CALIBRATION FOR NATURAL LANGUAGE PROCESSING SYSTEMS

An Honor Thesis

Presented by

KHANH X. NGUYEN

Approved by:

____

Brendan O'Connor, School of Computer Science

____

Erik Learned-Miller, School of Computer Science

ABSTRACT

Title: **Calibration for natural language processing systems**

Author: **Khanh X. Nguyen**

Thesis/Project Type: **Project**

Approved By: **Brendan O'Connor, School of Computer Science**

Approved By: **Erik Learned-Miller, School of Computer Science**

Statistical natural language processing (NLP) models assign a posterior distribution to the set of possible outcomes and make their predictions based on those scores. Current performance metrics for NLP systems only take into account the model final decisions, which not only depends on the quality of the systems model but also the inference scheme, rather than explicitly reflect the quality of the prediction posterior distribution. In this thesis, I propose a metric for directly examining the quality of the posterior distribution, the calibration test. First of all, I will present the theoretical foundation for the concept of calibration. After that, I apply the test to numerous families of NLP models and show that the calibration is complementary to the traditional metrics in the sense that it provides a more comprehensive insights into the performances of NLP systems.

CHAPTER 1

INTRODUCTION

The ambiguity of natural language motivated the use of the probabilistic model for tackling natural language processing (NLP) tasks. Typically, a probabilistic model produces a posterior distribution over the space of all possible output labels for the input. The probability of an output label can be interpreted as the degree of belief that the model holds for the occurence of that possibility. The model then makes its final decisions based on a decision-making scheme. In this setting, the posterior distribution output by the model is unknown to the users. In addition, performance metrics such as accuracy score or F1 score also solely measure how to the final decisions align with the true labels and neglected the quality of the predicted posterior distribution.

Many NLP systems use maximum-a-posteriori (MAP), i.e. choosing the most likely outcome, as their decision-making scheme. In complex NLP pipelines, however, MAP inference seems to be inappropriate since downstream models suffer from errors accumulated from upstream models. Using the K-top-most-likely predictions was suggested and shown to yield better performance for several NLP tasks (Sutton and McCallum, 2005; Wellner et al., 2004). Finkel et al. (2006) proposed a more efficient alternative to K-best list, using Monte Carlo sampling to an approximate posterior distributions of upstream models and passed the samples as input for downstream models. Those approaches does not examine the quality of the posterior disbutions and assume that they are reliable. This is problematic in practice since most NLP models are imperfect. As a result, the full potentials of those approaches would not have been recognized.

Knowledge about the uncertainty of predictions is also useful for users of NLP systems. Communicating the uncertainty of a model with its users is essential for calculating risk. Consider two predictions for a binary variable: one gives a confidence score of 0.51 that the variable is 1 and and the other assigns a score of 0.8 for the same event. Using a MAP decision-making scheme with a threshold of 0.5, the user will receive identical final predictions reporting that the value for the variable is 1 and will trust them equally. However, if they knew the true confidence scores, we would expect that they would trust the former predictions less than the latter one. This type of situation is common for business users of NLP systems. Prediction confidence score are vital for them to calculate long-term revenue and make investment decisions.

Following BLAH BLAH, I propose a calibration-refinement framework for assessing the quality of posterior distributions of NLP models. Refinement measure is often encountered in the form of log loss or mean squared error. On the other, calibration measure receives less attention although it is complementary to refinement measure. The focus of this thesis is to develop a general procedure for measure calibration that is independent of the choice of model. The procedure takes a posterior distribution and the true data labels as input. The output can either be a visualizable calibration plot or a single calibration score depending on the need of the user.

In order to develop such procedure, first of all, a review the theoretical foundation of calibration is present. Several desirable characteristics of a well-calibrated model are shown. Next, a procedure for conducting calibration test on a model is described in details. Finally, the procedure will be applied to several families of NLP models. I show that BLAH BLAH BLAH.

CHAPTER 2

SUMMARY OF WORK OF PREVIOUS RESEARCHERS

The problem of miscalibration in prediction models that employ single-best inference scheme was addressed by Draper (1995). A Bayesian approach was proposed as an alternative. Finkel et. al (2009) apply this idea to tackle the problem of cascade NLP models. In this approach, the prediction posterior distribution of one task, which is approximated by a sample of the distribution, was passed as the input for other tasks. As suggested by the author, it is more general but easier to implement than approaches using K-best list (Sutton and McCallum (2005), Wellner et al. (2004), Huang and Chiang (2005), Toutanova et al. (2005)).

The concept of calibration was developed in the field of meteorology (Miller (1962), Murphy (1973)), referred to as validity or reliability. In Rubin (1982), it was argued that the applied statistician should Bayesian in principle and calibrated to the real world in practice. Murphy and Winkler (1982) proposed a general framework for forecast verification based on the joint distribution of forecasts and observations. They showed that the joint distribution of predictions and observations contains all of the information needed for assessing the forecast quality. They investigated it through two its Bayesian factorizations: the calibration-refinement factorization and the likelihood-base rate factorization. Their study is general in the sense that it can applied to any other type of prediction that produces a joint distribution between prediction labels and true labels. Rubin (2006) presented a method for validating software for Bayesian models using posterior quantiles. This idea will be applied in my thesis to formulate the notion of calibration for prediction problems where the predicted variable is continuous.

TODO: work on metrics for machine learning.

CHAPTER 3

EXPLANATION OF CURRENT METHODOLOGY AND GOALS

3.1 Background

3.1.1 Calibration-refinement framework

Throughout this paper, we will consider a binary prediction problem, where each to-be-predicted instance can either be labeled positive (denoted by 1), or negative (denoted by 0). A probabilistic model for this problem assigns each instance i a *prediction* $q_i \in [0, 1]$, which represents the confidence level that the instance is in the positive class. After the model makes their predictions, the set of true observations will be given for model assessment. The true observation for instance i is denoted by $y_i \in \{0, 1\}$.

Let $S = \{(q_1, y_1), (q_2, y_2), \cdots, (q_n y_n)\}$ be a set of prediction-observation pairs produced by a probabilistic model. Follow Murphy and Winkler (1984), we assume that the elements of S are drawn from a hypothetical joint distribuion $P(y, q)$, where y and q are the random variables for the prediction and the true label, respectively. $P(y, q)$ contains all the information needed for analyzing the quality of the predictions. The *calibration-refinement framework* is based on the Bayesian factorization of $P(y, q)$:

$$P(y, q) = P(y \mid q)P(q) \tag{3.1}$$

The conditional probability $P(y = 1 \mid q)$ is called the *realistic frequency* with respect to the prediction value q. It indicates how often the true label turns out to be positive among all instances that are predicted to be positive with a confidence level of q. A model is said to be *perfectly calibrated* (or perfectly reliable) if its predictions match with their realistic frequencies for all confidence levels. A more formal definition of perfect calibration is presented in section 3.1.2. On the other hand, the marginal distribution $P(q)$ reflects a model's refinement. A model is said to be *refined* (or sharp) if $P(q)$ concentrates about 0 and 1. This characteristic indicates that the model is capable of discriminating instances from the positive class from instances from the negative class.

For a more concrete view of the calibration-refinement framework, consider a classic example: *precipitation forecast*. In this task, the forecaster is required to give an assessment on the likelihood of precipitation of each single day in a period of time. If a reliable forecaster give a prediction such as "There is 30% chance that it will rain tomorrow", we should expect that among all the days on which that type of prediction is announced, exactly 30% of them will be rainy. Moreover, we should also expect the same condition to hold for all types of predictions (between 0 and 1). However, a reliable forecaster is not always a "good" predictor. Consider the scenario when a forecaster always predicts the climatological probability, i.e. the long-term frequency of precipitation, for any day. The forecaster will be perfectly calibrated but his or her predictions would be useless for regions where the climatological likelihood of raining and not raining are equally likely. In those cases, such unrefined predictions imply a lot of uncertainty and do not help with finalizing binary decisions.

As we can see, maintaining calibration allows posterior predictions to be more realistic whereas having refinement in predictions reduces uncertainty in the decision-making process. Hence, calibration and refinement are orthogonal and complementary concepts.

3.1.2 Definition of perfect calibration

Consider the set of prediction-observation pairs S defined in the previous section.

Definition 3.1. Given a value q between 0 and 1, inclusively, the *realistic frequency* with respect to q, denoted by p_q, is defined as:

$$p_q = P(y = 1 \mid q) = \lim_{n\to\infty} \frac{\sum_{i=1}^{n} I\{q_i = q\} I\{y_i = 1\}}{\sum_{i=1}^{n} I\{q_i = q\}}$$

where $I\{.\}$ is the indicator function and $(q_i, y_i) \in S$.

Definition 3.2. A set of prediction-observation pairs S is said to be *perfectly calibrated* if and only if:

$$p_q = q \qquad \forall q \in [0, 1].$$

3.1.3 Measuring miscalibration

When a model does fulfill definition 3.2, we say that it is *miscalibrated.* The notion of miscalibration is more interesting to study than perfect calibration since most NLP models fall into this category. It is a natural tempting to devise a metric that quantifies miscalibration. Following DeGroot and Fienberg (1982), we introduce concepts that are necessary for constructing such a metric.

Definition 3.3. Let p be real number in $[0, 1]$. A *strictly proper scoring rule* specified by an increasing function $g_1(x)$ and a decreasing function $g_2(x)$ is a function of x that has the following form:

$$f(x) = pg_1(x) + (1 - p)g_2(x) \tag{3.2}$$

and satisfies that $f(x)$ is maximized only at $x = p$.

Theorem 3.4. If $g_1(x)$ and $g_2(x)$ specify a strictly proper function rule, the overall score S for predictions for a probabilistic predictive model can be expressed in the form $S = S_1 + S_2$, where

$$\begin{aligned} S_1 &= E_q \left[p_q \left(g_1(q) - g_1(p_q)\right) + (1 - p_q) \left(g_2(q) - g_2(p_q)\right)\right] \\ S_2 &= E_q \left[p_q g_1(p_q) + (1 - p_q)\, g_2(p_q)\right] \end{aligned} \tag{3.3}$$

It can be proved that S_1 and S_2 have the following properties:

1. S_1 is zero only for perfectly calibrated model and negative otherwise.
2. If two model A and B are both perfectly calibrated and A is at least as sharp as B, the value of S_2 will be at least as large for A as it is for B.

Choosing $g_1(x) = (x-1)^2$ and $g_2(x) = x^2$, S_1 becomes the expected mean squared error between probabilistic predictions and the corresponding realistic frequencies:

$$CalibMSE = E_q[p_q - q]^2$$

We will refer to this quantity by the *MSE calibration score* or simply calibration score, interchangeably.

3.2 Practical calibration analysis

The realistic frequency p_q defined in section 3.1.2 is an unknown quantity. Therefore, the true value of the MSE calibration score cannot be calculated exactly. A general approach for this problem is to replace p_q in the score's formula by an approximation computed from data. We will describe adaptive binning as a simple method for doing it.

Parametric regression is not an appropriate choice since it would not be flexible enough for exloring different models' calibration patterns. Conversely, non-parametric methods only impose weak assumptions on the model choice but still gives close approximation.

3.2.1 Adaptive binning procedure

Adaptive binning is a modified version of *regressogram* (CITE Wasserman 2006). Instead of dividing the interval $[0, 1]$ into equally spaced like regressogram does, adaptive binning assigns an equal number of data points to each bin. This is advantangeous in the context of assessing NLP models, where the distribution of predictions is often skewed toward 0 and 1. Adaptive binning ensures that the mid-range approximations have roughly the same standard errors as those near the boundaries.

Concretely, the adaptive binning procedure is described as follows:

Data: A set of n data points $\{(q_1, y_1), (q_2, y_2), \cdots, (q_n, y_n)\}$.

Parameter: bin size b, the number of points in each bin.

Step 1: Sort the data points by q_i in ascending order.

Step 2: Label the k^{th} data point in the sorted order by $\lfloor \frac{k-1}{b} \rfloor + 1$.

Step 3: Put all the points that have the same label in one bin. If the last bin has size less than b, merge it with the second last bin (if exits). Let $\{B_1, B_2, \cdots, B_T\}$ be the set of bins obtained.

Step 4: For all points k in some bin B_i, define:

$$\hat{p}_i = \frac{1}{|B_i|} \sum_{i \in B_i} y_i$$

and

$$\hat{q}_i = \frac{1}{|B_i|} \sum_{i \in B_i} q_i$$

Step 5: The MSE calibration score is calculated as:

$$CalibMSE = \frac{1}{n} \sum_{i=1}^{T} |B_i| (\hat{q}_i - \hat{p}_i)^2$$

3.2.2 Confidence interval estimation

The 95% confidence interval for $\hat{p}_i$ is approximated by:

$$\hat{p}_i \pm 1.96\hat{se}_i$$

where $\hat{se}_i = \sqrt{\frac{\hat{p}_i(1-\hat{p}_i)}{|B_i|}}$ is the standard error at for the i^{th} bin.

It should be clear that the above formula is only an estimate of the confidence interval and thus does not guarantee true coverage. However, we found that this method is simple to implement and works well in practice.

In order to obtain the confidence interval for the MSE score, we use the following sampling procedure:

Data: A set of approximations $\{\hat{p}_1, \hat{p}_2, \cdots, \hat{p}_T\}$.

Parameter: Number of samples N_s.

Step 1: Sample N_s times. Each time, for i from 1 to T, draw $\hat{p}_i^* \sim \mathcal{N}\left(\hat{p}_i, \hat{se}_i^2\right)$.

Step 2: Report 95% confidence interval for the calibration score as:

$$CalibMSE_{avg} \pm 1.96\hat{se}_{MSE}$$

where $CalibMSE_{avg}$ and $\hat{se}_{MSE}$ are the mean and the standard error of the MSE calibration scores calculated from the samples.

3.3 Visualize calibration

To have a more general view of a model's degree of miscalibration, we can plot on the pairs $(\hat{p}_1, \hat{q}_1), (\hat{p}_2, \hat{q}_2), \cdots, (\hat{p}_T, \hat{q}_T)$ obtained from the adaptive binning procedure and visualize the *calibration curve* of the model (Figure BLAH BLAH). Calibration curve provides a fine-grained insight into the behavior of the model. To be perfectly calibrated, the curve has to coincide with the diagional line "y = x", or the *perfect calibration curve* (PCC). At places where the calibration curve lies above the PCC, the model predicts with less confidence then it should have done. Converesely, the model is overconfident where the calibration curve lies below the PCC.

An advantage of using the points obtained from the adaptive binning procedure in visualizing calibration is that the plot also captures the refinement aspect of the model. The distribution of points' x coordinates corresponds to the distribution of the model's predictions. On the other hand, if using equally spaced bins, one would need an extra plot to demonstrate that distribution.

3.4 Applications of calibration in NLP

Calibration analysis is model-indepedent and can easily be applied to analyze posterior predictions of models for structure prediction problems. In this setting, binary events are well-defined as polar queries on (sub)structures of the model such as single words, entity-spans or parsing substree. It is not necessary to test whether a model is calibrated for all types of queries. Depending on different downstream taks, we want to have good calibration on different types of queries. Take syntatic parsing as an example. If the downstream tasks is a sentiment analysis tasks, it is important for the model to be reliable in predicting if a phrase is an adjective phrase. For a different application such as coreference resolution, we would care more about noun phrase's boundaries.

3.4.1 Sequence models

Logistic regression has been shown to give better calibrated probabilities than Naive Bayes (CITE Caruana), which is a signal that discriminative models would be better than generative models in posterior predictions. We take one step towards proving this claim by examining a two popular classes of sequence models for POS tagging, Hidden Markov Models (HMMs) and Conditional Random Fields (CRFs). First of all, we conduct an experiment on a regular WSJ data set extracted from

CoNLL-2011 (CITE ConLL). Predicting POS tagging on the WSJ data set is almost solved problem (CITE somebody). Choosing such an easy task, we would like to analyze the behavior of well-calibrated models. Next, we test the two models on a more challenging task, predicting POS for tweets (CITE Noah). The difficulty of this task does not only come from the linguistic structure of tweets but also from the relatively small size of the data set.

The queries we choose to test calibration on are in the form: "Is the current word a(n) X?". In the first experiment, X is the "NN" tag. In the second experiment, we change to predicting the "V" tag since we find that predicting nouns in tweets is still easy for both models. Elegant dynamic programming algorithms are available for compute the marginal probability of a particular tag configuration of a substring in the sentence (CITE Finkel?). The forward-backward algorithm, used in training CRF, provides an efficient way to compute the marginal probability for substrings of length 1 or 2.

Also, since our focus is on model structure, we only use basic features for in our CRF implementation. Our feature set consists of only transition features (tag-tag) and emission features (tag-current word). By doing so, we disentangle the advantage of CRF's model assumptions from the advantage of it having more parameters than HMM. Later, we also conduct an ablation test on the CRF to verify the affect of features on calibration.

3.4.2 Coreference resolution

Coreference resolution is the task of BLAH BLAH. It remains a challenging task in NLP with state-of-the-art techniques are around 60-70% accuracy on the CoNLL data set. In fact, the performances of those techniques, even on the same data set, are often not comparable due to the lack of an evaluating procedure for coreference resolution. Selecting one techinque for a end-to-end system is a really a matter of tradeoff and depends on the nature of the downstream tasks. Calibration analysis is suitable for coreference resolution for two reasons. First, when the accuracy scores are all low, a few percentages difference between imperfect models in accuracy are not significant since the models will make unsatisfiable mistakes eventually. At that time, it is more important to choose a model that is capable of truely quantifying its own mistakes so that risks can be estimated and possibly mitigated. This is when calibration analysis comes to help as a tool to separate the reliable imperfect models from the non-reliable ones. Second, calibration analysis is flexible. The calibration query can be altered for different downstream goals as long as the tested model provides a mechanism for calculating the confidence level of the query.

We conduct a calibration analysis on the Berkeley coreference system (CITE Klein) on the CoNLL-2011 data set. The core of this system is a mention-ranking log-linear model that, for each mention, computes the confidence score of that mention referring to itself (singleton) and each of the previous mentions. Entity clusters are implicit during inference and only constructed after the relationships between the mentions have been determined. Unfortunately, due to this structure, the marginal probabilities of crucial queries such as “Does this pair of mentions belong to the same cluster?” is complicated to derive exactly. However, we can still obtain a fairly good estimate of the confidence scores via Monte Carlo sampling. Concretely, we sample directly the distribution ouput by the model and construct the set of clusters for each sample. The approximated confidence score for a query is the empirical proportion of the number of times the answer for the query is yes over a lot of samples.

CHAPTER 4

REPORT AND DISCUSSION OF RESEARCH RESULTS

4.1 Part-of-speech tagging

4.1.1 Data

We extract WSJ articles from the CoNLL-2011 dataset for this experiment. The original data set has already been splitted the into training, development and testing sets so we filter WSJ articles from each set and join the sentences into together. This process results in 11772 sentences for training, 1632 sentences for development and 1382 sentences for testing. The query tested is whether a word has the "NN" tag.

We train a HMM model using maximum likelihood principle. For CRF training, we implement the L2-regularized mini-batch AdaGrad method with a batch size of 100.

4.1.2 Results

Firstly, we compare HMM with a CRF model with basic features (CRF-Basic). As mentioned in section 3.4.1, CRF-Basic contains only the transition features and the emission features. Figure 4.1 shows calibration curves of the two models. CRF-Basic attains a significantly lower MSE calibration score than HMM does (0.019 vs. 0.035). Moreover, it also produces more refined predictions than those of HMM, as seen from the distributions of points along the x-axis.

To measure the affect of features on calibration, we conduct an ablation test for CRF. including surrounding words, word shape, word length, prefixes and suffixes. As we discover that as we use better template for CRF, we obtain more refined and

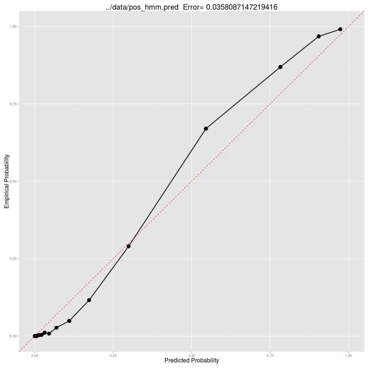

Figure 4.1. Calibration curve for HMM (POS), Acc = ??, CalibScore = ??

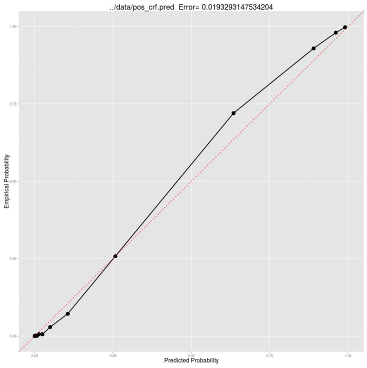

Figure 4.2. Calibration curve for CRF-Basic (POS), Acc = ??, CalibScore = ??

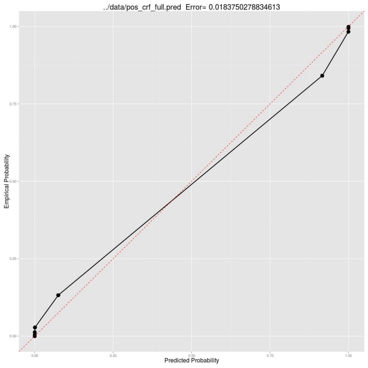

Figure 4.3. Calibration curve for CRF-Advanced (POS), Acc = ??, CalibScore = ??

calibrated predictions (FIGURE blah blah). A fully featurized CRF, which achieves a 96% accuraccy on the task, produces a calibration curve just slightly off the PCC.

4.2 Twitter part-of-speech tagging

4.2.1 Data

We repeat our comparision between HMMs and CRFs on a harder task, predicting POS tags for tweets. We use the ARK's Twitter POS data set (CITE NOAH), which consists of 1000 sentences for training, 327 sentences for development, 500 sentences for testing. The query tested is whether a word has the "V" tag.

We conduct the same experiments as in Section BLAH BLAH and obtain similar patterns. CRF-Basic's miscalibration is about half HMM's (Figure 4.4). On the other hand, equipped with better features, CRF-Advanced demonstrates a significant improvement from CRF-Basic, reducing further the miscalibration level by one half. It should also be noticed that CRF-Advanced does not give perfectly accurate predictions (87% accuracy) but those are reliable predictions.

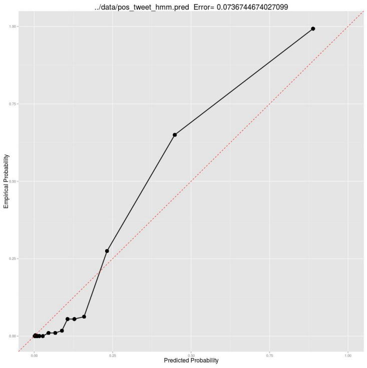

Figure 4.4. Calibration curve for HMM (POS Tweet), Acc = ??, CalibScore = ??

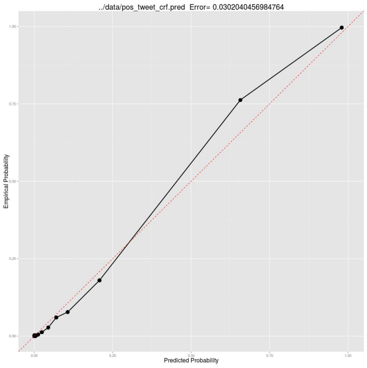

Figure 4.5. Calibration curve for CRF-Basic (POS Tweet), Acc = ??, CalibScore = ??

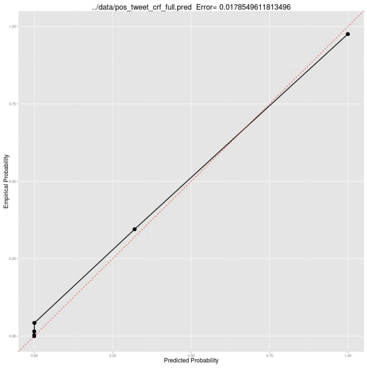

Figure 4.6. Calibration curve for CRF-Advanced (POS Tweet), Acc = ??, CalibScore = ??

4.3 Calibration analysis on synthesis data

4.3.1 Data

Approximating calibration statistics is very difficult since the distribution of the true labels is unknown. Therefore, we investigate the behavior of calibration statistics on synthesized set of prediction-observation pairs that mimics common NLP data distributions.

Each prediction-observation pair is generated as follows. First of all, the value of the prediction is drawn from a beta distribution. Then, the observation is obtained by sampling from a Bernoulli distribution whose parameter is a transformation of the value of the prediction. To obtain a perfectly calibrated set of pairs, we use the identity transformation. For uncalibrated condition, we use this function $t(p)$:

$$t(p) = \begin{cases} \max(0, p - k), & \text{if } 0 \leq x \leq 0.5 \\ \min(0, p + k), & \text{otherwise} \end{cases}$$

where k ∈ [0, 0.5].

4.3.2 Effect of bin size on calibration score

We investigate the the effect of varying the bin size on the value of the MSE calibration score. Theoretically, as we double the bin size, the score will not increase. This fact is obtain by using Jensen's inequality, leveraging the fact that the quaratic function is convex. In our experiment, we vary the bin size from 2^1 to 2^16 to calculate the MSE calibration score on a data set consists of 10^5 pairs. Our result (Figure BLAH BLAH) supports the theoretical hypothesis. The score monotonically decreases as the bin size exponentally increases. We also alter the parameters of our beta distribution and witness the same pattern. We attempt to generalize this pattern to a contious range of bin size values. Figure BLAH BLAH portrays the behavior of the score of a perfectly calibrated predictor as the bin size goes from 10^3 to 5.10^4 with a step size

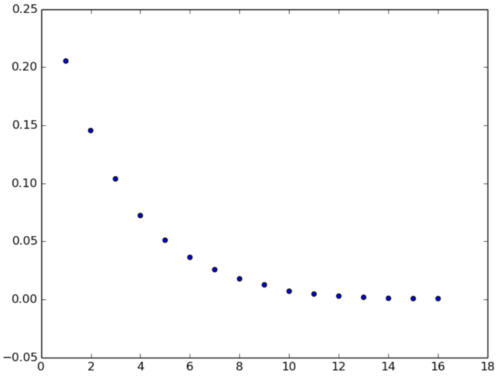

Figure 4.7. MSE calibration score versus bin size (Log scale) for calibrated predictions

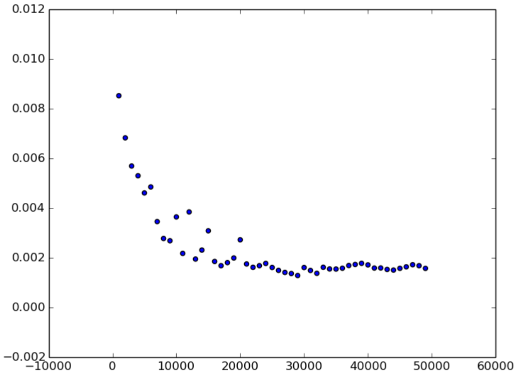

Figure 4.8. MSE calibration score versus bin size for calibrated predictions

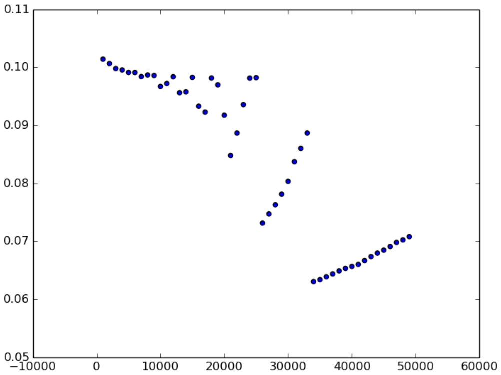

Figure 4.9. MSE calibration score versus bin size for uncalibrated predictions

of 10^3. As we can see, although the points do not always monotonically decrease, we see a similiar trend as the log-scale plot. However, for an uncalibrated predictor, the score is much more unpredictable (Figure BLAH BLAH).

4.3.3 Effect of sample size on calibration score

As pointed out by Foster (1998), we expect the calibration score of a perfectly calibrated predictor to go to zero as the sample size goes to infinity. We set up experiment to verify this fact. Using a range of sample size from 10^4 to 5.10^4, we compute

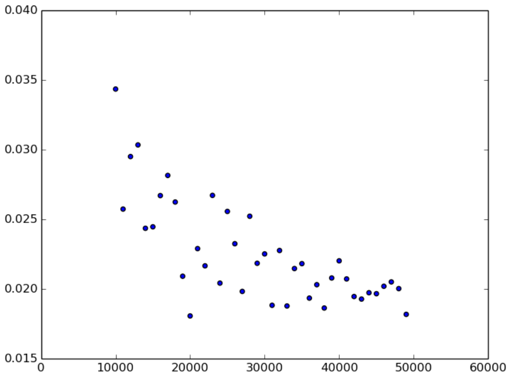

Figure 4.10. MSE calibration score versus sample size for calibrated predictions

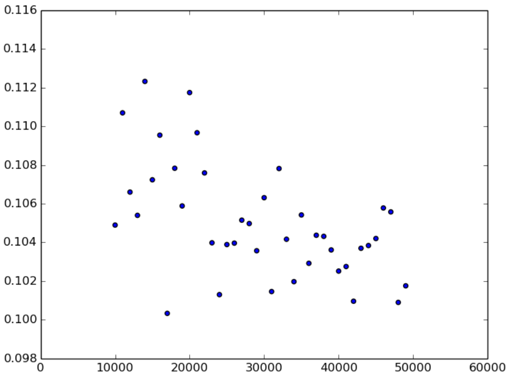

Figure 4.11. MSE calibration score versus sample size for uncalibrated predictions

the calibration score for three set of predictions: perfectly calibrated (PERFECT), uncalibrated using the function t(p) as true distribution with k = 0.1 (UNCALIB). For each experiement, we set the bin size to be the square root of the sample size. We observed distinguishing pattern between PERFECT and UNCALIB. The points in the CALIB's plot clearly approach zero while those of the UNCALIB's plot converge weakly.

BIBLIOGRAPHY

Charles Sutton and Andrew McCallum. Joint parsing and semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, pages 225–228. Association for Computational Linguistics, 2005.

Ben Wellner, Andrew McCallum, Fuchun Peng, and Michael Hay. An integrated, conditional model of information extraction and coreference with application to citation matching. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 593–601. AUAI Press, 2004.