Active Learning with c-Certainty

Eileen A. Ni and Charles X. Ling

Department of Computer Science The University of Western Ontario London, Ontario, Canada {ani,cling}@csd.uwo.ca

Abstract. It is well known that the noise in labels deteriorates the performance of active learning. To reduce the noise, works on multiple oracles have been proposed. However, there is still no way to guarantee the label quality. In addition, most previous works assume that the noise level of oracles is evenly distributed or example-independent which may not be realistic. In this paper, we propose a novel active learning paradigm in which oracles can return both labels and confidences. Under this paradigm, we then propose a new and effective active learning strategy that can guarantee the quality of labels by querying multiple oracles. Furthermore, we remove the assumptions of the previous works mentioned above, and design a novel algorithm that is able to select the best oracles to query. Our empirical study shows that the new algorithm is robust, and it performs well with given different types of oracles. As far as we know, this is the first work that proposes this new active learning paradigm and an active learning algorithm in which label quality is guaranteed.

Key words: Active learning; multiple oracles; noisy data;

1 Introduction

It is well known that the noise in labels deteriorates learning performance, especially for active learning, as most active learning strategies often select examples with noise on many natural learning problems [1]. To rule out the negative effects of the noisy labels, querying multiple oracles has been proposed in active learning [2–4]. This multiple-oracle strategy is reasonable and useful in improving label quality. For example, in paper reviewing, multiple reviewers (i.e., oracles or labelers) are requested to label a paper (as accepted, weak accepted, weak rejected or rejected), so that the final decision (i.e., label) can be more accurate.

However, there is still no way to guarantee the label quality in spite of the improvements obtained in previous works [3–5]. Furthermore, strong assumptions, such as even distribution of noise [3], and example-independent (fixed) noise level [4], have been made. These assumptions, in the paper reviewing example mentioned above, imply that all the reviewers are at the same level of expertise and have the same probability in making mistakes.

Obviously, the assumptions may be too strong and not realistic, as it is ubiquitous that label quality (or noise-level) is example-dependent in real-world data. In the paper reviewing example, the quality of a label given by a reviewer should depend heavily on how close the reviewer's research is to the topic of the paper. The closer it is, the higher quality the label has. Thus, it is necessary to study this learning problem further.

In this paper, we propose a novel active learning paradigm, under which oracles are assumed to return both labels and confidences. This assumption is reasonable in real-life applications. Taking paper reviewing as an example again, usually a reviewer is required to give not only a label (accept, weak accept, weak reject or reject) for a paper, but also his confidence (high, medium or low) for the labeling.

Under the paradigm, we propose a new active learning strategy, called c-certainty learning. C-certainty learning guarantees the label quality to be equal to or higher than a threshold c (c is the probability of correct labeling; see later) by querying oracles multiple times. In the paper reviewing example, with the labels and confidences given by reviewers (oracles), we can estimate the certainty of the label. If the certainty is too low (e.g., lower than a given c), another reviewer has to be sought to review the paper to improve the label quality.

Furthermore, instead of assuming noise level to be example-independent in the previous works, we allow it to be example-dependent. We design an algorithm that is able to select the *Best Multiple Oracles* to query (called BMO) for each given example. With BMO, fewer queries are required on average for a label to meet the threshold c compared to random selection of oracles. Thus, for a given query budget, BMO is expected to obtain more examples with labels of high quality due to the selection of best oracles. As a result, more accurate models can be built.

We conduct extensive experiments on the UCI datasets by generating various types of oracles. The results show that our new algorithm BMO is robust, and performs well with the different types of oracles. The reason is that BMO can guarantee the label quality by querying oracles repeatedly and ensure the best oracles can be queried. As far as we know, this is the first work that proposes this new active learning paradigm.

The rest of this paper is organized as follows. We review related works in Section 2. Section 3 introduces the learning paradigm and the calculation of certainty after querying multiple oracles. We present our learning algorithm, BMO, in Section 4 and the experiment results in Section 5. We conclude our work in Section 6.

2 Previous Works

Labeling each example with multiple oracles has been studied when labeling is not perfect in supervised learning [5–7]. Some principled probabilistic solutions have been proposed on how to learn and evaluate the multiple-oracle problem.

However, as far as we know, few of them can guarantee the labeling quality to be equal to or greater than a given threshold c, which can be guaranteed in our work.

Other recent works related to multiple oracles have some assumptions which may be too strong and unrealistic. One assumption is that the noise of oracles is equally distributed [3]. The other type of assumption is that the noise level of different oracles are different as long as they do not change over time [4, 8]. Their works estimate the noise level of different oracles during the learning process and prefer querying the oracles with low noise levels. However, it is ubiquitous that the quality of an oracle is example-dependent. In this paper, we remove all the assumptions and allow the noise level of oracles vary among different examples.

Active learning on the data with example-dependent noise level was studied in [9]. However, it focuses on how to choose examples considering the tradeoff between more informative examples and examples with lower noise level.

3 C-Certainty Labeling

C-Certainty labeling is based on the assumption that oracles can return both labels and their confidences in the labelings. For this study, we define *confidence* formally first here. *Confidence* for labeling an example x is the probability that the label given by an oracle is the same as the true label of x. We assume that the confidences of oracles on any example are greater than 0.5^1 .

By using the labels and confidences given by oracles, we guarantee that the label certainty of each example can meet the threshold c ($c \in (0.5, 1]$) by querying oracles repeatedly (called *c-certainty labeling*). That is, a label is valid if its certainty is or equal to than c. Otherwise, more queries would be issued to different oracles to improve the certainty.

How to update the label certainty of an example x after obtaining a new answer from an oracle? Let the set of previous n-1 answers be A^{n-1} , and the new answer be A_n in the form of (P, f_n) , where P indicates positive and f_n is the confidence. The label certainty of x, $C(T_P|A^n)$, can be updated with Formula 1 (See Appendix for the details of its derivation).

$$C(T_P|A^n) = \begin{cases} \frac{p(T_P) \times f_n}{p(T_P) \times f_n + p(T_N) \times (1 - f_n)}, & \text{if } n = 1 \text{ and } A_n = \{P, f_n\} \\ \frac{C(T_P|A^{n-1}) \times f_n}{C(T_P|A^{n-1}) \times f_n + (1 - C(T_P|A^{n-1})) \times (1 - f_n)}, & \text{if } n > 1 \text{ and } A_n = \{P, f_n\}, \end{cases}$$

$$(1)$$

where T_P and T_N are the true positive and negative labels respectively. Formula 1 can be applied directly when A_n is positive (i.e., $A_n = \{P, f_n\}$); while for a negative answer, we can transform it as $A_n = \{N, f_n\} = \{P, (1 - f_n)\}$ such that

¹ This assumption is reasonable, as usually oracles can label examples more correctly than random.

Active Learning with c-Certainty

4

Formula 1 is also applicable. In addition, Formula 1 is for calculating the certainty of x to be positive. If $C(T_P|A^n) > 0.5$, the label of x is positive; otherwise, the label is negative and the certainty is $1 - C(T_P|A^n)$. With Formula 1, the process of querying oracles can be repeated for x until $max(C(T_P|A^n), 1 - C(T_P|A^n))$ is greater than or equal to c.

However, from Formula 1 we can see that the certainty, $C(T_P|A^n)$, is not monotonic. It is possible that the certainty dangles around and is always lower than c. For example, in paper reviewing, if the labels given by reviewers are with low confidence or alternating between positive and negative, the certainty may not be able to reach the threshold c even many reviewers are requested.

To guarantee that the threshold c is reachable, we will propose an effective algorithm to improve the efficiency of selecting oracles.

4 BMO (Best-Multiple-Oracle) with C-Certainty

To improve the querying efficiency, the key issue is to select the best oracle for *every* given example. This is very different from the case when the noise level is example-independent [4,8], as in our case the performance of each oracle varies on labeling different examples.

4.1 Selecting the Best Oracle

How to select the best oracle given that the noise levels are example-dependent? The basic idea is that an oracle can probably label an example x with high confidence if it has labeled x_j confidently and x_j is close to x. This idea is reasonable as the confidence distribution (expertise level) of oracles is usually continuous, and does not change abruptly. More specifically, we assume that each of the m oracle candidates (O_1, \dots, O_m) has labeled a set of examples \mathcal{E}_i $(1 \le i \le m)$. E_{k_i} $(1 \le i \le m)$ is the set of k (k = 3 in our experiment) nearest neighbors of x in \mathcal{E}_i $(1 \le i \le m)$. BMO chooses the oracle O_i such that examples in E_{k_i} are of high confidence and close to the example x. The potential confidence for each oracle in labeling x can be calculated with Formula 2.

$$P_{c_i} = \frac{\frac{1}{k} \times \sum_{j=1}^{k} f_{x_j}^{o_i}}{1 + \frac{1}{k} \times \sum_{j=1}^{k} |x - x_j|},$$
 (2)

where $x_j \in E_{k_i}$, $f_{x_j}^{o_i}$ is the confidence of oracle O_i in labeling x_j , and $|x-x_j|$ is the Euclidean distance between x_j and x. The numerator of Formula 2 is the average confidence of the k nearest neighbors of x. The last item in the denominator is the average distance, and the 1 is added to prevent the denominator from being zero. High confidence and short distance indicate that the oracle O_i will more likely label x with a higher confidence. Thus, BMO selects the oracle O_i if $i = \arg\max_i (P_{c_1}, \dots, P_{c_i}, \dots, P_{c_m})$.

4.2 Active Learning Process of BMO

BMO is a wrapper learning algorithm, and it treats the strategy of selecting examples to label as a black box. Any existing query strategies in active learning, such as uncertainty sampling [10], expected error reduction [11] and the density-weighted [12] method can be fit in easily.

We assume that BMO starts with an empty training set, and the learning process is as follows. For an example x_i ($x_i \in \mathcal{E}_u$) selected by an example-selecting strategy (e.g.,uncertain sampling), BMO selects the best oracle among the ones that have not been queried for x_i yet to query, and updates the label certainty of x_i with Formula 1. This process repeats until the certainty meets the threshold c. Then BMO adds x_i into its labeled example set \mathcal{E}_l . This example-labeling process continues until certain stop criterion is met (such as the predefined query budget is used up in our experiment). (See Algorithm 1 for details.)

Algorithm 1: BMO (Best Multiple Oracles)

```
Input: Unlabeled data: \mathcal{E}_u; oracles: \mathcal{O}; oracles queried: \mathcal{O}_q; threshold: c;
                 queries budge: budget;
     Output: labeled example set \mathcal{E}_l
    begin
 1
          while budget > 0 do
 3
                x_i \leftarrow \text{selection with uncertain sampling } (x_i \in \mathcal{E}_u);
                \mathcal{O}_q \leftarrow null; \quad certainty \leftarrow 0;
 4
 5
                while certainty < c do
                     for each O_i \in (\mathcal{O} - \mathcal{O}_q) do
 6
                           Po\_c_i \leftarrow Formula 2;
 7
                     end
 8
                     O_m \leftarrow \text{oracle with maximal } P_c;
 9
                     certainty \leftarrow update with Formula 1;
10
                     \mathcal{O}_q \leftarrow \mathcal{O}_q \cup O_m; budget \leftarrow budget - 1;
11
12
                \mathcal{E}_l \leftarrow \mathcal{E}_l \cup anditscertainty;
13
               update the current model;
14
          end
15
          return \mathcal{E}_l;
16
17 end
```

By selecting the best oracles, BMO can improve the label certainty of a given example to meet the threshold c with only a few queries (See Section 5). That is, more labeled examples can be obtained for a predefined query budget compared to random selection of oracles. Thus, the model built is expected to have better performance.

5 Experiments

In our experiment, to compare with BMO, we implement two other learning strategies. One is $Random\ selection\ of\ Multiple\ Oracles\ (RMO)$. Rather than selecting the best oracle in BMO, RMO selects oracles randomly to query for a given example and repeats until the label certainty is greater than or equal to c. The other strategy is $Random\ selection\ of\ Single\ Oracle\ (RSO)$. RSO queries for each example only once without considering c, which is similar to traditional active learning algorithms.

Since RSO only queries one oracle for each example, it will have the most labeled examples for a predefined query budget but with the highest noise level. To reduce the negative effect of noisy labels, we *weight* all labeled examples according to their label certainty when building final models. To make all the three strategies comparable, we also use weighting in BMO and RMO. In addition, all the three algorithms take uncertain sampling as the example-selecting strategy and decision tree (J48 in WEKA [13]) as their base learners. The implementation is based on the WEKA source code.

The experiment is conducted on UCI datasets [14], including abolone, anneal, cmc_new, credit, mushroom, spambase and splice, which are commonly used in the supervised learning research. As the number of oracles cannot be infinite in real world, we only generate 10 oracles for each dataset. If an example has been presented to all the 10 oracles, the label and the certainty obtained will be taken in directly. The threshold c is predefined to be 0.8 and 0.9 respectively. The experiment results presented are the average of 10 runs, and t-test results are of 95% confidence.

In our previous discussion, we take the confidence given by an oracle as the true confidence. However, in real life, oracles may overestimate or underestimate themselves intentionally or unintentionally. If the confidence given by an oracle O does not equal the true confidence, we call O an unfaithful oracle; otherwise, it is faithful. To observe the robustness of our algorithm, we conduct our empirical studies with both faithful and unfaithful oracles² in the following.

5.1 Results on Faithful Oracles

As no oracle is provided for the UCI data, we generate a faithful oracle as follows. Firstly, we select one example x randomly as an "expertise center" and label it with the highest confidence. Then, to make the oracle faithful, we calculate the Euclidean distance from each of the rest examples to x, and assign them confidences based on the distances. The further the distance is, the lower confidence the oracle has in labeling the example. Noise is added into labels according to the confidence level. Thus the oracle is faithful.

² Actually it is difficult to model the behaviors of unfaithful oracles with a large confidence deviation. In our experiment, we show that our algorithm works well given unfaithful oracles slightly deviating from the true confidence.

The confidence is supposed to follow a certain distribution. We choose three common distributions, linear, normal and dual normal distributions. Linear distribution assumes the confidence reduces linearly as the distance increases. For normal distribution, the reduction of confidence follows the probability density function $f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} exp(-\frac{x^2}{2\sigma^2}) - 0.55$. Dual normal distribution indicates that the oracle has two "expertise centers" (see Figure 1). As mentioned earlier, we generate 10 oracles for each dataset in this experiment. Among the 10 oracles, three of them follow the linear distribution, three the normal distribution and four the dual normal distribution.

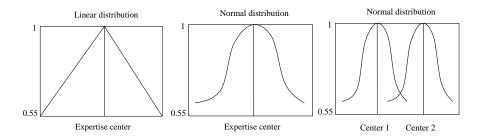


Fig. 1. Three distributions

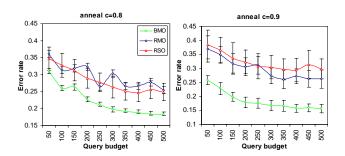


Fig. 2. Error rate on faithful oracles

Due to the similar results of different datasets, we only show the details of one dataset (anneal) in Figure 2 and a summary of the comparison afterwards. Figure 2 shows the testing error rates of BMO, RMO and RSO for the threshold 0.8 (left) and 0.9 (right) respectively. The x axis indicates the query budgets while the y axis represents the error rate on test data. On one hand, as we expected that, for both thresholds 0.8 and 0.9, the error rate of BMO is much lower than that of RMO and RSO for all different budgets, and the performances of the

latter two are similar. On the other hand, the curve of RMO when c=0.8 is not as smooth as the other ones.

The different performances of the three learning strategies can be explained by two factors, the noise level and the number of examples. Due to limited space, we only show how the two factors affect the performances through one dataset (anneal) when the query budget is 500 in Figure 3.

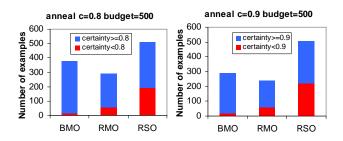


Fig. 3. The number of examples and label quality

Figure 3 shows that on average BMO only queries about 1.4 (c=0.8) and 1.7 (c=0.9) oracles for each example; while RMO queries more oracles (1.7 and 2.0). That is, BMO obtains more labeled examples than RMO for a given budget. Moreover, the examples labeled by BMO have much higher label certainty than that by RMO³. On the other hand, the examples labeled by RSO is much more noisy than BMO (i.e., the red portion is much larger). It is the noise that deteriorates the performance of RSO. Thus, BMO outperforms the other two strategies because of its guaranteed label quality and the selection of the best oracles to query.

By looking closely into the curves in Figure 2, we find that the curve of RMO when c=0.8 is not as smooth as the other ones. The reason is that RMO of c=0.8 has fewer labeled examples when compared to BMO and RSO of c=0.8 and has more noise when compared to that of c=0.9. Fewer examples make the model learnt more sensitive to the quality of each label; while the label quality of c=0.8 is not high enough. Thus, the stability of RMO when c=0.8 is weakened.

In addition, we also show the t-test results in terms of the error rate on all the seven UCI datasets in Table 1. As for each dataset 10 different query budgets are considered, the total times of t-test for each group is 70. Table 1 shows that BMO wins RMO 94 times out of 140 (c = 0.8 and c = 0.9) and wins RSO 86 out of 140 without losing once. It is clear that BMO outperforms RMO and RSO significantly.

In summary, with faithful oracles, the experiment results show that BMO does work better by guaranteeing the label quality and selecting the best oracles

 $^{^3}$ Some of the examples still have certainty lower than c due to the limited oracles in our experiment.

	BMO	vs. RMO	BMO	vs. RSO	RMO	vs. RSO
	c = 0.8	c=0.9	c = 0.8	c=0.9	c = 0.8	c=0.9
Win	43	51	40	42	0	9
Draw	27	19	30	38	70	51
Lose	0	0	0	0	0	10

Table 1. T-test results on 7 datasets with 10 different budgets

to query. On the other hand, even though RMO also can guarantee the label quality, its strategy of randomly selecting oracles reduces the learning performance. Furthermore, the results of RSO illustrate that weighting with the label quality may reduce the negative influence of noise but still its effect is limited.

5.2 Results on Unfaithful Oracles

Unfaithful oracles are generated for each dataset by building models over 20% of the examples. More specifically, to generate an oracle, we randomly select one example x as an "expertise center", and sample examples around it. The closer an example x_i is to x, the higher the probability it will be sampled with. Thus, the oracle built on the sampled examples can label the examples closer to x with higher confidences. The sampling probability follows exactly the same distribution in Figure 1. For each data set, 10 oracles are generated and three follow the linear distribution, three the normal distribution and four the dual normal distribution.

As sampling rate declines with the increasing distance, the oracle built may fail to give true confidence for the examples that are far from the "center". As a result, the oracle is unfaithful. That is, the oracles are unfaithful due to "insufficient knowledge" rather than "lying" deliberately.

We run BMO, RMO and RSO on the seven UCI datasets and show the testing error rates and the number of labeled examples on one data set (anneal) in Figure 4 and a summary on all the datasets afterwards. It is surprising that the performances of BMO on unfaithful oracles are similar to that on faithful oracles. That is, the error rate of BMO is much less than that of RMO and RSO, and the latter two are similar. The examples labeled by BMO are more than that by RMO and its label quality is higher than that of both RMO and SMO, which are also similar to that on faithful oracles.

The comparison shows clearly that BMO is robust even for unfaithful oracles. The reason is that BMO selects the best multiple oracles to query, and it is unlikely that all the best oracles are unfaithful at the same time as our unfaithful oracles do not "lie" deliberately as mentioned. Thus, BMO still performs well.

Table 2 shows the t-test results on 10 different query budgets for all the seven UCI datasets. We can see that BMO wins RMO 95 times out of 140 and wins RSO 98 out of 140, which indicates that BMO works significantly better than RMO and RSO under most of the circumstances. However, BMO loses to RMO 19 times and RSO 10 times, which are different from the results on

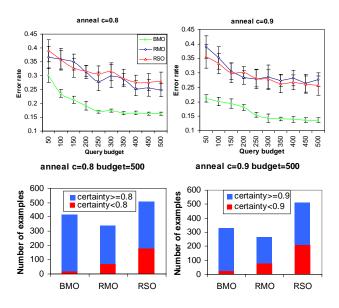


Fig. 4. Experiment results on unfaithful oracles

faithful oracles. Thus, even though BMO is robust, still it works slightly worse on unfaithful oracles than on faithful ones.

Table 2. T Test results for all datasets and budgets on unfaithful oracles.

	BMO	vs. RMO	BMO	vs. RSO	RMO	vs. RSO
	c = 0.8	c=0.9	c = 0.8	c=0.9	c = 0.8	c=0.9
Win	53	42	53	45	12	0
Draw	6	22	17	15	50	50
Lose	11	8	0	10	8	20

In summary, BMO is robust for working with unfaithful oracles, even though its good performance may be reduced slightly. This property is crucial for BMO to be applied successfully in real applications.

6 Conclusion

In this paper, we proposed a novel active learning paradigm, c-certainty learning, in which oracles can return both labels and confidence. Under this new paradigm, the label quality is guaranteed to be greater than or equal to a given threshold c by querying multiple oracles. Furthermore, we designed the learning algorithm BMO to select the best oracles to query so that the threshold c can be met

with fewer queries compared to selecting oracles randomly. Empirical studies are conducted for both faithful and unfaithful oracles. The results show that BMO works robustly and outperforms other active learning strategies significantly on both faithful and unfaithful oracles, even though its performance can be affected slightly by unfaithful oracles.

References

- M. Balcan, A. Beygelzimer, and J. Langford, "Agnostic active learning," in Proceedings of the 23rd international conference on Machine learning. ACM, 2006, pp. 65–72.
- 2. B. Settles, "Active Learning Literature Survey," *Machine Learning*, vol. 15, no. 2, pp. 201–221, 1994.
- 3. V. Sheng, F. Provost, and P. Ipeirotis, "Get another label? improving data quality and data mining using multiple, noisy labelers," in *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2008, pp. 614–622.
- P. Donmez, J. Carbonell, and J. Schneider, "Efficiently learning the accuracy of labeling sources for selective sampling," in *Proceedings of the 15th ACM SIGKDD* international conference on Knowledge discovery and data mining. ACM, 2009, pp. 259–268.
- V. Raykar, S. Yu, L. Zhao, A. Jerebko, C. Florin, G. Valadez, L. Bogoni, and L. Moy, "Supervised Learning from Multiple Experts: Whom to trust when everyone lies a bit," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 889–896.
- 6. R. Snow, B. O'Connor, D. Jurafsky, and A. Ng, "Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2008, pp. 254–263.
- 7. A. Sorokin and D. Forsyth, "Utility data annotation with amazon mechanical turk," in Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on. IEEE, 2008, pp. 1–8.
- 8. Y. Zheng, S. Scott, and K. Deng, "Active learning from multiple noisy labelers with varied costs," in 2010 IEEE International Conference on Data Mining. IEEE, 2010, pp. 639–648.
- 9. J. Du and C. Ling, "Active learning with human-like noisy oracle," in 2010 IEEE International Conference on Data Mining. IEEE, 2010, pp. 797–802.
- D. Lewis and W. Gale, "A sequential algorithm for training text classifiers," in Proceedings of the 17th annual international ACM SIGIR conference. Springer-Verlag New York, Inc., 1994, pp. 3–12.
- 11. N. Roy and A. McCallum, "Toward optimal active learning through sampling estimation of error reduction," in *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE*-. Citeseer, 2001, pp. 441–448.
- 12. B. Settles, M. Craven, and S. Ray, "Multiple-instance active learning," in *In Advances in Neural Information Processing Systems (NIPS.* Citeseer, 2008.
- 13. WEKA Machine Learning Project, "Weka," URL http://www.cs.waikato.ac.nz/~ml/weka.
- 14. A. Asuncion and D. Newman, "UCI machine learning repository," URL http://www.ics.uci.edu/ mlearn/mlrepository.html, 2007.

Appendix: Derivation of Formula 1

$$C(T_P|A^n) = \frac{P(A^n|T_P) \times P(T_P)}{P(A^n)}$$

$$= \frac{P(A^{n-1}, A_n|T_P) \times P(T_P)}{P(A^n)}$$

$$= \frac{P(A^{n-1}|T_P) \times P(T_P) \times P(A_n|T_P) \times P(A^{n-1})}{P(A^{n-1}) \times P(A^n)}$$

$$= C(T_P|A^{n-1}) \times p(A_n|T_P) \times \frac{p(A^{n-1})}{p(A^n)}$$
(3)

The last item in Equation 3 can be further transformed as follows.

$$\begin{split} &\frac{p(A^{n-1})}{p(A^n)} \\ &= \frac{p(A^{n-1})}{p(A^n|T_P) \times p(T_P) + p(A^n|T_N) \times p(T_N)} \\ &= \frac{p(A^{n-1})}{p(A^{n-1}|T_P) \times p(T_P) \times p(A_n|T_P) + p(A^{n-1}|T_N) \times p(T_N) \times p(A_n|T_N)} \\ &= \frac{1}{(C(T_P|A^{n-1})) \times p(A_n|T_P) + C(T_N|A^{n-1}) \times p(A_n|T_N))} \\ &\text{As } A_n = (P, f_n), \\ &C(T_P|A^n) \\ &= \frac{C(T_P|A^{n-1}) \times p(A_n|T_P)}{C(T_P|A^{n-1}) \times p(A_n|T_P) + (1 - C(T_P|A^{n-1})) \times p(A_n|T_N)} \\ &= \frac{C(T_P|A^{n-1}) \times f_n}{C(T_P|A^{n-1}) \times f_n + (1 - C(T_P|A^{n-1})) \times (1 - f_n)} \end{split}$$