Online Stratified Sampling: Evaluating Classifiers at Web-Scale

Paul N. Bennett*
Microsoft Research
One Microsoft Way, Redmond WA USA
pauben@microsoft.com

Vitor R. Carvalho
Microsoft, Inc.
One Microsoft Way, Redmond WA USA
vitor@microsoft.com

ABSTRACT

Deploying a classifier to large-scale systems such as the web requires careful feature design and performance evaluation. Evaluation is particularly challenging because these large collections frequently change. In this paper we adapt *stratified sampling* techniques to evaluate the precision of classifiers deployed in large-scale systems. We investigate different types of stratification strategies, and then we derive a new online sampling algorithm that incrementally approximates the theoretical optimal disproportionate sampling strategy. In experiments, the proposed algorithm significantly outperforms both simple random sampling as well as other types of stratified sampling, with an average reduction of about 20% in labeling effort to reach the same confidence and interval-bounds on precision.

Categories and Subject Descriptors

H.3.4 [**Systems and Software**]: Performance evaluation (efficiency and effectiveness)

General Terms

Algorithms, Design, Experimentation

Keywords

Stratified sampling, classification, web scale

1. INTRODUCTION

Evaluating classifiers has long been an important topic of investigation [2, 15, 10, 3, 5]. Within information retrieval, classifiers have been useful for a variety of tasks including routing, web junk & spam identification, accelerated searching, and filtering. However, to deploy classification technology within a larger retrieval system, it is important to bound the performance of the classifier with high confidence.

Furthermore, the underlying collection is changing and this necessitates ongoing checks to determine that the classifiers fall within tolerable performance ranges as determined by the system designers. Additionally, classifiers might be trained over data that represent a different distribution than that of the whole collection. For example, the web directory ODP [11] has often been used for building text classifiers, however it is unclear whether the distribution of the URLs in ODP is a representative sample of the web. These differences often lead to questions of how the classifier will perform *in the wild*—that is, in a setting similar to its deployment.

Here we demonstrate how to efficiently bound the precision of a classifier using minimal amounts of labeled data by adapting the techniques of stratified sampling [13, 9] to the problem of classifier evaluation. In particular, we demonstrate that the output score of the classifier serves as a good basis for stratification by identifying regions of similar classifier behavior because of the typical monotonic relationship between classifiers and the true class-conditional posterior [2]. Given a stratification into regions or strata, there is an optimal strategy [13, 9] for distributing the number of samples across these strata. However, this optimal strategy relies on knowing the variance of the classifier within each stratum which would preclude the need for evaluation.

We develop a novel algorithm that draws samples according to the current allocation estimate, refines variance estimates and consequent allocation estimates based on these samples, and then iterates until a convergence criterion is satisfied. This algorithm is "online" in the machine learning sense that the feedback it has received so far via the label judgments influences the next set of samples that are drawn. Our algorithm exploits this feedback to draw a total number of samples that are near-optimal given a chosen stratification strategy.

2. RELATED WORK

Stratified sampling has been employed in evaluation in different areas of science [8, 6, 17, 7, 4]. Most similarly, stratified sampling has been used in information retrieval to estimate different ranking algorithms' performance on the very large evaluation problem of the Million Query track [14, 1]. Our work differs from theirs in several ways. First, they focus on ranking algorithms and average precision where we focus on classification algorithms and precision above a threshold. Since average precision averages the precision at each positive example, each identified positive example contributes in a different way to average precision (1/(num relevant) * (precision at point)) than it does to precision (1 / (num predicted positive)). For example, suppose when sorted by classifier score, the first 5 examples are relevant (positive), and the remaining are all irrelevant (negative),

*Authorship alphabetical.

CIKM'10, October 26–30, 2010, Toronto, Ontario, Canada.

but the top 200 are predicted to be relevant. Any stratification method that samples the top heavily will quickly converge to the correct estimate of average precision (1.0) but to estimate precision requires sampling the remaining irrelevant examples in the top 200 to know with high confidence that the precision lies near 0.025. Thus, it is unclear how gains in estimating either of these translates to the other. Furthermore, we explore several stratification strategies applicable to any classifier which produces a score and analyze the properties that are key to generalizing our methods to other classification scenarios. Additionally, our approach is an iterative method that can make use of labeled data as it is collected to improve the choice of which examples should be labeled next and allows the user to specify a desired significance level and interval width for the estimate.

3. MOTIVATION

In this work, we focus on the precision of classifiers although our techniques can be generalized to other measures. We assume that the number of items to which the classifier might be applied is quite large (*e.g.*, all web URLs indexed by a search engine) and while the percentage of time the classifier predicts positive may be small, the absolute number of predicted positives is still large. Our goal is to identify a method that is both computationally efficient (subquadratic) and label-effort efficient (significantly better than random sampling) in these large-scale conditions.

4. SAMPLING OVERVIEW

In this paper, sampling is used to select elements (web pages) to be labeled by human judges. While we present formula for both cases, sampling with replacement is simpler in both formulation and in edge cases in implementation. Additionally, in the large scale, there is a vanishingly small probability the same item will be selected. Thus, we use sampling with replacement in the empirical study.

In basic random sampling, one uses the mean, $\widehat{p}$, to estimate the true proportion of elements belonging to a class C after sampling n elements from a population of N where each element e_i has an associated binary class variable y_i defined as: $y_i = 1$ if $e_i \in C$, and $y_i = 0$, otherwise. The variance of $\widehat{p}$ can be shown [9] to be $var(\widehat{p}) = (1-f)\frac{\widehat{p}(1-\widehat{p})}{n-1}$ where the *finite population correction* (fpc) factor, $(1-f) = (1-\frac{n}{N})$, is only used when sampling without replacement.

Stratified sampling is the process of dividing the population into different disjoint strata, sampling from each stratum separately, and then combining the different stratum-based estimates to produce a combined estimate for the entire population [9, 13]. This process often produces more accurate population estimates when the subpopulation data are more homogeneous, *i.e.*, less variance within stratum.

Formally, for a class C and a population of N elements that can be split into K disjoint subpopulations, each with N_k elements such that $\sum_k N_k = N$, one can estimate $\widehat{p}$ and its variance as a weighted average of the subpopulation estimates [9]. This approach yields:

$$\widehat{p} = \sum_{k=1}^{K} \frac{N_k}{N}\widehat{p}_k \qquad var(\widehat{p}) = \sum_{k=1}^{K} (\frac{N_k}{N})^2\, var(\widehat{p}_k) \qquad (1)$$

where $\widehat{p}_k$ and $var(\widehat{p}_k)$ are the sample proportion mean estimator and corresponding variance for subpopulation k – which can be calculated as in basic random sampling by selecting n_k samples from subpopulation k. The standard error (*s.e.*) is $s.e.(\widehat{p}) = \sqrt{var(\widehat{p})}$.

To apply stratified sampling, one must choose the total number, n, of samples to label and the allocation into the n_k for each strata such that $\sum n_k = n$. This leads to the question of whether there is an optimal strategy for allocating between the strata. There is, in fact, and it depends on the relative distribution of the population and *variance* across the strata. When sampling n points, the optimal strategy is to sample $n_k \propto N_k * \sigma_k$ which is commonly referred to as disproportionate optimal sampling in the literature [13]. However this requires knowing the standard deviation within a subpopulation, σ_k.

5. ONLINE STRATIFIED SAMPLING

In order to approximate optimal sampling we must design a way to incorporate the estimation of the standard deviations (which requires having labeled samples) into the sampling process where we are obtaining labeled samples – a chicken and the egg problem. Furthermore, we must identify a property that can be used to stratify classifiers in general. Finally, we need a reasonable stopping criterion to indicate when we have sampled a sufficient number of points. In this section, we introduce a novel approach that proposes solutions to each of these research problems.

While the classifier may be calibrated [12, 2, 16] over the training set, because of the possible divergence in train and test distributions, it is unreasonable to assume the classifier is calibrated over the test distribution. However, in the literature on calibration [12, 2, 16], it is noted that nearly every classifier produces a score or probability of which the true posterior tends to be a monotonic function. Assuming this weaker property holds, stratifying into contiguous classifier score intervals will break the distribution into segments that are more homogeneous on average than the overall distribution. Hence, we obtain a ready-made stratification approach that is applicable to any classification model. Thusforth, we use "bin" interchangeably with "stratum".

We can break the interval between the max and the minimum scores into equal ranges (called *equal* for "equal score"). Alternatively, we break the interval so each bin covers an equal portion of the distribution (*e.g.*, quartiles when using four bins) which we term the *perc* method for "percentile". The obvious difference between these two conditions is that while the first has bins that cover an equal range, the balance of the number of points falling in each bin can be highly skewed since points are not uniformly distributed over the score range typically. By studying both, we can observe the effect of balanced strata to skewed strata in terms of size.

If we are particularly "unlucky" a small sample containing all positives or negatives might give rise to an estimate of $\widehat{p}_i$ of 0 or 1. To reduce the impact of this, we use a common Bayesian smoothing technique call m-estimate smoothing. Given h_k "positive" outcomes out of n_k trials, the m-estimate is $\hat{p}_k = \frac{h_k + m p_k}{n_k + m}$ where p_k is a prior and m corresponds to the weight of "virtual examples" that will be given to the prior. We use $p_k = 0.5$ and m is 2 if $n_k = 0$ and $1/\sqrt{n_k}$ otherwise. By using an adaptive m that decreases at the same rate that the standard error decreases, we find the effect to be smooth and consistent near the boundaries.

To determine a stopping criterion, we allow a user to request a significance level, α, and interval radius, δ, such that

they want to bound the true precision within $\pm\delta$ of the estimate with (1-α) confidence. Given these parameters and the standard error ($s.e.$) of an estimator, it is straightforward to use the normal approximation for confidence intervals to derive a significance threshold for stopping.

To avoid a single spurious false positive, we require this criterion be met for $T = 2$ rounds in a row.

Finally, to estimate the optimal allocation, we run an iterative or "online" algorithm starting with no labeled data. At each iteration, more labeled data becomes available after the identified samples have been labeled, and that data is used to determine the next samples that will be labeled. At any given iteration, we use all the labeled data we have sampled thus far to compute smoothed subpopulation estimates and use these terms to estimate the optimal allocation. We then draw a small number of samples according to this allocation distribution, observe their labels, and iterate until convergence. The full algorithm is given in Figure 1.

given
  K bins
  D_0 (number of samples for initial draw)
  D (total number of samples at each step)
  α (Desired significance level, 0.05 for 95% confidence)
  δ (With $1-\alpha$ confidence estimate within $\pm\delta$ of truth)
  T must satisfy confidence criteria in T sequential rounds
Calculate smoothed $\hat{p}_k$ and σ_k
Draw n_k where $\sum n_k = D_0$ from Mult($N_1\sigma_1, \ldots, N_K\sigma_K$)
$\forall$k Sample n_k points uniformly from stratum k
t := 0
do
  Calculate smoothed $\hat{p}_k$ and σ_k
  Draw n_k where $\sum n_k = D$ from Mult($N_1\sigma_1, \ldots, N_K\sigma_K$)
  $\forall$k Sample n_k points uniformly from stratum k
  Calculate overall precision $\widehat{p}$ and $var(\widehat{p})$
  if (MeetsConfidenceCriterion($\sqrt{var(\widehat{p})}$,δ,α))
    t++
  else
    t := 0
while (t < T)
Return $\widehat{p}$ and $var(\widehat{p})$

Figure 1: Incremental Sampling Algorithm.

We call the approach that draws the n_k in each round as specified in Figure 1 the *opt* style of stratified sampling. In order to isolate the effects of binning on the estimate versus the allocation of samples among bins, we also consider other allocation approaches. For example, if we allocate n_k proportional to N_k then we get a method that allocates solely based on the proportion of instances falling in a stratum. This allows us to isolate any gain achieved through estimation of the variance. We term this style of sampling *pps* for "proportional sampling" (as it is commonly referred to in the literature [13]). Likewise we can also allocate n_k uniformly over all of the strata. This allows us to isolate any gain solely achieved due to the binning versus the allocation procedure. We term this style of sampling *uni*. Finally, we can implement a simple random sampler as analogously as possible by simply having a single bin. We term this *Random* and consider this the default baseline. All approaches with more than one bin we refer to as *Strat* below for stratified.

6. EXPERIMENTS

We trained classifiers over a subset of the top-level categories of ODP. A classifier for each topic was trained using a binary logistic regression model with regularization and standard tfidf settings over a crawl of ODP from early 2008. The data consist of approximately 1.7M documents. We split the data into a 70%/15%/15% train/validation/test split. We estimate the precision of each binary classifier over the test set and report averages.

Our primary goal is the "Savings" of labeled examples attained with respect to the baseline of random sampling. We run each experiment 1K times and record the average number of examples needed to converge as well as the standard error of the number of examples to judge significance. We also examine: (1) the percent change "%Change" in number of examples relative to the baseline where an increasingly negative change is good; (2) the proportion of time the true precision falls within the requested $\pm\delta$ of the true precision, *InConf* – this number is expected to be close to 1 - α.

We can also use the standard normal confidence interval approach to estimate the number of samples needed if we had full knowledge of the true precision overall and in each bin beforehand. This serves as a reasonable approximate lower bound that is conditioned on the method's binning. We present the percent change with respect to this oracle as "%Oracle" where a lower number is desired.

Here, α is 0.05. To keep the number of samples per bin constant, we allow each method a "budget" of two samples per bin. This favors the baseline since a method can only halt between iterations. We set the number of bins to $K = 4$.

	InConf	%Change	%Oracle	Savings
Random	0.94	0.0	-0.6	0
strat, equal, uni	0.95	133.1	145.3	-92249
strat, equal, pps	0.95	-4.2	0.8	2921
strat, equal, opt	0.94	-13.2	-8.6	9148
strat, perc, uni	0.95	-7.6	10.1	5274
strat, perc, pps	0.95	-7.6	10.1	5273
strat, perc, opt	0.93	**-17.3**	-1.4	**11973**

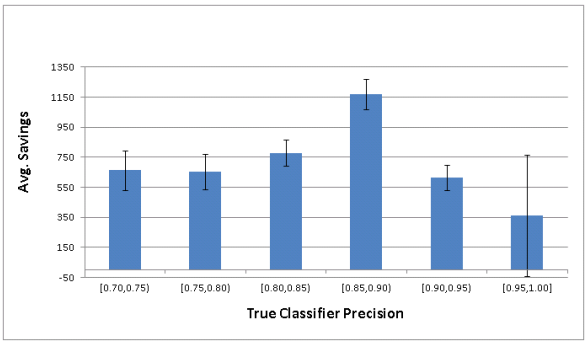
Table 1: Average over 15 top-level ODP categories.

Results & Discussion

Table 1 presents the results over the ODP data set when using a bound on the estimate of $\delta = 0.01$. This was chosen for the tightness of bound often needed for classifiers in a production environment. We have only indicated statistical significance (*underline*) for the best method relative to the baseline to focus on the key research question.

First, note that equal score stratification with uniform sampling has extremely poor performance. Equal score stratification creates vastly skewed strata subpopulation sizes — typically with the most populous bin near the classifier threshold and the least populous bin at the most confident end. Sampling uniformly among these bins means that far too many samples are wasted on the small bins where estimates converge more rapidly.

Correcting for skewed bin size by either stratifying by percentile or sampling proportionally achieves moderate gains. However, because the bins are created by using the classifier score, the variance in the subpopulations can range widely. Thus, stratification using proportional sampling is not able to perform as well as the optimal sampling that weights by both estimated variance and subpopulation size.



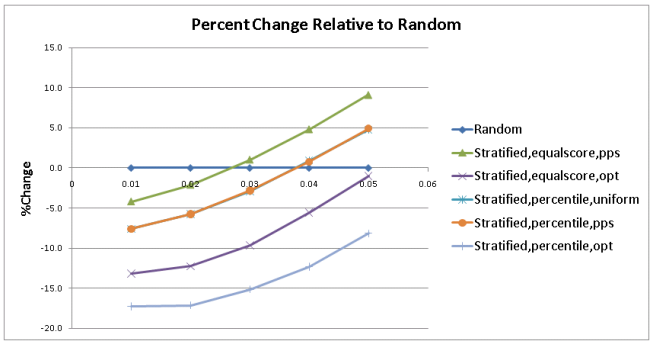


Figure 2: (a) Average savings binned by per-class precision. (b) Impact of varying $\pm\delta$ bound on efficiency.

Finally, when comparing each method using equal score stratification to its percentile counterpart, we see that the percentile method typically works better. This is for two reasons. First, as mentioned above, the percentile binning creates equal subpopulation sizes which is a simpler case, but second, even though ODP is large, the small size of some particular bins means that those bins have a relatively small number of examples in them (hundreds or thousands). This number is small enough that the fpc factor (see Section 4) comes into play in these cases and sampling within these bins should be done by sampling without replacement to be most efficient. Understanding this point is important if one wishes to apply the same technique to other non-uniform strata such as stratifying the output of a decision tree where the subpopulation sizes may be very non-uniform.

With respect to *InConf*, we note that nearly all of the methods are close to the expected value of 0.95. Overall, the best performing sampling method regardless of stratification choice is optimal disproportionate sampling, and within those, the percentile stratification with optimal disproportionate sampling performs the best for the reasons noted above. Additionally noteworthy is that fact that the optimal methods allocate quite closely to the total number of examples that would be allocated by the oracle method while staying in the expected range of *InConf*. Recall that the oracle depends on properties of the strata. Thus, both optimal methods can be near their respective oracles while the percentile binning outperforms equal in the absolute sense.

Next, we seek to understand how much reduction in labeling cost can be expected based on classifier properties. Since it is the best performing method, we focus on the percentile stratified disproportionate optimal sampling. By examining the correlation between precision and the decrease in *relative* labeling cost on a per class basis, we see that the more precise the classifier the larger the decrease. This relationship is very strong (0.88 negative correlation). Indicating classifier precision is an important property for reduction in labeling cost using stratification. From Figure 2(a) we see that in terms of *absolute* reduction in labeling cost, the peak actually starts declining after a point. Considering the relationship of the normal confidence interval estimate to standard error, the absolute number of labeled examples needed declines as the precision approaches 1. Thus, even though our relative reduction is high, the absolute savings starts declining. Likewise, since the absolute number of labeled examples needed increases as precision goes to 0.5, the absolute savings remain reasonable even though our relative gains are shrinking because of the decrease in signal from the classifier for stratification.

Additionally, Figure 2(b) demonstrates the impact of different choices in the bound on the estimate, δ, in the ODP data set. In particular, as the user requires a looser bound (increasing δ), simple random sampling becomes increasingly competitive although the opt methods retain an edge.

7. CONCLUSIONS

In this paper we demonstrated how to adapt *stratified sampling* techniques to evaluate the precision of classifiers deployed in large-scale systems. The version of this method that bins by equal percentiles demonstrated strong and significant reductions in the number of labeled examples required across a variety of parameter settings and classification problems. In addition, in comparison with the expected number of examples required to achieve similar bounds even given an oracle like knowledge of how to sample, the method comes within a few percentage points of optimal allocation. Overall, the method provides a scalable and efficient approach to evaluating classifiers over large-scale data.

8. REFERENCES

[1] J. Allan, B. Carterette, J. A. Aslam, V. Pavlu, B. Dachev, and E. Kanoulas. Million query track 2007 overview. In E. M. Voorhees and L. P. Buckland, editors, *The Sixteenth Text REtrieval Conference Proceedings (TREC 2007)*. National Institute of Standards and Technology, December 2008. NIST Special Publication SP 500-274.

[2] P. N. Bennett. Using asymmetric distributions to improve text classifier probability estimates. In *SIGIR '03*, 2003.

[3] J. Carletta. Assessing agreement in classification tasks: the kappa statistic. *Computational Linguistics*, 22(2):249–254, 1996.

[4] S. Chaudhuri, G. Das, and V. Narasayya. Optimized stratified sampling for approximate query processing. *ACM TODS*, 32(2), 2007.

[5] G. Cormack and T. Lynam. Online supervised spam filter evaluation. *ACM TOIS*, 25(3), 2007.

[6] P. Dixon, A. Ellison, and N. Gotelli. Improving the precision of estimates of the frequency of rare events. *Ecology*, 86(5), 2005.

[7] S. Fernandes, C. Kamienski, J. Kelner, D. Mariz, and D. Sadok. A stratified traffic sampling methodology for seeing the big picture. *Computer Networks*, 52:2677–2689, 2008.

[8] X. He, L. Duan, Y. Zhou, and B. Dom. Threshold selection for web-page classification with highly skewed class distribution. In *WWW '09*, 2009.

[9] L. Kish. *Survey Sampling*. John Wiley & Sons, Inc., 1965.

[10] D. D. Lewis. Evaluating and optimizing autonomous text classification systems. In *SIGIR '95*, 1995.

[11] Netscape Communication Corporation. Open directory project. http://www.dmoz.org.

[12] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In A. J. Smola, P. Bartlett, B. Scholkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*. MIT Press, 1999.

[13] S. K. Thompson. *Sampling*. Wiley-Interscience, 2002.

[14] E. Yilmaz, E. Kanoulas, and J. A. Aslam. A simple and efficient sampling method for estimating AP and NDCG. In *SIGIR '08*, 2008.

[15] B. Zadrozny. Learning and evaluating classifiers under sample selection bias. In *ICML '04*, 2004.

[16] B. Zadrozny and C. Elkan. Reducing multiclass to binary by coupling probability estimates. In *KDD '02*, 2002.

[17] T. Zseby. Stratification strategies for sampling-based non-intrusive measurements of one-way delay. In *Passive and Active Measurement Workshop (PAM 2003)*, 2003.