	54.5 (15%)	57.0	59.0 (4%)	46.8 (-18%)	61.4 (8%)	47.2 (-17%)
128	55.2	59.7 (8%)	54.0 (-2%)	66.2 (20%)	59.8 (8%)	63.0	63.8 (1%)	50.7 (-20%)	66.4 (5%)	52.2 (-17%)

10,000 positive documents with an eye toward future studies of variation across training sets. Each category has a Reuters Business Briefing (RBB) description of between one and seventeen words that we use as our keyword query.

Text processing simply replaced punctuation with whitespace, and then formed tokens at whitespace boundaries. We used BM25 within document weights, i.e. saturated TF weights.

4.2 Evaluation

The impact of prior knowledge depends on training set size. As usual in supervised learning research, we nested smaller training sets in larger ones. Training sets of size 1 consisted of a single randomly selected positive example. Larger training sets (from 2 to 128 documents by powers of two) were balanced 50/50 between random positive and random negative examples, mimicking the balance sought by the active learning algorithms used in high recall retrieval [2]. All training data were drawn from a random 40% of the collection, with 60% reserved for estimating effectiveness.

Variability in effectiveness between training sets is high for small training sets and low richness. To produce more stable results for the Jeb Bush collection we averaged across ten randomly drawn training sets of 128 documents and their included balanced subsets. With the larger number, and higher richness, of categories for RCV1-v2 averaging over replicates was less necessary for stability (and more computationally expensive), and was deferred for future work.

Documents were ranked by logistic regression scores, with ties (common in some conditions) broken by MD5 hash of document ID. We used *R-precision* (precision of documents above a cutoff equal to the number of testset relevant documents) as our effectiveness measure, as is common in HRR research. We computed test set R-precision for each run, averaged it across replicates when used, and then averaged across categories for a given training set size and penalty level.

5 RESULTS AND ANALYSIS

We compare our four methods with two baselines: a BM25 query formed from the keyword query for each topic, and UPZM logistic regression. Text retrieval baselines are rarely used in research on domain knowledge in supervised learning, but should be. As Table 1 shows, BM25 (with effectiveness of 38.6% and 37.9% on Jeb Bush and RCV1-v2 respectively) dominates UPZM until 16 to 32 examples are available.

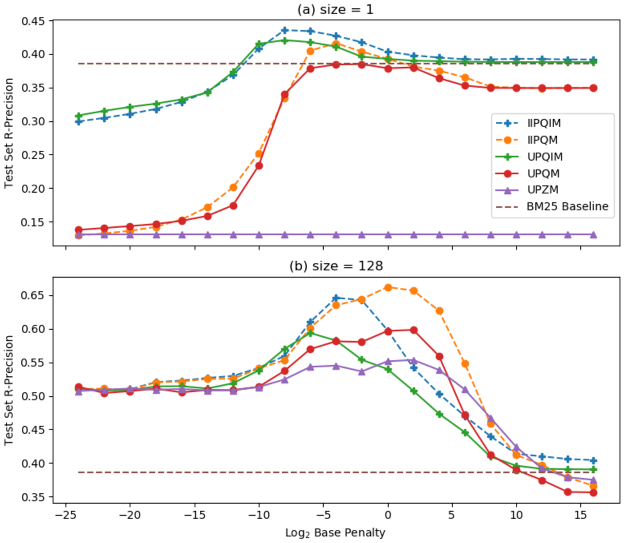

Figure 1: Mean test set R-precision for training sets of size 1 and 128, as a function of the base regularization penalty ($\log_2$ scale) on the Jeb Bush collection.

Table 1 shows the mean R-precision values for each of our four methods plus UPZM, all with a base penalty of 1.0 (Section 5.1). All four IDF-based priors vastly outperform UPZM for small training set sizes. The QM methods, particularly IIPQM, maintain this dominance for all training set sizes. The QIM methods are more skewed, excelling a bit at smaller training set sizes and faltering a bit at larger ones.

5.1 Choosing the Base Penalty

Figure 1 shows the impact of varying the base penalty value for each of the supervised methods, on training sets of size 1 and 128

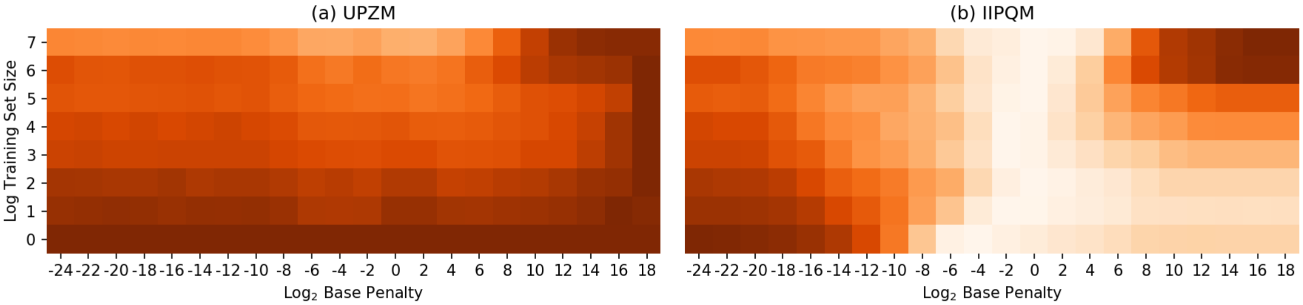

Figure 2: A heatmap display of the interaction between base penalty strength and training set size, both with $\log_2$ scaling, for UPZM and IIPQM on Jeb Bush. Lighter cells indicate higher average R-precision. Colors are scaled row-by-row, so that lighter colors indicate dominance for that training set size across both algorithms and all penalties.

from the Jeb Bush data. We see classical regularization curves with maximum effectiveness at intermediate regularization strengths.

Training on a single positive example provides insight into the methods. At high penalties (right hand side of graph), the example is largely ignored. Our methods converge to their prior modes: a QTF query for QM methods, and a BM25 query for QIM methods. Conversely, at low penalties (left side), the prior mode is largely ignored. IIP methods converge to BM25 querying-by-document [16]. UP methods (including UPZM) converge to unweighted querying-by-document with saturated TF weights. Since we did not omit stopwords or use within-document IDF weighting, the resulting effectiveness is little better than random.

The joint relationship between training set size and base penalty strength is shown in Figure 2. We use a heatmap to compare UPZM and IIPQM for all penalty strengths and training set sizes (on a log base 2 scale) for the Jeb Bush data. Lighter values indicate higher values of mean R-precision. The color scale is normalized *separately for each row* taking both UPZM and IIPQM values into account, since we are interested in relative effectiveness for a given training set size.

We see that UPZM is dominated under almost all conditions. The IIPQM results show that a base penalization in the range 2^{-4} to 2^4 provides good average effectiveness across all training set sizes. Examining the averaged RCV1-v2 data shows a similar "ridge of light." Per-category data on both datasets shows this range of penalties is optimal for most individual categories as well.

6 FUTURE WORK

Our experiments examined the generalization behavior of IDF-based priors under controlled variations in base penalty strength and training set size. Operational HRR scenarios are more complex. First, a variety of active learning algorithms and batch size schemes are used to iteratively generate training sets [2]. Second, an HRR dataset is both the source of training data and the target of prioritization, so that documents that are labeled no longer need to be scored. Both factors will require experimentation to understand in full.

The success of inverse IDF regularization penalties when keywords are available suggests using this technique even without prior knowledge. IDF weighting has been used in application areas as diverse as images, video, music, and genomic data, so the technique may have broad applicability.

7 CONCLUSION

Regularized logistic regression is a standard workhorse for machine learning, but has faltered when applied to tiny training sets. We show that its effectiveness can be improved under all conditions, and vastly for small training sets, by IDF-based priors.

REFERENCES

[1] Mustafa Abualsaud, Nimesh Ghelani, Haotian Zhang, Mark D Smucker, Gordon V Cormack, and Maura R Grossman. 2018. A System for Efficient High-Recall Retrieval.. In *SIGIR*. 1317–1320.

[2] Gordon F. Cormack and Maura F. Grossman. 2014. Evaluation of machine-learning protocols for technology-assisted review in electronic discovery. *SIGIR 2014* (2014), 153–162. https://doi.org/10.1145/2600428.2609601.

[3] Aynur Dayanik, David D Lewis, David Madigan, Vladimir Menkov, and Alexander Genkin. 2006. Constructing informative prior distributions from domain knowledge in text classification. In *SIGIR 2006*. ACM.

[4] Alexander Genkin, David D. Lewis, and David Madigan. 2007. Large-Scale Bayesian Logistic Regression for Text Categorization. *Technometrics* 49, 3 (Aug. 2007), 291–304. https://doi.org/10.1198/004017007000000245

[5] Maura R. Grossman, Gordon V. Cormack, and Adam Roegiest. 2016. TREC 2016 Total Recall Track Overview.

[6] David D. Lewis and William A Gale. 1994. A sequential algorithm for training text classifiers. In *SIGIR 1994*. 3–12.

[7] David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. 2004. RCV1: A New Benchmark Collection for Text Categorization Research. *JMLR* 5 (2004), 361–397.

[8] Andrew Y Ng and Michael I Jordan. 2002. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in neural information processing systems*. 841–848.

[9] Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22, 10 (2010), 1345–1359.

[10] Stephen Robertson. 2004. Understanding inverse document frequency: on theoretical arguments for IDF. *JDoc* 60, 5 (2004), 503–520.

[11] Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *F&T in IR* 3, 4 (2009), 333–389.

[12] Adam Roegiest and Gordon V. Cormack. 2015. TREC 2015 Total Recall Track Overview. (2015).

[13] Harrisen Scells and Guido Zuccon. 2018. Generating Better Queries for Systematic Reviews.. In *SIGIR*. 475–484.

[14] Shai Shalev-Shwartz and Shai Ben-David. 2014. *Understanding machine learning: From theory to algorithms*. Cambridge university press.

[15] Eugene Yang, David D. Lewis, and Ophir Frieder. 2019. A Regularization Approach to Combining Keywords and Training Data in Technology-Assisted Review. In *ICAIL 2019*. Montreal, Canada.

[16] Eugene Yang, David D. Lewis, Ophir Frieder, David Grossman, and Roman Yurchak. 2018. Retrieval and Richness when Querying by Document. *DESIRES* (2018).

[17] Yi Zhang. 2004. Using bayesian priors to combine classifiers for adaptive filtering. In *SIGIR 2004*. ACM, 345–352.