An Average-Case Analysis of the k-Nearest Neighbor Classifier for Noisy Domains

Seishi Okamoto
Fujitsu Laboratories Ltd.
2-2-1 Momochihama, Sawara-ku
Fukuoka 814, Japan
seishi@flab.fujitsu.co.jp

Yugami Nobuhiro
Fujitsu Laboratories Ltd.
2-2-1 Momochihama, Sawara-ku
Fukuoka 814, Japan
yugami@flab.fujitsu.co.jp

Abstract

This paper presents an average-case analysis of the fc-nearest neighbor classifier (k-NN). Our analysis deals with m-of-n// concepts, and handles three types of noise: relevant attribute noise, irrelevant attribute noise, and class noise. We formally compute the expected classification accuracy of fc-NN after a certain fixed number of training instances. This accuracy is represented as a function of the domain characteristics. Then, the predicted behavior of fc-NN for each type of noise is explored by using the accuracy function. We examine the classification accuracy of fc-NN at various noise levels, and show how noise affects the accuracy of fc-NN. We also show the relationship between the optimal value of *k* and the number of training instances in noisy domains. Our analysis is supported with Monte Carlo simulations.

1 Introduction

The fc-nearest neighbor classifier (fc-NN) is one of the most widely applied learning algorithms. Although fc-NN is a powerful algorithm and has been studied by many researchers, it is *not* clear how noise affects the classification accuracy of fc-NN. Moreover, it is also unclear which value should be chosen for *k* to maximize accuracy in noisy domains. These are crucial problems in fc-NN applications, because there are few noise-free problems in practical domains.

Variants of fc-NN have been proposed to tolerate noise *(e.g.,* [Aha and Kibler, 1989]), and to choose an appropriate value of *k (e.g.,* [Creecy *et al,* 1992]). These proposals exhibit the high performance of fc-NN by empirical evaluations. However, the noise effects on the accuracy of fc-NN and on the optimal value of *k* still remain unclear. It is therefore important to understand the noise effects on fc-NN and the optimal *k* by theoretical evaluations.

There have been several theoretical analyses of fc-NN. The upper bound of fc-NN error rate (risk) is twice the optimal Bayes risk under the assumption of an infinite number of training instances [Cover and Hart, 1967]. Moreover, fc-NN risk converges to the optimal Bayes risk as fc approaches infinity [Cover, 1968]. For a finite set of training instances, the new bounds of 1-NN risk are given using Bayes risk [Drakopoulos, 1995]. Aha *et al.* [1991] analyze 1-NN with a similar model to PAC (Probably Approximately Correct) learning, and this analysis is generalized to fc-NN [Albert and Aha, 1991]. Although these theoretical results are important and give some insights into the behavior of fc-NN, all of these studies assume noise-free instances.

An average-case analysis is a useful theoretical framework to understand the behavior of learning algorithms [Pazzani and Sarrett, 1992]. This framework is based on the formal computation of the expected accuracy of a learning algorithm for a certain fixed class of concepts. Using the result of this computation, we can explore the predicted behavior of an algorithm. There have been some average-case analyses of fc-NN. Langley and Iba [1993] analyzed 1-NN for conjunctive concepts, and we analyzed fc-NN for m-of-n concepts without irrelevant attribute [Okamoto and Satoh, 1995]. However, these studies assumed noise-free instances. Recently, we presented an average-case analysis of 1-NN for m-of-n concepts with irrelevant attribute in noisy domains [Okamoto and Yugami, 1996]. This paper generalizes our recent study for 1-NN to fc-NN.

In this paper, we present an average-case analysis of fc-nearest neighbor classifier for noisy domains. Our analysis handles m-of-n concepts with I irrelevant attributes, and deals with three types of noise: relevant attribute noise, irrelevant attribute noise, and class noise. First, we formally compute the expected classification accuracy *(i.e.,* predictive accuracy) of fc-NN after *N* training instances are given. This accuracy is represented as a function of the domain characteristics: fc, *N,* m, n, /, the probabilities of occurrence for relevant and irrelevant attributes, and noise rates. Using the accuracy function, we explore the predicted behavior of fc-NN in noisy domains. We describe the predictive accuracy of fc-NN at various noise levels, and show the effects of noise on the accuracy of fc-NN. We also show the relationship between the optimal value of fc and the number of training instances in noisy domains. Our theoretical analysis is supported with Monte Carlo simulations.

2 Problem Description

Our analysis deals with m-of-n// concepts defined over the threshold m, n relevant and l irrelevant Boolean attributes [Murphy and Pazzani, 1991]. These concepts classify an instance as positive if and only if at least m out of n relevant attributes occur (i.e., take the value 1) in its instance.

Our analysis handles three types of noise. Each type of noise is independently introduced by the following common definition. Relevant (irrelevant, *resp.)* attribute noise flips an arbitrary relevant (irrelevant, *resp.)* attribute value in each instance with a certain probability σ_r (σ_i, *resp.*). Class noise replaces the class label for each instance with its negation with a certain probability σ_c.

We investigate a k-nearest neighbor classifier using hamming distance (i.e., the number of attributes on which two instances differ) as a distance measure. For the distribution over the instance space, our analysis assumes every relevant and irrelevant attribute independently occurs with a certain probability p and q. Each training instance is independently drawn from the instance space. After the effects of each type of noise, all training instances are stored into memory to allow for duplication. When a test instance is given, A;-NN classifies the test instance into a majority class (positive or negative) among its k nearest training instances. If the number of positive instances equals that of negative instances among its k nearest neighbors, then k-NN randomly determines the class of the test instance (this situation can occur only when k is an even number).

3 Predictive Accuracy

We formally compute the predictive accuracy of k-NN for m-of-n// target concepts after TV training instances are given. The predictive accuracy is represented as a function of the domain characteristics: k, N, m, n, /, p, q, σ_r, σ_i, and σ_c. However, to avoid complicated notation, we will not explicitly express these characteristics as parameters of the accuracy function with the exception of k.

We compute the predictive accuracy in the case where each type of noise affects only training instances. After this computation, we also give the accuracy function in the case where noise affects both test and training instances.

To compute the predictive accuracy, we use a set of instances in which x relevant attributes and y irrelevant attributes simultaneously occur (we denote this set with $I(x,y)$). Let $P_{occ}(x,y)$ be the probability that an arbitrary noise-free instance belongs to I(x,y). This probability is given by

$$P_{occ}(x,y) = \binom{n}{x}\binom{l}{y} p^x (1-p)^{n-x} q^y (1-q)^{l-y}.$$

Under our assumptions given in Section 2, k-NN has the same expected probability of correct classification for an arbitrary test instance in $I(x,y)$. Hence, we can represent the predictive accuracy of k-NN after N training instances as

$$A(k) = \sum_{y=0}^{l} \left\{ \sum_{x=0}^{m-1} P_{occ}(x,y)\,(1 - P_{pos}(k,x,y)) + \sum_{x=m}^{n} P_{occ}(x,y) P_{pos}(k,x,y) \right\},$$

where $P_{pos}(k,x,y)$ represents the probability that k-NN classifies an arbitrary test instance in $I(x,y)$ as positive.

Let $t(x,y)$ be an arbitrary test instance in $I(x,y)$. To represent $P_{pos}(x,y)$, we compute the appearance probability for an arbitrary training instance with distance $e(0 \le e \le n+l)$ from $t(x,y)$. Let $P_{dp}(x,y,e)$ ($P_{dn}(x,y,e)$, *resp.*) be this probability for an arbitrary training instance with the positive (negative, *resp.*) class label. $P_{dp}(x,y,e)$ and $P_{dn}(x,y,e)$ were computed using Eq.(14) and Eq.(15) in our previous paper [Okamoto and Yugami, 1996]. Hence, we simply state the computation of these probabilities here.

$P_{dp}(x,y,e)$ and $P_{dn}(x,y,e)$ are given by

$$P_{dp}(x,y,e) = \sum_{X=0}^{n} \sum_{Y=0}^{l} P_p(X,Y) P_{dis}(x,y,X,Y,e),$$
$$P_{dn}(x,y,e) = \sum_{X=0}^{n} \sum_{Y=0}^{l} P_n(X,Y) P_{dis}(x,y,X,Y,e).$$

In these equations, $P_p(X,Y)$ ($P_n(X,Y)$, *resp.*) represents the probability that an arbitrary training instance belongs to $I(X,Y)$ and has the positive (negative, *resp.*) class label. Moreover, $P_{dis}(x,y,X,Y,e)$ denotes the probability that an arbitrary training instance in $I(X,Y)$ has distance e from $t(x,y)$.

First, we represent $P_p(X,Y)$ and $P_n(X,Y)$ by considering the effects of each type of noise on the training instances. These probabilities are represented as

$$P_p(X,Y) = (1-\sigma_c) P_{p_0}(X,Y) + \sigma_c P_{n_0}(X,Y),$$
$$P_n(X,Y) = \sigma_c P_{p_0}(X,Y) + (1-\sigma_c) P_{n_0}(X,Y),$$

where $P_{p_0}(X,Y)$ ($P_{n_0}(X,Y)$, *resp.*) denotes the appearance probability for an arbitrary positive (negative, *resp.*) training instance in $I(X,Y)$, before the effect of class noise. $P_{p_0}(X,Y)$ and $P_{n_0}(X,Y)$ are given by

$$P_{p_0}(X,Y) = \sum_{X_0=m}^{n} \sum_{Y_0=0}^{l} P_{occ}(X_0,Y_0) P_{nr}(X_0,X) P_{ni}(Y_0,Y),$$
$$P_{n_0}(X,Y) = \sum_{X_0=0}^{m-1} \sum_{Y_0=0}^{l} P_{occ}(X_0,Y_0) P_{nr}(X_0,X) P_{ni}(Y_0,Y),$$

where $P_{nr}(X_0,X)$ ($P_{ni}(Y_0,Y)$, *resp.*) represents the probability that the number of relevant (irrelevant, *resp.*) attributes occurring in an arbitrary training instance is changed from X_0 (Y_0, *resp.*) to X (Y, *resp.*) by the effect of relevant (irrelevant, *resp.*) attribute noise. These probabilities are represented as

$$P_{\rm nr}(X_0,X) = \sum_{s=\max(0,X_0-X)}^{\min(X_0,n-X)} \left\{ \binom{X_0}{s}\binom{n-X_0}{X-X_0+s} \times \sigma_{\rm r}^{X-X_0+2s}(1-\sigma_{\rm r})^{n-(X-X_0+2s)} \right\},$$

$$P_{\rm ni}(Y_0,Y) = \sum_{t=\max(0,Y_0-Y)}^{\min(Y_0,l-Y)} \left\{ \binom{Y_0}{t}\binom{l-Y_0}{Y-Y_0+t} \times \sigma_{\rm i}^{Y-Y_0+2t}(1-\sigma_{\rm i})^{l-(Y-Y_0+2t)} \right\}.$$

Next, we represent $P_{\rm dis}(x,y,X,Y,e)$. Let $z_{\rm r}$ ($z_{\rm i}$, *resp.*) be the number of relevant (irrelevant, *resp.*) attributes which occur in both $t(x,y)$ and an arbitrary training instance in $I(X,Y)$. Then, $P_{\rm dis}(x,y,X,Y,e)$ is given by

$$P_{\rm dis}(x,y,X,Y,e) = \sum_{(z_{\rm r},z_{\rm i})\in S} \frac{\binom{x}{z_{\rm r}}\binom{n-x}{X-z_{\rm r}}}{\binom{n}{X}} \frac{\binom{y}{z_{\rm i}}\binom{l-y}{Y-z_{\rm i}}}{\binom{l}{Y}},$$

where S is a set of a pair of $z_{\rm r}$ and $z_{\rm i}$, that satisfies all conditions of

$$\max(0,x+X-n) \le z_{\rm r} \le \min(x,X),$$
$$\max(0,y+Y-l) \le z_{\rm i} \le \min(y,Y),$$
$$z_{\rm r}+z_{\rm i} = \frac{x+y+X+Y-e}{2}.$$

We have represented $P_{\rm dp}(x,y,e)$ and $P_{\rm dn}(x,y,e)$. Using these probabilities, we compute $P_{\rm pos}(k,x,y)$ in the accuracy function. For this computation, we consider the distance from $t(x,y)$ to the k-th nearest neighbor. We denote this distance with $d(0 \le d \le n+l)$. When the k-th nearest neighbor has distance d from $t(x,y)$, then exactly $a(0 \le a \le k-1)$ out of N training instances have the distance less than d from $t(x,y)$, and exactly b training instances have distance d. Here, we have $(k-a) \le b \le (N-a)$ from $k \le (a+b) \le N$. We use $P_{\rm num}(x,y,d,a,b)$ to denote the probability that this situation occurs. We also use $P_{\rm sp}(k,x,y,d,a,b)$ to designate the probability that k-NN classifies $t(x,y)$ as positive in this situation. Combining $P_{\rm num}(x,y,d,a,b)$ and $P_{\rm sp}(k,x,y,d,a,b)$, $P_{\rm pos}(k,x,y)$ can be computed as

$$P_{\rm pos}(k,x,y) = \sum_{d=0}^{n+l}\sum_{a=0}^{k-1}\sum_{b=k-a}^{N-a} P_{\rm num}(x,y,d,a,b)P_{\rm sp}(x,y,d,a,b).$$

We can represent $P_{\rm num}(x,y,d,a,b)$ as

$$P_{\rm num}(x,y,d,a,b) = \binom{N}{a}\binom{N-a}{b}P_{\rm l}(x,y,d)^a \times P_{\rm d}(x,y,d)^b(1-P_{\rm l}(x,y,d)-P_{\rm d}(x,y,d))^{N-a-b},$$

where $P_{\rm l}(x,y,d)$ and $P_{\rm d}(x,y,d)$ denotes the probability that an arbitrary training instance has the distance less than and equal to d from $t(x,y)$, respectively. $P_{\rm l}(x,y,d)$ and $P_{\rm d}(x,y,d)$ are given by

$$P_{\rm d}(x,y,d) = P_{\rm dp}(x,y,d)+P_{\rm dn}(x,y,d),$$
$$P_{\rm l}(x,y,d) = \sum_{e=0}^{d-1}\{P_{\rm dp}(x,y,e)+P_{\rm dn}(x,y,e)\}.$$

Finally, we compute $P_{\rm sp}(k,x,y,d,a,b)$ by considering the following situations. When exactly a training instances have the distance less than d from $t(x,y)$, we let exactly u out of these a instances have the positive class label. We use $P_{\rm lp}^{\rm u}(a,u)$ to denote the probability that this situation occurs. To get k nearest neighbors for $t(x,y)$, k-NN selects exactly $(k-a)$ out of b training instances with distance d from $t(x,y)$. We let w out of these $(k-a)$ instances have the positive class label. Under these situations, if we have $u+w > k/2$, then k-NN always classifies $t(x,y)$ as positive, and if $u+w < k/2$, then always classifies $t(x,y)$ as negative. Moreover, if $u+w = k/2$, then $t(x,y)$ is classified as positive with the probability of $1/2$. Hence, we can represent $P_{\rm sp}(k,x,y,d,a,b)$ as

$$P_{\rm sp}(k,x,y,d,a,b) = \sum_{u=0}^{a}\sum_{v=0}^{b} P_{\rm lp}^{\rm u}(a,u)P_{\rm dp}^{\rm v}(b,v) \times \left\{ \sum_{w=\lceil\frac{k+1}{2}\rceil-u}^{v} P_{\rm dp}^{\rm w}(k,a,b,v,w) + \frac{1}{2}P_{\rm dp}^{\rm w}(k,a,b,v,\frac{k}{2}-u) \right\}.$$

In this equation, $P_{\rm dp}^{\rm v}(b,v)$ denotes the probability that exactly v out of b training instances with distance d from $t(x,y)$ have the positive class label. Moreover, $P_{\rm dp}^{\rm w}(k,a,b,v,w)$ denotes the probability that exactly w out of $(k-a)$ training instances, selected by k-NN from b instances with distance d from $t(x,y)$, have the positive class label. Note that $P_{\rm dp}^{\rm w}(k,a,b,v,\frac{k}{2}-x)$ becomes zero, when k is an odd number.

We can represent $P_{\rm lp}^{\rm u}(a,u)$ as

$$P_{\rm lp}^{\rm u}(a,u) = \binom{a}{u}\left(\frac{P_{\rm lp}(x,y,d)}{P_{\rm l}(x,y,d)}\right)^u\left(\frac{P_{\rm ln}(x,y,d)}{P_{\rm l}(x,y,d)}\right)^{a-u},$$

where $P_{\rm lp}(x,y,d)$ ($P_{\rm ln}(x,y,d)$, *resp.*) is the probability that an arbitrary instance has the distance less than d from $t(x,y)$ and has the positive (negative, *resp.*) class label. $P_{\rm lp}(x,y,d)$ and $P_{\rm ln}(x,y,d)$ are give by

$$P_{\rm lp}(x,y,d) = \sum_{e=0}^{d-1}P_{\rm dp}(x,y,e),$$
$$P_{\rm ln}(x,y,d) = \sum_{e=0}^{d-1}P_{\rm dn}(x,y,e).$$

In a similar manner, we can represent $P_{\rm dp}^{\rm v}(b,v)$ as

$$P_{\rm dp}^{\rm v}(b,v) = \binom{b}{v}\left(\frac{P_{\rm dp}(x,y,d)}{P_{\rm d}(x,y,d)}\right)^v\left(\frac{P_{\rm dn}(x,y,d)}{P_{\rm d}(x,y,d)}\right)^{b-v}$$

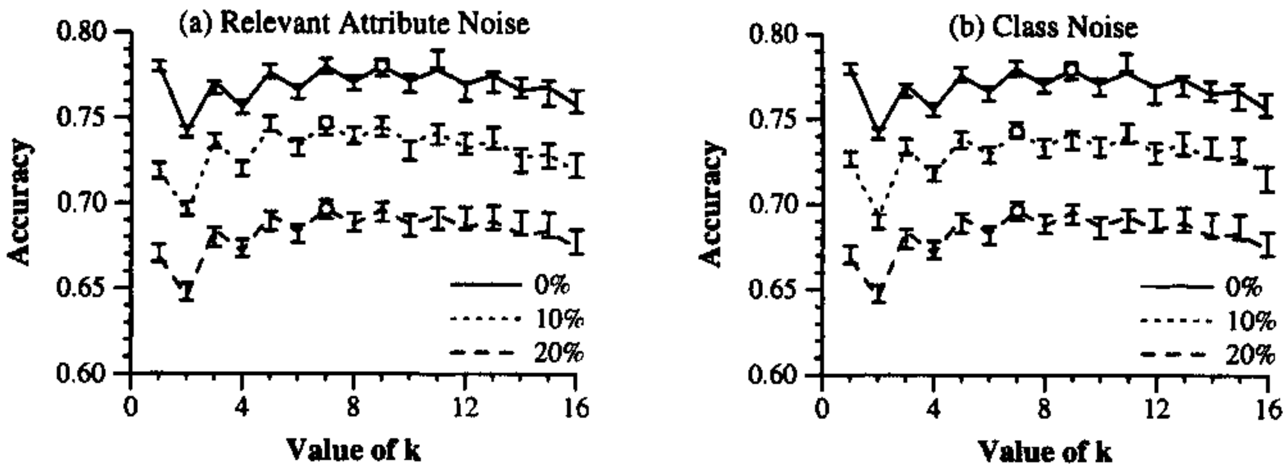


Figure 1: The predictive accuracy of fc-NN against the value of fc for a 3-of-5/2 concept. The lines and the error bars represent the theoretical results and the empirical results of Monte Carlo simulations. Each circle denotes the accuracy for the optimal fc at the corresponding noise level. The number of training instances is fixed at 32.

When fc-NN selects exactly (A: - *a)* out of *b* training instances with distance *d* from *t(x,y),* these *(k - a)* training instances comprise exactly *w* out of *v* instances with the positive class label and exactly (fc - *a - w)* out of (6 - *v)* instances with the negative class label. Hence, $P_{\rm dp}^{\rm w}(k,a,b,v,w)$ is given by

$$P_{\rm dp}^{\rm w}(k,a,b,v,w)=\frac{\binom{v}{w}\binom{b-v}{k-a-w}}{\binom{b}{k-a}}.$$

We have computed the predictive accuracy of fc-NN in the case where each type of noise affects only the training instances. When noise affects test instances, the appearance probability for an arbitrary test instance with the positive (negative, *resp.)* class label in *I(x,y)* is $P_p(x,y)$ ($P_n(x,y)$, *resp.).* Hence, when noise affects both test and training instances, the predictive accuracy of fc-NN after TV training instances can be represented as

$$A(k)=\sum_{x=0}^{n}\sum_{y=0}^{l}\{P_{\rm n}(x,y)\,(1-P_{\rm pos}(k,x,y)) +P_{\rm p}(x,y)P_{\rm pos}(k,x,y)\}.$$

4 Predicted Behavior

Using the accuracy function described in Section 3, we explore the predicted behavior of fc-NN. Although the accuracy function was obtained for both noise-free and noisy test instances, our exploration deals with only noise-free test instances for lack of space. Moreover, we investigate the effects of each individual noise type on fc-NN.

For irrelevant attribute noise, we can formally prove the following claim from the accuracy function (the proof is omitted here due to space limitations).

Claim 1

If *the probability of occurrence for irrelevant attribute is* 1/2, *then the predictive accuracy of k-NN for m-of-n/l concepts is entirely independent of the noise rate for irrelevant attributes.*

From this claim, we can expect that irrelevant attribute noise does not greatly affect the classification accuracy of fc-NN, *nor* the optimal value of fc. Therefore, the following discussions focus on the effects of relevant attribute noise and class noise. Throughout our exploration, we set the probabilities of occurrence for both relevant and irrelevant attributes to 1/2.

In addition to the theoretical results from the accuracy function, we give the results of Monte Carlo simulations to confirm our analysis. For each case, 500 training sets are randomly generated in accordance with each noise rate, then the data is collected as the classification accuracy measured over the entire space of noise-free instances. For each case, we report a 95% confidence interval for the mean accuracy of 500 data items. In the following figures, the error bar indicates this confidence interval.

4.1 Accuracy against Value of *k*

First, we report the predicted behavior of fc-NN against the value of fc at several levels of noise, as shown in Figure 1. In this figure, the number of training instances is fixed at 32, and the target is a 3-of-5/2 concept. The lines indicate the theoretical results from the accuracy function, and the error bars represent the empirical results of Monte Carlo simulations. The theoretical results agree well with the empirical ones for both relevant attribute noise and class noise.

Figure 1 shows that the predictive accuracy of fc-NN markedly drops off for each noise level when fc is an even number. This negative influence of an even number for fc on the accuracy is caused by a random determination of class when a tie occurs. This negative influence suggests that a choice of even number for fc is undesirable when applying fc-NN.

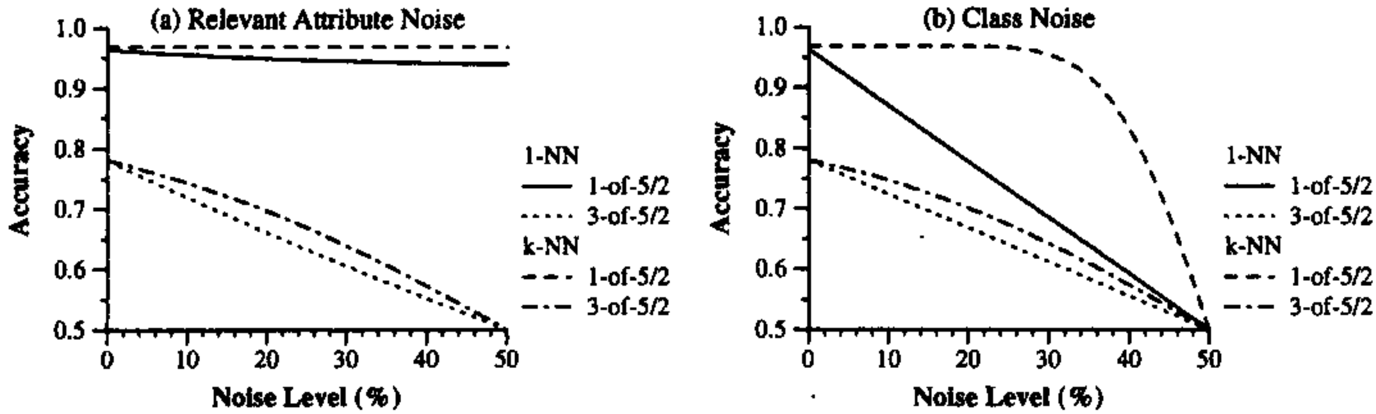


Figure 2: The effects of noise on the predictive accuracy of k-NN. Each curve for k-NN indicates the accuracy of k-NN with the optimal value of A;. The number of training instances is fixed at 32.

In Figure 1, each circle represents the predictive accuracy for the optimal value of *k* at the corresponding noise level. For each odd number for *k,* the accuracy of k-NN for a 0% noise level has two peaks. One appears at *k* = 1, while the other appears at the optimal value of *k.* In contrast, the accuracy for a 10% noise level and a 20% level have one peak at the corresponding optimal *k.* This is because the peak at *k* = 1 disappears due to the effect of noise.

4.2 Effects of Noise on the Accuracy

We further investigate the effects of noise on the predictive accuracy of k-NN, as shown in Figure 2. In this figure, the number of training instances is fixed at 32, and all curves come from the theoretical results. Each curve for k-NN represents the predictive accuracy of *k*-NN with the optimal value of *k* at each noise level.

Figure 2(a) shows the effects of relevant attribute noise on the predictive accuracies of 1-NN and the optimal *k*-NN. When the noise level is 0%, the accuracy of 1-NN is comparable to that for the optimal k-NN, for both 1-of-5/2 and 3-of-5/2 concepts. However, the predictive accuracy of 1-NN almost linearly decreases with an increase in the noise level. For a 50% noise level, the accuracy of 1-NN equals that of a random prediction algorithm which predicts the same class as that for a randomly selected training instance. These observations suggest that 1-NN is strongly sensitive to relevant attribute noise. In contrast, the predictive accuracy of the optimal k-NN exhibits slower degradation. For the disjunctive concept (l-of-5/2 concept), the accuracy of the optimal k-NN is *not* greatly changed as the noise level increases.

Figure 2(b) shows the effects of class noise on the predictive accuracies of 1-NN and the optimal k-NN. For the 3-of-5/2 concept, both 1-NN and the optimal k-NN exhibit similar behavior to the corresponding tests with relevant attribute noise. However, the effects of class noise on the accuracy differ entirely from ones of relevant attribute noise for the disjunctive concept. The predictive accuracy of 1-NN linearly decreases to 0.5. In contrast, the optimal k-NN's accuracy does *not* substantially change until about a 30% noise level, whereafter it rapidly decreases to 50%.

These observations show that the predictive accuracy of 1-NN is strongly affected by both relevant attribute noise and class noise. Also, they suggest that we can restrain the degradation in the predictive accuracy of k-NN caused by an increase in noise level by optimizing the value of *k*.

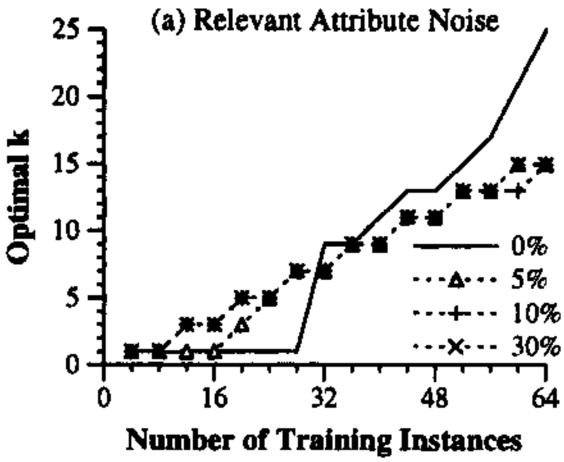
4.3 Optimal Value of *k*

Finally, we give the relationship between the optimal value of *k* and the number of training instances in noisy domains, as shown in Figure 3. In this figure, the optimal value of *k* comes from the theoretical results, and the target is a 3-of-5/2 concept. In the following discussions, we use *N* to refer to the number of training instances.

For a 0% noise level, the optimal value of *k* remains *k* = 1 until *N* = 28. There is a rapid increase in the optimal *k* at *N* = 32, and then the optimal *k* almost linearly increases with an increase of *N.* This rapid increase is caused by the change of the peak given the highest accuracy from *k* = 1 to another (as mentioned in Section 4.1, k-NN's predictive accuracy has two peaks).

For each level (5%, 10%, and 30%) for both relevant attribute noise and class noise, the optimal value of *k* is changed from *k* = 1 to another at small *N.* This observation can be explained by the strong sensitivity of the accuracy of 1-NN to both relevant attribute noise and class noise (as mentioned in Section 4.2). That is, the peak at *k* = 1 disappears due to the effect of noise, even though *N* is a small number. After changing from *k* = 1 to another, the optimal value of *k* almost linearly increases with an increase of *N.*

These observations from Figure 3 show that the optimal value of *k* almost linearly increases with an increase of *N* after the optimal *k* is changed from *k* = 1 to another, regardless of the noise level for both relevant attribute noise and class noise. That is, the optimal value of *k* strongly depends upon the number of training instances in noisy domains.



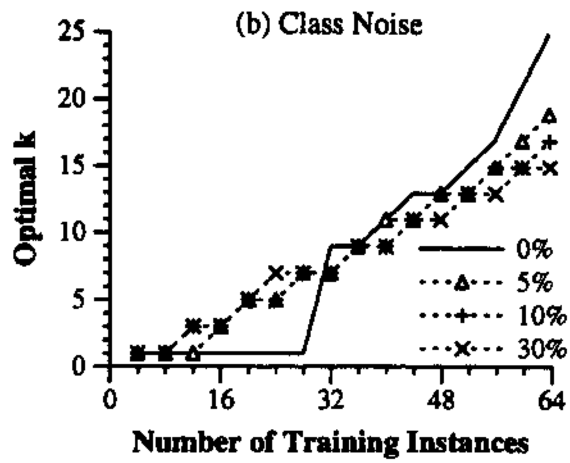


Figure 3: The optimal value of *k* against the number of training instances for a 3-of-5/2 concept.

5 Conclusion

In this paper, we presented an average-case analysis of the fc-nearest neighbor classifier (k-NN) for *m-of-n/l* target concepts in noisy domains. Our analysis dealt with three types of noise: relevant attribute noise, irrelevant attribute noise, and class noise.

We formally defined the predictive accuracy of k-NN as a function of the domain characteristics. Using the accuracy function, we explored the predicted behavior of k-NN for each type of noise. The predictive accuracy of k-NN was given at various levels of noise, then the noise effects on k-NN's accuracy were shown. We also show that the optimal value of *k* almost linearly increases with an increase in the number of training instances in noisy domains. Our analysis was supported with Monte Carlo simulations.

In the future, we will extend the framework of average-case analysis to relax many restrictions such as Boolean attributes, a fixed class of target concepts, and a fixed distribution over the instance space. Using the extended framework, we would like to analyze learning algorithms to give more useful insights into their practical applications.

References

[Aha and Kibler, 1989] Aha, D. and Kibler, D. Noise-Tolerant Instance-Based Learning Algorithms. In *Proceedings of IJCAI-89,* pages 794-799. Morgan Kaufmann.

[Aha *et al,* 1991] Aha, D., Kibler, D., and Albert, M. Instance-Based Learning Algorithms. *Machine Learning,* 6:37-66.

[Albert and Aha, 1991] Albert, M. and Aha, D. Analyses of Instance-Based Learning Algorithms. In *Proceedings of AAAI-91,* pages 553-558. AAAI Press/MIT Press.

[Cover and Hart, 1967] Cover, T. and Hart, P. Nearest Neighbor Pattern Classification. *IEEE Transactions on Information Theory,* 13(l):21-27.

[Cover, 1968] Cover, T. Estimation by the Nearest Neighbor Rule. *IEEE Transactions on Information Theory,* 14(1):50~55.

[Creecy *et al,* 1992] Creecy, H., Masand, M., Smith, J., and Waltz, D. Trading Mips and Memory for Knowledge Engineering. *Communications of the ACM,* 35(8):48-63.

[Drakopoulos, 1995] Drakopoulos, J. Bounds on the Classification Error of the Nearest Neighbor Rule. In *Proceedings of ICML-95,* pages 203-208. Morgan Kaufmann.

[Langley and Iba, 1993] Langley, P. and Iba, W. Average-Case Analysis of a Nearest Neighbor Algorithm. In *Proceedings of IJCAI-93,* pages 889-894. Morgan Kaufmann.

[Murphy and Pazzani, 1991] Murphy, P. and Pazzani, M. ID2-of-3: Constructive Induction of *M-of-N* Concepts for Discriminators in Decision Trees. In *Proceedings of IWML-91,* pages 183-187. Morgan Kaufmann.

[Okamoto and Satoh, 1995] Okamoto, S. and Satoh, K. An Average-Case Analysis of k-Nearest Neighbor Classifier. In *Proceedings of ICCBR-95* (Veloso, M. and Aamodt, A. Eds., *LNAI,* 1010), pages 243-264. Springer-Verlag.

[Okamoto and Yugami, 1996] Okamoto, S. and Yugami, N. Theoretical Analysis of the Nearest Neighbor Classifier in Noisy Domains. In *Proceedings of ICML-96,* pages 355-363. Morgan Kaufmann.

[Pazzani and Sarrett, 1992] Pazzani, M. and Sarrett, W. A Framework for Average Case Analysis of Conjunctive Learning Algorithms. *Machine Learning,* 9:349-372.

[Rachlin *et al,* 1994] Rachlin, J., Kasif, S., Salzberg, S., and Aha, D. Toward a Better Understanding of Memory-Based Reasoning Systems. In *Proceedings of ICML-94,* pages 242-250. Morgan Kaufmann.