Available at www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®
Theoretical Computer Science 298 (2003) 207–233
Theoretical Computer Science
www.elsevier.com/locate/tcs

Effects of domain characteristics on instance-based learning algorithms

Seishi Okamoto*, Nobuhiro Yugami
Fujitsu Laboratories, 1-9-3 Nakase, Mihama-ku, 213-8588 Chiba, Japan

Abstract

This paper presents average-case analyses of instance-based learning algorithms. The algorithms analyzed employ a variant of k-nearest neighbor classifier (k-NN). Our analysis deals with a monotone m-of-n target concept with irrelevant attributes, and handles three types of noise: relevant attribute noise, irrelevant attribute noise, and class noise. We formally represent the expected classification accuracy of k-NN as a function of domain characteristics including the number of training instances, the number of relevant and irrelevant attributes, the threshold number in the target concept, the probability of each attribute, the noise rate for each type of noise, and k. We also explore the behavioral implications of the analyses by presenting the effects of domain characteristics on the expected accuracy of k-NN and on the optimal value of k for artificial domains.

Keywords: Instance-based learning; k-Nearest neighbor classifier; Average-case analysis; Expected accuracy; Optimal value of k

1. Introduction

Instance-based learning (IBL) is one of the most widely applied learning framework, and many IBL algorithms have performed well in challenging learning tasks [1,2,7,26,30,32,33]. Most IBL algorithms are based on a k-nearest neighbor classifier (k-NN), originated in the field of the pattern recognition [8,9,11]. Informally, k-NN is explained as follows: k-NN stores the entire training set into memory. When a test instance is given, k-NN selects the k nearest training instances to the test instance, and predicts the majority class of these k instances as the class of the test instance.

* Corresponding author.
E-mail address: seishi@flab.fujitsu.co.jp (S. Okamoto).

0304-3975/03/$ - see front matter
PII: S0304-3975(02)00424-3

Many researches have theoretically analyzed the learning behavior of k-NN by comparing it with Bayesian induction, with the probably approximately correct (PAC) learning model, and with the best-case model. Cover and Hart [6] showed that the upper bound of k-NN error rate is twice the optimal Bayes risk under the assumption of an infinite number of training instances. Cover [5] also showed that k-NN risk converges to the optimal Bayes risk as k approaches infinity. Rachlin et al. [26] showed that their IBL algorithm PEBLS is not less accurate than the Bayesian classifier in the limit. Although these analyses are important, all of these studies assumed an infinite number of training instances, which are rarely available in practice. Moreover, these analyses assumed noise-free instances and did not deal with irrelevant attributes.

By using the PAC learning model [13,31], Aha et al. [2] showed the learnability and sample complexity of an IBL algorithm, IB1, for a class of closed regions bounded by a fixed length. Albert and Aha [3] extended this study to k-NN, and presented sample complexity for k-NN for the same target concept. By using the best-case analysis, Salzberg et al. [28,29] showed sample complexities of 1-NN for several types of geometric target concepts. Although these studies gave quite general results, their predictions of the learning behavior of IBL algorithms are often far from those observed in practice. This means that it is difficult to relate their results to experimental ones directly. Also, all of these studies assumed noise-free instances and did not take into account irrelevant attributes.

The framework of average-case analysis is useful for understanding the effects of domain characteristics, such as the number of training instances, the number of attributes, and noise rate on the behavior of a learning algorithm [24]. This is because the average-case analysis is based on the formal computation of the behavior of the learning algorithm as a function of these characteristics. Moreover, this framework enables us to explore the *average-case* behavior of the learning algorithm. Hence, formal results provided in this framework can be directly related to empirical ones.

Many learning algorithms have been analyzed using this framework, such as conjunctive learning algorithms [14,27,24], a Bayesian classifier [17,18], and decision-tree induction [15]. Also, average-case analyses of IBL algorithms have been presented, including 1-NN [16,21,23] and k-NN [20,22].

In this paper, we present average-case analyses of the k-NN classifier. Our analyses deal with m-of-n concepts whose positive instances are defined by having m or more of n relevant attributes, and with irrelevant attributes which play no role in the target concept. Moreover we handle three types of noise: relevant attribute noise, irrelevant attribute noise, and class noise. Our analyses are individually presented in a noise-free domain and in a noisy domain.

First, in the noise-free domain, our analysis formally represents the expected classification accuracy of k-NN after a certain number of training instances are given. The expected accuracy is represented as a function of domain characteristics including the number of training instances, the number of relevant and irrelevant attributes, the threshold number in the target concept, the probability of each attribute, and k. We also explore the behavioral implications of the analysis by predicting the effects of each domain characteristic on the expected accuracy and on the optimal value of k to achieve the highest accuracy for artificial domains.

Next, in the noisy domain, our analysis formally expresses the expected accuracy of k-NN as a function of the domain characteristics, including the noise rate for each type of noise. Then, we examine the behavioral implications of the analysis by presenting the effects of each type of noise on the expected accuracy and on the optimal value of k.

In closing, we discuss the implications of this work, and point out interesting directions for future research.

2. Problem description

This section gives the problem description used in our average-case analyses of an IBL algorithm employing a variant of k-NN classifier.

As the target concept, our analysis deals with a monotone m-of-n function of n relevant Boolean attributes which returns TRUE (positive class label) if at least m out of these n attributes occur (i.e., have the value of 1), and returns FALSE (negative class label) otherwise [19,25]. We further handle irrelevant Boolean attributes that play no role in the target concept. We express the m-of-n concept with l irrelevant attributes as the m-of-n/l concept. Then, given a certain vector $(w_1,\ldots,w_{n+l})\in\{0,1\}^{n+l}$ where $|\{w_i \mid w_i=1\}|=n$ and $|\{w_i \mid w_i=0\}|=l$, the m-of-n/l concept can be represented as

$$f:(a_1,\ldots,a_{n+l})\in\{0,1\}^{n+l}\mapsto\begin{cases}1 & \text{if } \sum_{i=1}^{n+l} w_i a_i \geqslant m,\\ 0 & \text{otherwise.}\end{cases} \tag{1}$$

Note that each a_i is a relevant attribute if $w_i=1$, and is an irrelevant attribute otherwise.

To generate probability distributions over instance space $\{0,1\}^{n+l}$, our analysis assumes that every relevant and irrelevant attribute independently occurs with a certain fixed probability p and q, respectively. Each instance $\chi\in\{0,1\}^{n+l}$ is independently drawn from the instance space in accordance with these probabilities, and then the class label $f(\chi)$ is attached to χ.

Each type of noise is introduced by the following common definition. Relevant attribute noise flips the value of every relevant attribute in each instance with a certain fixed probability σ_r $(0\leqslant\sigma_r\leqslant 1)$. In a similar way, irrelevant attribute noise flips the value of every irrelevant attribute with a certain fixed probability σ_i $(0\leqslant\sigma_i\leqslant 1)$. Class noise replaces the class label for each instance with its opposite with a certain fixed probability σ_c $(0\leqslant\sigma_c\leqslant 1)$. We assume that each noise type independently affects each instance.

The IBL algorithms analyzed employ a variant of k-NN classifier explained as follows: k-NN receives a set of training instances (k-NN knows the class label for each training instance), and stores all training instances into memory. When a test instance is given (the class label for test instance is unknown to k-NN), k-NN selects k nearest training instances to the test instance according to the Hamming distance (i.e., the number of attributes on which two instances differ) among all training instances. Then k-NN predicts that the test instance belongs to a majority class among these selected

Table 1
Domain characteristics used in the analysis

k	Number of nearest neighbors
N	Number of training instances
n	Number of relevant attributes
l	Number of irrelevant attributes
m	Threshold value in target concepts
p	Occurrence probability for relevant attribute
q	Occurrence probability for irrelevant attribute
σ_c	Noise rate for class
σ_r	Noise rate for relevant attribute
σ_i	Noise rate for irrelevant attribute

k nearest training instances. While several tie-breaking procedures have been proposed for the selection of k-NNs [4,10], we assume that k-NN randomly breaks a tie case for k-NNs. Moreover, if the selected k-NNs contain exactly the same number of positive and negative training instances, then k-NN randomly determines the class of the test instance. This situation can occur only when k is an even number.

The characteristics of the problem domain are summarized in Table 1. Our analyses will express the expected accuracy of the k-NN classifier as a function of these characteristics. Here, the expected accuracy is the probability that the classifier predicts correctly the class label for an arbitrary test instance. To get accuracy function, we will often use the binomial probability, the trinomial probability, and the hypergeometric probability. The expressions of these probabilities are given as the following definitions.

Definition 1. For any α $(0 \leqslant \alpha \leqslant 1)$, and for any integers a and x $(x \leqslant a)$, the binomial probability is expressed as

$$B(x; a, \alpha) = \begin{cases} \binom{a}{x} \alpha^x (1-\alpha)^{a-x} & \text{if } x \geqslant 0, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Definition 2. For any α, β $(0 \leqslant \alpha, \beta \leqslant 1)$, and for any integers a, x $(x \leqslant a)$, and y $(y \leqslant a - x)$, the trinomial probability is expressed as

$$T(x, y; a, \alpha, \beta) = \begin{cases} \binom{a}{x}\binom{a-x}{y} \alpha^x \beta^y (1-\alpha-\beta)^{a-x-y} & \text{if } x \geqslant 0 \text{ and } y \geqslant 0, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Definition 3. For any integers a, b, and c $(0 \leqslant b, c \leqslant a)$, and for any integer x, the hypergeometric probability is expressed as

$$H(x;a,b,c) = \begin{cases} \dfrac{\binom{b}{x}\binom{a-b}{c-x}}{\binom{a}{c}} & \text{if} \quad x \geqslant \max(0, c-a+b) \text{ and} \\ & \quad x \leqslant \min(b,c), \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

3. Analysis in a noise-free domain

In this section, we assume that every instance is noise-free, and present an average-case analysis of the k-NN classifier.

First, our analysis represents the expected classification accuracy of k-NN for the m-of-n/l target concept, after k-NN receives N training instances. The expected accuracy is represented as a function of the domain characteristics given in Table 1 with the exception of the noise rate for each type of noise: σ_r, σ_i, and σ_c. However, to avoid complicated notation, we do *not* explicitly express these characteristics as the arguments of the accuracy function. Then, using the accuracy function, we make average-case predictions about the behavior of k-NN, including the effects of each domain characteristic on the expected accuracy of k-NN, and on the optimal value of k.

3.1. Expected accuracy

In this subsection, our analysis represents the expected accuracy of the k-NN for m-of-n/l target concepts in the noise-free domain.

To simplify the computation of the expected accuracy of k-NN, our analysis uses a set of instances in which x relevant attributes and y irrelevant attributes simultaneously occur. This set is referred to as $\Psi(x,y)$, and let $P_{\text{occ}}(x,y)$ be the probability that an arbitrary instance drawn from the instance space belongs to $\Psi(x,y)$. From the assumption of independence of attributes, using the binomial probabilities, we can express $P_{\text{occ}}(x,y)$ as

$$P_{\text{occ}}(x,y) = B(x;n,p)B(y;l,q). \tag{5}$$

Let $P_{\text{pos}}(x,y)$ be the probability that k-NN classifies any test instance in $\Psi(x,y)$ as positive after an arbitrary training set with the size of N. For any test instance in $\Psi(x,y)$, this probability has the same value. Moreover, from the definition of the target concept, we clearly have that any test instance in $\Psi(x,y)$ belongs to the negative class if $x < m$ and to the positive class label if $x \geqslant m$. Therefore, after k-NN receives N training instances, the expected accuracy of k-NN for the m-of-n/l target concept can be represented as

$$\mathscr{A} = \sum_{y=0}^{l} \left\{ \sum_{x=0}^{m-1} P_{\text{occ}}(x,y)(1 - P_{\text{pos}}(x,y)) + \sum_{x=m}^{n} P_{\text{occ}}(x,y)P_{\text{pos}}(x,y) \right\}. \tag{6}$$

Let $\tau(x,y)$ be an arbitrary test instance in $\Psi(x,y)$. To compute $P_{\mathrm{pos}}(x,y)$, we use the distance from $\tau(x,y)$ to its kth nearest training instances. When the kth nearest training instance has the distance d $(0 \leqslant d \leqslant n+l)$ from $\tau(x,y)$, we consider the situation that exactly N_{l} out of N training instances occur with the distance less than d from $\tau(x,y)$ and exactly N_{e} training instances appear with the distance d. Let $P_{\mathrm{num}}(x,y,d,N_{\mathrm{l}},N_{\mathrm{e}})$ be the probability that this situation occurs. Also, when this situation occurs, let $P_{\mathrm{sp}}(x,y,d,N_{\mathrm{l}},N_{\mathrm{e}})$ be the probability that k-NN classifies $\tau(x,y)$ as positive. Then, we can represent $P_{\mathrm{pos}}(x,y)$ by summing over all possible numbers of $N_{\mathrm{l}}, N_{\mathrm{e}}$, and d, in each case multiplying $P_{\mathrm{sp}}(x,y,d,N_{\mathrm{l}},N_{\mathrm{e}})$ by $P_{\mathrm{num}}(x,y,d,N_{\mathrm{l}},N_{\mathrm{e}})$. For the possible regions of N_{l} and N_{e}, we have clearly $0 \leqslant N_{\mathrm{l}} \leqslant k-1$ and $(k-N_{\mathrm{l}}) \leqslant N_{\mathrm{e}} \leqslant (N-N_{\mathrm{l}})$. That is, $P_{\mathrm{pos}}(x,y)$ can be expressed as

$$P_{\mathrm{pos}}(x,y) = \sum_{d=0}^{n+l} \sum_{N_{\mathrm{l}}=0}^{k-1} \sum_{N_{\mathrm{e}}=k-N_{\mathrm{l}}}^{N-N_{\mathrm{l}}} P_{\mathrm{num}}(x,y,d,N_{\mathrm{l}},N_{\mathrm{e}}) P_{\mathrm{sp}}(x,y,d,N_{\mathrm{l}},N_{\mathrm{e}}). \tag{7}$$

First, we represent $P_{\mathrm{num}}(x,y,d,N_{\mathrm{l}},N_{\mathrm{e}})$ by letting $P_{\mathrm{e}}(x,y,d)$ and $P_{\mathrm{l}}(x,y,d)$ be the probabilities that an arbitrary training instance occurs with the distance equal to d and less than d from $\tau(x,y)$, respectively. Then, $P_{\mathrm{num}}(x,y,d,N_{\mathrm{l}},N_{\mathrm{e}})$ can be represented by using the trinomial probability for $P_{\mathrm{e}}(x,y,d)$ and $P_{\mathrm{l}}(x,y,d)$. That is, we can obtain $P_{\mathrm{num}}(x,y,d,N_{\mathrm{l}},N_{\mathrm{e}})$ as

$$P_{\mathrm{num}}(x,y,d,N_{\mathrm{l}},N_{\mathrm{e}}) = T(N_{\mathrm{l}},N_{\mathrm{e}};N,P_{\mathrm{l}}(x,y,d),P_{\mathrm{e}}(x,y,d)). \tag{8}$$

For any integer u $(0 \leqslant u \leqslant n)$ and v $(0 \leqslant v \leqslant l)$, let $\xi(u,v)$ be an arbitrary training instance in $\Psi(u,v)$, and $P_{\mathrm{dis}}(x,y,u,v,e)$ be the probability that $\xi(u,v)$ has the distance e from $\tau(x,y)$. Then, we can obtain $P_{\mathrm{e}}(x,y,d)$ by summing the product of $P_{\mathrm{occ}}(u,v)$ and $P_{\mathrm{dis}}(x,y,u,v,d)$ over all possible numbers of u and v. That is, we can represent $P_{\mathrm{e}}(x,y,d)$ as

$$P_{\mathrm{e}}(x,y,d) = \sum_{u=0}^{n} \sum_{v=0}^{l} P_{\mathrm{occ}}(u,v) P_{\mathrm{dis}}(x,y,u,v,d). \tag{9}$$

Also, $P_{\mathrm{l}}(x,y,d)$ is clearly given by

$$P_{\mathrm{l}}(x,y,d) = \sum_{u=0}^{n} \sum_{v=0}^{l} P_{\mathrm{occ}}(u,v) \sum_{e=0}^{d-1} P_{\mathrm{dis}}(x,y,u,v,e). \tag{10}$$

To compute $P_{\mathrm{dis}}(x,y,u,v,e)$, we use the number of relevant attributes which take the value of 1 in both $\tau(x,y)$ and $\xi(u,v)$, and use the analogous number for irrelevant attributes. We denote the former number as s_{r} and the latter one as s_{i}. Then, the number of relevant attributes with the value of 1 in $\tau(x,y)$ but with the value of 0 in $\xi(u,v)$ is $x-s_{\mathrm{r}}$. In contrast, the number of relevant attribute with the value 0 in $\tau(x,y)$ but with 1 in $\xi(u,v)$ is $u-s_{\mathrm{r}}$. Similarly, the number of irrelevant attributes with the value 1 in $\tau(x,y)$ but with 0 in $\xi(u,v)$ is $y-s_{\mathrm{i}}$, and the number of irrelevant attributes with 0 in $\tau(x,y)$ but with 1 in $\xi(u,v)$ is $v-s_{\mathrm{i}}$. Using these, the distance e between

$\tau(x, y)$ and $\xi(u, v)$ is given by

$$\begin{aligned} e &= (x - s_r) + (u - s_r) + (y - s_i) + (v - s_i) \\ &= x + u + y + v - 2(s_r + s_i). \end{aligned} \tag{11}$$

Rearranging, we have

$$s_r + s_i = \frac{x + u + y + v - e}{2}. \tag{12}$$

For the possible regions of s_r and s_i, we have

$$\max(0, x + u - n) \leqslant s_r \leqslant \min(x, u). \tag{13}$$

$$\max(0, y + v - l) \leqslant s_i \leqslant \min(y, v). \tag{14}$$

Let $\mathscr{S}$ be the set of all pairs (s_r, s_i) that satisfy all of conditions (12), (13), and (14). Then, we can represent $P_{dis}(x, y, u, v, e)$ by summing over all possible pairs of (s_r, s_i) in $\mathscr{S}$, in each case multiplying the hypergeometric probabilities according to the occurrence of relevant and irrelevant attributes. That is, we can represent $P_{dis}(x, y, u, v, e)$ as

$$P_{dis}(x, y, u, v, e) = \sum_{(s_r, s_i) \in \mathscr{S}} H(s_r; n, x, u) H(s_i; l, y, v), \tag{15}$$

where

$$\mathscr{S} = \left\{ (s_r, s_i) \;\middle|\; \begin{array}{l} s_r + s_i = \dfrac{x + u + y + v - e}{2}, \\ \max(0, x + u - n) \leqslant s_r \leqslant \min(x, u), \\ \max(0, y + v - l) \leqslant s_i \leqslant \min(y, v). \end{array} \right\} \tag{16}$$

Note that we have $P_{dis}(x, y, u, v, e) = 0$, when $\mathscr{S} = \emptyset$.

Next, we compute $P_{sp}(x, y, d, N_l, N_e)$ in Eq. (7). Let $P_{lp}(x, y, d)$ be the probability that an arbitrary training instance appears with the distance less than d from $\tau(x, y)$ and has the positive class label. From Eq. (10), this probability is clearly given by

$$P_{lp}(x, y, d) = \sum_{u=m}^{n} \sum_{v=0}^{l} P_{occ}(u, v) \sum_{e=0}^{d-1} P_{dis}(x, y, u, v, e). \tag{17}$$

When exactly N_l training instances have the distance less than d from $\tau(x, y)$, let us consider the situation that exactly N_{lp} out of these N_l instances belong to positive class. We denote the occurrence probability for this situation by $P_{lps}(x, y, d, N_l, N_{lp})$. Using the binomial probability, we can represent this probability as

$$P_{lps}(x, y, d, N_l, N_{lp}) = B\left(N_{lp}; N_l, \frac{P_{lp}(x, y, d)}{P_l(x, y, d)}\right). \tag{18}$$

In a similar way, when exactly N_e training instances have the distance d from $\tau(x, y)$, we consider the case that exactly N_{ep} out of these N_e instances belong to positive. Let

$P_{\mathrm{eps}}(x, y, d, N_{\mathrm{e}}, N_{\mathrm{ep}})$ be the occurrence probability for this case, and $P_{\mathrm{ep}}(x, y, d)$ be the probability that an arbitrary training instance occurs with the distance of d from $\tau(x, y)$ and has the positive label. From Eq. (9), the latter probability is clearly given by

$$P_{\mathrm{ep}}(x, y, d) = \sum_{u=m}^{n} \sum_{v=0}^{l} P_{\mathrm{occ}}(u, v) P_{\mathrm{dis}}(x, y, u, v, d). \tag{19}$$

Using the binomial probability, we can represent $P_{\mathrm{eps}}(x, y, d, N_{\mathrm{e}}, N_{\mathrm{ep}})$ as

$$P_{\mathrm{eps}}(x, y, d, N_{\mathrm{e}}, N_{\mathrm{ep}}) = B\left(N_{\mathrm{ep}}; N_{\mathrm{e}}, \frac{P_{\mathrm{ep}}(x, y, d)}{P_{\mathrm{e}}(x, y, d)}\right). \tag{20}$$

When exactly N_{lp} out of N_{l} training instances with the distance less than d from $\tau(x, y)$ have the positive class label and exactly N_{ep} out of N_{e} instances with the distance d have the positive label, let $P_{\mathrm{ksp}}(N_{\mathrm{l}}, N_{\mathrm{e}}, N_{\mathrm{lp}}, N_{\mathrm{ep}})$ be the probability that k-NN classifies $\tau(x, y)$ as positive. Then, we can obtain the probability $P_{\mathrm{sp}}(x, y, d, N_{\mathrm{l}}, N_{\mathrm{e}})$ as

$$P_{\mathrm{sp}}(x, y, d, N_{\mathrm{l}}, N_{\mathrm{e}}) = \sum_{N_{\mathrm{lp}}=0}^{N_{\mathrm{l}}} \left(P_{\mathrm{lps}}(x, y, d, N_{\mathrm{l}}, N_{\mathrm{lp}}) \sum_{N_{\mathrm{ep}}=0}^{N_{\mathrm{e}}} P_{\mathrm{eps}}(x, y, d, N_{\mathrm{e}}, N_{\mathrm{ep}}) P_{\mathrm{ksp}}(N_{\mathrm{l}}, N_{\mathrm{e}}, N_{\mathrm{lp}}, N_{\mathrm{ep}}) \right). \tag{21}$$

At this point, we have only to compute the probability $P_{\mathrm{ksp}}(N_{\mathrm{l}}, N_{\mathrm{e}}, N_{\mathrm{lp}}, N_{\mathrm{ep}})$. To get k nearest training instances for $\tau(x, y)$, k-NN randomly selects exactly $(k - N_{\mathrm{l}})$ out of N_{e} training instances with the distance d from $\tau(x, y)$. Let us consider the case that N_{epk} out of these $(k - N_{\mathrm{l}})$ instances belong to positive class. The occurrence probability for this case is denoted with $P_{\mathrm{epk}}(N_{\mathrm{l}}, N_{\mathrm{e}}, N_{\mathrm{ep}}, N_{\mathrm{epk}})$. In this case, there exist exactly $(N_{\mathrm{lp}} + N_{\mathrm{epk}})$ positive instances among selected k nearest training instances. That is, in accordance with the value of $(N_{\mathrm{lp}} + N_{\mathrm{epk}})$, k-NN classifies $\tau(x, y)$ as follows:

- When $N_{\mathrm{lp}} + N_{\mathrm{epk}} > k/2$, k-NN always classifies $\tau(x, y)$ as positive.
- When $N_{\mathrm{lp}} + N_{\mathrm{epk}} < k/2$, k-NN always classifies $\tau(x, y)$ as negative.
- When $N_{\mathrm{lp}} + N_{\mathrm{epk}} = k/2$, k-NN classifies $\tau(x, y)$ as positive with the probability of $\frac{1}{2}$.

Note that the third condition never holds for any odd number of k. Hence, we can obtain $P_{\mathrm{ksp}}(N_{\mathrm{l}}, N_{\mathrm{e}}, N_{\mathrm{lp}}, N_{\mathrm{ep}})$ as

$$P_{\mathrm{ksp}}(N_{\mathrm{l}}, N_{\mathrm{e}}, N_{\mathrm{lp}}, N_{\mathrm{ep}}) = \sum_{N_{\mathrm{epk}}=\lceil k+1/2 \rceil - N_{\mathrm{lp}}}^{N_{\mathrm{ep}}} P_{\mathrm{epk}}(N_{\mathrm{l}}, N_{\mathrm{e}}, N_{\mathrm{ep}}, N_{\mathrm{epk}}) + \frac{1}{2} P_{\mathrm{epk}}\left(N_{\mathrm{l}}, N_{\mathrm{e}}, N_{\mathrm{ep}}, \frac{k}{2} - N_{\mathrm{lp}}\right), \tag{22}$$

where the second term is always 0 for any odd number of k. When k-NN selects exactly $(k - N_{\mathrm{l}})$ out of N_{e} training instances with the distance d from $\tau(x, y)$, these $(k - N_{\mathrm{l}})$ instances comprise exactly N_{epk} out of N_{ep} instances belonging to the positive class and exactly $(k - N_{\mathrm{l}} - N_{\mathrm{epk}})$ out of $(N_{\mathrm{e}} - N_{\mathrm{ep}})$ instances belonging to the negative.

That is, using the hypergeometric probability, we can represent $P_{\mathrm{epk}}(N_{\mathrm{l}}, N_{\mathrm{e}}, N_{\mathrm{ep}}, N_{\mathrm{epk}})$ as

$$P_{\mathrm{epk}}(N_{\mathrm{l}}, N_{\mathrm{e}}, N_{\mathrm{ep}}, N_{\mathrm{epk}}) = \begin{cases} 0 & \text{if } N_{\mathrm{epk}} \text{ is not integer,} \\ H(N_{\mathrm{epk}}; N_{\mathrm{e}}, N_{\mathrm{ep}}, k - N_{\mathrm{l}}) & \text{otherwise.} \end{cases} \quad (23)$$

3.2. Predicted behavior

In the previous subsection, we gave a formal description of k-NNs behavior in the noise-free domain as the accuracy function of the domain characteristics, but the implications of our analysis are not obvious. However, we can use the analysis to make average-case predictions about k-NNs accuracy and the optimal value of k under different domain characteristics. In this subsection, we explore the behavioral implications of the analysis by presenting the effects of domain characteristics on k-NN, such as the effects of the values of p and q on the accuracy, the effect of the parameter k on the accuracy, learning curves of 1-NN and k-NN with the optimal value of k, the storage requirement to achieve a certain accuracy against the number of irrelevant attributes, and the optimal value of k against the size of the training set.

3.2.1. Effect of occurrence probability for each attribute

First, we explore the effect of the occurrence probability for each attribute on the expected accuracy of 1-NN.

Fig. 1(a) shows the effect of the occurrence probability p for relevant attributes on 1-NNs accuracy. In this study, we used $q=\frac{1}{2}$ as the probability for irrelevant attributes and $l=3$ as the number of irrelevant attributes, but we varied the number of training instances N, the threshold value m, the number of relevant attributes n, where we hold $m=(n+1)/2$. For each N and each concept, 1-NN exhibits the worst performance when $p=\frac{1}{2}$, and its accuracy rapidly increases as p is far apart from the value of $\frac{1}{2}$.

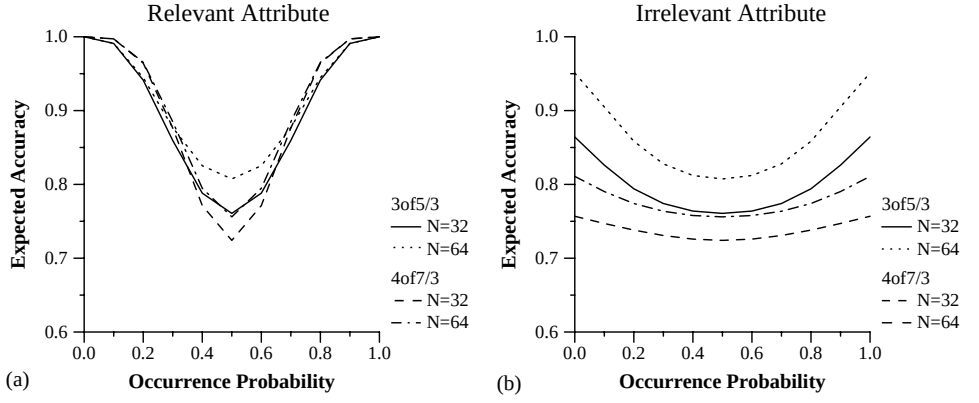

Fig. 1. The effects on the expected accuracy of 1-NN of (a) the occurrence probability p for relevant attributes and (b) the probability q for irrelevant attributes.

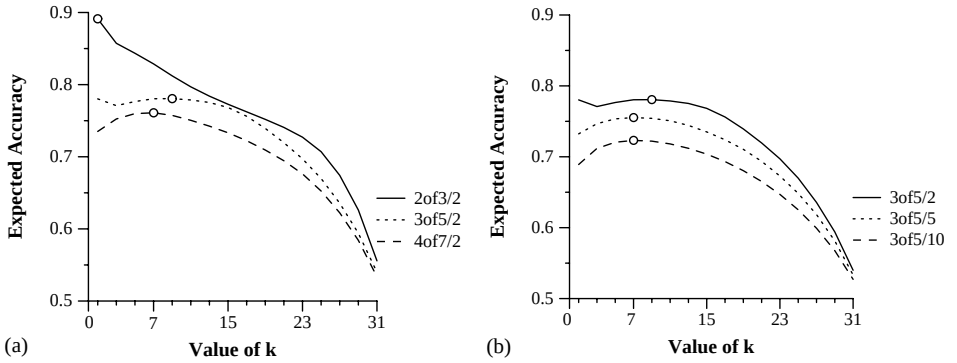

Fig. 2. The effects of the value of k on k-NNs accuracy (a) for the domain with two irrelevant attributes and with varying the threshold value and the number of relevant attributes, and (b) when the domain involves 3-of-5 target concepts with several numbers of irrelevant attributes. Each circle indicates the highest accuracy for the corresponding concept.

Especially, when $p=0$ and $p=1$, 1-NNs accuracy is perfect. This is because all instances drawn from the instance space belong to the negative class if $p=0$ and to positive if $p=1$. Thus, the probability p strongly affects the appearance probabilities for negative and positive instances, and the expected accuracy of 1-NN is very sensitive to the value of p.

Fig. 1(b) presents the corresponding effect of the probability q for irrelevant attribute. Here, we used $p=\frac{1}{2}$ as the probability for relevant attributes. As before, we varied m, n, and N. Although the sensitivity of q to 1-NNs accuracy is less than that for p, the accuracy again gradually increases as q is far apart from $\frac{1}{2}$ for each setting. This is because the probability q does not affect the appearance probability of positive and negative instances, but the effect of irrelevant attributes on 1-NNs accuracy becomes less with an increase or a decrease of q from $\frac{1}{2}$. Especially, when $q=0$ or 1, 1-NNs accuracy is the same as that without irrelevant attributes.

3.2.2. *Effect of the parameter k*

Next, we analyze the effect of the parameter k on the expected accuracy of k-NN.

Fig. 2(a) shows k-NNs accuracy as a function of the odd value of k for several target concepts with two irrelevant attributes. We used $N=32$ as the number of training instances and $p=q=\frac{1}{2}$ as the probability for each attribute, but varied both the threshold m and the number of relevant attributes n, where we have $m=(n+1)/2$. Each circle indicates the optimal value of k for the corresponding target concept. For a 2-of-3/2 concept, the optimal value of k is 1, and k-NNs accuracy gradually decreases with an increase in the value of k. For a 3-of-5/2 concept, k-NNs performance exhibits two peaks at $k=1$ and the optimal $k=9$. For a 4-of-7/2 concept, the expected accuracy increases with an increase in k, then reaches a maximum before starting to deteriorate. For each concept, the expected accuracy of k-NN markedly decreases with an increase in the value of k after the optimal k. Especially, the classification performance of k-NN is quit poor when the number of k is closed to the number of training instances.

Fig. 2(b) presents the corresponding effect for 3-of-5 target concepts with several numbers of irrelevant attributes. Again, we used $N=32$ and $p=q=\frac{1}{2}$, and each circle denotes the optimal k. However, here we varied the number of irrelevant attributes l. While two peaks appear at $k=1$ and the optimal $k=9$ when $l=2$, the peak at $k=1$ disappears with an increase in the number of irrelevant attributes. When $l=5$ and 10, the accuracy improves as the value of k increases, then reaches a maximum before starting to deteriorate. As before, k-NNs accuracy markedly decreases with an increase in k after the optimum.

As can be seen in both Fig. 2(a) and (b), when k is closed to the size of the training set, the classification performance of k-NN is quite poor. Especially when the value of k equals to the number of training instances, we can express the expected accuracy of k-NN as follows.

When $k=N$, $P_{\mathrm{pos}}(x,y)$ in given Eq. (7) can be represented as

$$P_{\mathrm{pos}}(x,y)=\sum_{w=\lceil N+1/2\rceil}^{N} B(w;N,P^{\mathrm{p}})+\frac{1}{2}\binom{N}{\frac{N}{2}}P^{\mathrm{p}^{N/2}}(1-P^{\mathrm{p}})^{N/2}, \tag{24}$$

where P^{p} represents the probability that an arbitrary training instance has the positive class label, and is given by

$$P^{\mathrm{p}}=\sum_{z=m}^{n} B(z;n,p). \tag{25}$$

As shown in Eq. (24), when $k=N$, $P_{\mathrm{pos}}(x,y)$ is independent of the values of x and y. We simply denote this probability by P_{pos}. From Eqs. (24) and (25), the following claim clearly holds.

Claim 4. *When we have $k=N$, the expected accuracy of k-NN given in Eq. (6) can be represented as*

$$\mathscr{A}=(1-P^{\mathrm{p}})(1-P^{\mathrm{pos}})+P^{\mathrm{p}}P^{\mathrm{pos}}. \tag{26}$$

In Fig. 2(a) and (b), we used $m=(n+1)/2$ and $p=\frac{1}{2}$. For this setting, we straightforwardly have the following corollary.

Corollary 5. *When we have $k=N$, $p=\frac{1}{2}$, and $m=(n+1)/2$, the expected accuracy of k-NN given in Eq. (6) is always $\frac{1}{2}$.*

3.2.3. Learning curve

Next, our analysis illustrates the learning curves of 1-NN and k-NN with the optimal value of k to achieve the highest accuracy. The learning curves of the optimal k-NN were obtained by collecting the expected accuracy of k-NN with optimal k for each number of training instances.

Fig. 3(a) shows the effects of relevant attribute on learning rates of 1-NN and the optimal k-NN. In this study, we used $l=2$ as the number of irrelevant attributes and

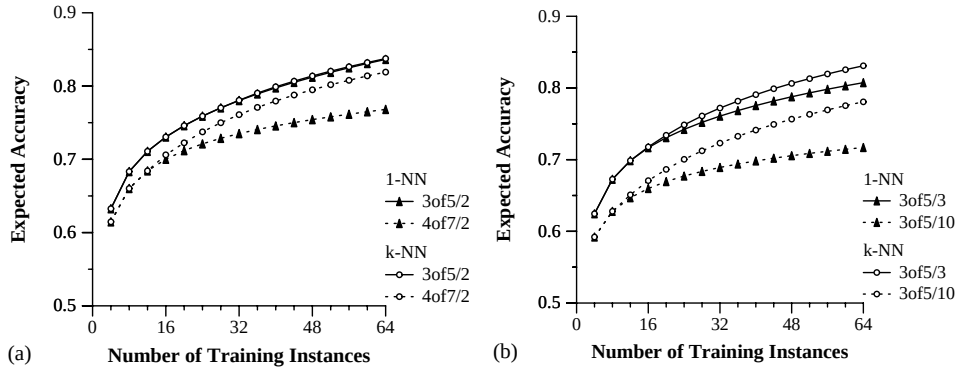

Fig. 3. Learning curves of 1-NN and the optimal k-NN.

$p=q=\frac{1}{2}$ as the probability for each attribute, but varied both the threshold m and the number of relevant attributes n, where we have $m=(n+1)/2$. As typical with learning curves, the accuracies begin low and gradually improve with the size of the training set. Also, each accuracy of 1-NN and the optimal k-NN for the 4-of-7/2 concept is lower than that for the 3-of-5/2 concept for each number of training instances. However, the optimal k-NN exhibits almost the same learning rate for each concept. That is, the optimal k-NNs rate of learning is mostly not affected by relevant attributes. On the other hand, 1-NNs learning rate is sensitive to the number of relevant attributes.

Fig. 3(b) shows the corresponding effect of irrelevant attributes on learning rates. Again, we used $p=q=\frac{1}{2}$ as the probability for each attribute. Here we used 3-of-5 concepts, but varied the number of irrelevant attributes. As before, the accuracies begin low and gradually improve with an increase in the size of the training set, and both accuracies of 1-NN and the optimal k-NN drop off when $l=10$ for each number of training instances. Also, the optimal k-NNs rate of learning is mostly not affected by irrelevant attributes, whereas 1-NNs rate is very sensitive to irrelevant attributes.

3.2.4. Storage requirement

Our analysis further explores the effect of irrelevant attributes on k-NN. In this exploration, we represent the theoretical number of training instances required to achieve a certain level of accuracy as a function of the number of irrelevant attributes.

Fig. 4 shows the storage requirement to achieve 85% and 90% accuracies for 1-NN and the optimal k-NN for 2-of-3 target concepts. This study used $p=q=\frac{1}{2}$ as the probability for each attribute, but varied the number of irrelevant attributes. From Fig. 4, we can observe that the required number of training instances for 1-NN exponentially increases with an increase in the number of irrelevant attributes for 2-of-3 target concepts. That is, the learning behavior of 1-NN is strongly sensitive to the number of irrelevant attributes. On the other hand, the storage requirement for the optimal k-NN is almost linear of the number of irrelevant attributes. Thus, by optimizing the value of k, we can dramatically restrain an increase in the required number of training instances caused by increasing the number of irrelevant attributes. Especially for the domain with

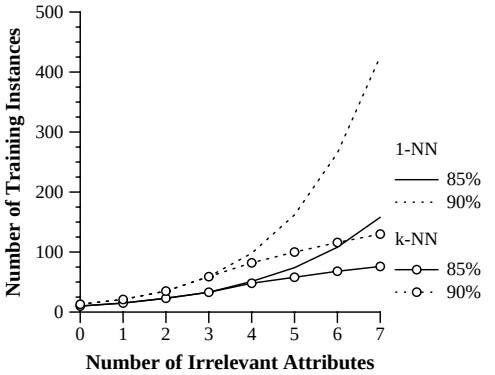

Fig. 4. The number of training instances required to achieve a certain level of accuracy for 1-NN and the optimal k-NN for 2-of-3 concepts as a function of the number of irrelevant attributes.

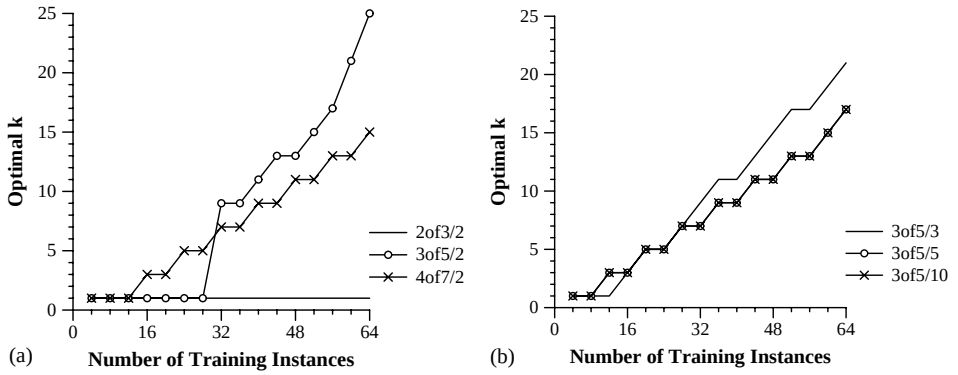

Fig. 5. The optimal value of k to achieve the highest accuracy as a function of the size of the training set.

a large number of irrelevant attributes, optimizing k is very important to achieve good performance of k-NN.

3.2.5. Optimal value of k

Finally, we investigate the optimal value of k as a function of the number of training instances. In this study, we used $p=q=\frac{1}{2}$ as the probability for each attribute.

Fig. 5(a) shows the optimal value of k against the number of training instances for several numbers of relevant attributes. We used $l=2$ as the number of irrelevant attributes, but varied both the threshold m and the number of relevant attributes n, where we have $m=(n+1)/2$. For a 2-of-3/2 concept, the optimal value of k remains steady at 1 with an increase in the number of training instances. However, for both 3-of-5/2 and 4-of-7/2 concepts, the optimal value of k grows almost linearly with an increase in the size of training set, after the optimal k leaves from $k=1$. For a large

number of relevant attributes, the optimal value of k strongly depends on the number of training instances.

Fig. 5(b) illustrates the corresponding optimal k for different numbers of irrelevant attributes. We used 3-of-5 concepts, while varied the number of irrelevant attributes. For each number of irrelevant attributes, the optimal value of k almost linearly increases with an increase in the number of training instances. That is, the optimal value of k is strongly sensitive to the size of training set regardless of the number of irrelevant attributes. Moreover, the change in the optimal value of k is almost the same for each number of irrelevant attributes. Especially, for 5 and 10 irrelevant attributes, the optimal values of k are entirely the same at each size of the training set. Thus, the number of irrelevant attributes does not significantly affect the optimal value of k.

4. Analysis in a noisy domain

In this section, we extend the analysis given in the previous section to handle noise, and present an average-case analysis of the k-NN classifier in a noisy domain. This study deals with three types of noise: relevant attribute noise, irrelevant attribute noise, and class noise.

First, our analysis formally represents the expected classification accuracy of k-NN as a function of the domain characteristics, including noise rate for each type of noise, given in Table 1. However, to avoid complicated notation, we do *not* explicitly express these characteristics as the arguments of the accuracy function. Our analysis expresses three sorts of expected accuracy of k-NN according to the way that noise affects the instances. One is the expected accuracy of k-NN when each type of noise affects only training instances but not test instances, another is when noise affects only test instances but not test instances, and the last is when noise affects both test and training instances.

Then, our analysis investigates the behavioral implications of the analysis by presenting the effects of each type of noise on the expected accuracy of k-NN and on the optimal value of k.

4.1. Expected accuracy

In this subsection, our analysis represents the expected accuracy of k-NN for m-of-n/l target concepts in the noisy domain after k-NN receives N training instances.

In the same way to the analysis in the noise-free domain given in Section 3.1, we use $\Psi(x, y)$ which is a set of noise-free instances in which x relevant attributes and y irrelevant attributes simultaneously occur. Also, let $\Psi'(x', y')$ be a set of noisy instances in which x' relevant attributes and y' irrelevant attributes simultaneously occur after the effects of all types of noise. To compute the expected accuracy of k-NN in the noisy domain, our analysis begins with representing the following probabilities according to the occurrence probability for instances after the effect of noise.

- $P'_{\mathrm{occ}}(x', y')$: the probability that an arbitrary instance drawn from the instance space belongs to $\Psi'(x', y')$.

- $P'_{p'}(x', y')$: the probability that an arbitrary instance is in $\Psi'(x', y')$ and has the positive class label.
- $P'_{n'}(x', y')$: the probability that an arbitrary instance is in $\Psi'(x', y')$ and has the negative class label.

First, we compute $P'_{occ}(x', y')$ by considering the effects of each type of attribute noise individually.

For relevant attribute noise, let $P_{nr}(x, x')$ be the probability that the number of relevant attributes with the value of 1 in any instance in $\Psi(x, y)$ is changed from x to x' by the effect of relevant attribute noise. To compute this probability, our analysis considers relevant attributes corrupted from 1 to 0 and from 0 to 1 by relevant attribute noise. Let s be the number of corrupted relevant attributes from 1 to 0, where $\max(0, x - x') \leqslant s \leqslant \min(x, n - x')$. In this case, the number of corrupted relevant attributes from 0 to 1 is always $x' - x + s$. Then, $P_{nr}(x, x')$ can be obtained by summing over all the possible numbers of s, in each case multiplying the probability that exactly s out of x relevant attributes are changed from 1 to 0 and the probability that exactly $x' - x + s$ out of $n - x$ relevant attributes are changed from 0 to 1. That is, using the binomial probability, we can represent $P_{nr}(x, x')$ as

$$P_{nr}(x, x') = \sum_{s=\max(0, x-x')}^{\min(x, n-x')} B(s; x, \sigma_r) B(x' - x + s; n - x, \sigma_r). \tag{27}$$

In a similar way to relevant attribute noise, we compute the probability that the number of irrelevant attributes with the value of 1 in any instance in $\Psi(x, y)$ is changed from y to y' by irrelevant attribute noise. We denote this probability by $P_{ni}(y, y')$, and let t be the number of irrelevant attributes changed from 1 to 0 by irrelevant attribute noise. Then, we can represent $P_{ni}(y, y')$ as

$$P_{ni}(y, y') = \sum_{t=\max(0, y-y')}^{\min(y, l-y')} B(t; y, \sigma_i) B(y' - y + t; l - y, \sigma_i). \tag{28}$$

The probability that an arbitrary noise-free instance belongs to $\Psi(x, y)$ was represented as $P_{occ}(x, y)$ in Eq. (5). Hence, we can obtain $P'_{occ}(x', y')$ by summing the product of $P_{nr}(x, x')$, $P_{ni}(y, y')$, and $P_{occ}(x', y')$ over all the possible numbers of x and y. This is because class noise does not affect the occurrence probability of instances and we assume the independence of each type of noise. That is, we can express $P'_{occ}(x', y')$ as

$$P'_{occ}(x', y') = \sum_{x=0}^{n} \sum_{y=0}^{l} P_{occ}(x, y) P_{nr}(x, x') P_{ni}(y, y'). \tag{29}$$

Let $\Psi'_{an}(x', y')$ be a set of noisy instances in which x' relevant attributes and y' irrelevant attributes simultaneously occur after each type of attribute noise but before class noise. To represent $P'_{p'}(x', y')$ and $P'_{n'}(x', y')$, our analysis computes the occurrence probability for an arbitrary instance in $\Psi'_{an}(x', y')$ with positive and negative class labels. Let $P'_{p}(x', y')$ be the former probability and $P'_{n}(x', y')$ be the latter. From

Eq. (29), these probabilities are straightforwardly given by

$$P'_{\mathrm{p}}(x', y') = \sum_{x=m}^{n} \sum_{y=0}^{l} P_{\mathrm{occ}}(x, y) P_{\mathrm{nr}}(x, x') P_{\mathrm{ni}}(y, y'). \tag{30}$$

$$P'_{\mathrm{n}}(x', y') = \sum_{x=0}^{m-1} \sum_{y=0}^{l} P_{\mathrm{occ}}(x, y) P_{\mathrm{nr}}(x, x') P_{\mathrm{ni}}(y, y'). \tag{31}$$

Using these occurrence probabilities, from the assumption of independence of each type of noise, we can represent $P'_{\mathrm{p}'}(x', y')$ and $P'_{\mathrm{n}'}(x', y')$ as

$$P'_{\mathrm{p}'}(x', y') = (1 - \sigma_{\mathrm{c}}) P'_{\mathrm{p}}(x', y') + \sigma_{\mathrm{c}} P'_{\mathrm{n}}(x', y'), \tag{32}$$

$$P'_{\mathrm{n}'}(x', y') = \sigma_{\mathrm{c}} P'_{\mathrm{p}}(x', y') + (1 - \sigma_{\mathrm{c}}) P'_{\mathrm{n}}(x', y'). \tag{33}$$

At this point, we obtain $P'_{\mathrm{occ}}(x', y')$, $P'_{\mathrm{p}'}(x', y')$ and $P'_{\mathrm{n}'}(x', y')$. Using these probabilities, our analysis represents three sorts of the expected accuracy of k-NN.

First, we compute the expected accuracy when each type of noise affects only training instances but not test instances. In this case, the occurrence probability for test instances in $\Psi(x, y)$ is $P_{\mathrm{occ}}(x, y)$ given in Eq. (5). Hence, after k-NN receives N training instances affected by each type of noise, the expected accuracy of k-NN for noise-free test instances as

$$\mathscr{A}' = \sum_{y=0}^{l} \left\{ \sum_{x=0}^{m-1} P_{\mathrm{occ}}(x, y)(1 - P'_{\mathrm{pos}}(x, y)) + \sum_{x=m}^{n} P_{\mathrm{occ}}(x, y) P'_{\mathrm{pos}}(x, y) \right\}, \tag{34}$$

where $P'_{\mathrm{pos}}(x, y)$ represents the probability that k-NN classifies an arbitrary test instance in $\Psi(x, y)$ as positive when each type of noise affects N training instances.

In the same way that we obtained $P_{\mathrm{pos}}(x, y)$ in Eq. (7), let $\tau(x, y)$ be an arbitrary test instance in $\Psi(x, y)$, and we use the distance d from $\tau(x, y)$ to the kth nearest training instance. Let us consider the case that exactly N_{l} $(0 \leqslant N_{\mathrm{l}} \leqslant k - 1)$ out of N training instances occur with the distance less than d from $\tau(x, y)$, and exactly N_{e} training instances appear with the distance d. In this case, we have $(k - N_{\mathrm{l}}) \leqslant N_{\mathrm{e}} \leqslant (N - N_{\mathrm{l}})$. We use $P'_{\mathrm{num}}(x, y, d, N_{\mathrm{l}}, N_{\mathrm{e}})$ to refer to the probability that this case occurs. We also use $P'_{\mathrm{sp}}(x, y, d, N_{\mathrm{l}}, N_{\mathrm{e}})$ to denote the probability that k-NN classifies $\tau(x, y)$ as positive in this case. By multiplying $P'_{\mathrm{num}}(x, y, d, N_{\mathrm{l}}, N_{\mathrm{e}})$ and $P'_{\mathrm{sp}}(x, y, d, N_{\mathrm{l}}, N_{\mathrm{e}})$ over all possible values of d, N_{l}, and N_{e}, we can represent $P'_{\mathrm{pos}}(x, y)$ as

$$P'_{\mathrm{pos}}(x, y) = \sum_{d=0}^{n+l} \sum_{N_{\mathrm{l}}=0}^{k-1} \sum_{N_{\mathrm{e}}=k-N_{\mathrm{l}}}^{N-N_{\mathrm{l}}} P'_{\mathrm{num}}(x, y, d, N_{\mathrm{l}}, N_{\mathrm{e}}) P'_{\mathrm{sp}}(x, y, d, N_{\mathrm{l}}, N_{\mathrm{e}}). \tag{35}$$

Let $P'_{\mathrm{l}}(x, y, d)$ and $P'_{\mathrm{e}}(x, y, d)$ be the occurrence probability for an arbitrary training instance with the distance less than and equal to d from $\tau(x, y)$ respectively. These

probabilities are given by

$$P'_{\mathrm{l}}(x,y,d)=\sum_{u'=0}^{n}\sum_{v'=0}^{l}P'_{\mathrm{occ}}(u',v')\sum_{e=0}^{d-1}P_{\mathrm{dis}}(x,y,u',v',e), \tag{36}$$

$$P'_{\mathrm{e}}(x,y,d)=\sum_{u'=0}^{n}\sum_{v'=0}^{l}P'_{\mathrm{occ}}(u',v')P_{\mathrm{dis}}(x,y,u',v',d), \tag{37}$$

where $P_{\mathrm{dis}}(x,y,u',v',e)$ is the probability that an arbitrary training instance in $\Psi'(u',v')$ has the distance e from $\tau(x,y)$, and this probability was given in Eq. (15). Then we have

$$P'_{\mathrm{num}}(x,y,d,N_{\mathrm{l}},N_{\mathrm{e}})=T(N_{\mathrm{l}},N_{\mathrm{e}};N,P'_{\mathrm{l}}(x,y,d),P'_{\mathrm{e}}(x,y,d)). \tag{38}$$

Let $P'_{\mathrm{lps}}(x,y,d,N_{\mathrm{l}},N_{\mathrm{lp}})$($P'_{\mathrm{eps}}(x,y,d,N_{\mathrm{e}},N_{\mathrm{ep}})$, resp.) be the probability that, when exactly N_{l} (N_{e}, resp.) training instances affected by noise have the distance less than (equal to, resp.) d from $\tau(x,y)$, exactly N_{lp} (N_{ep}, resp.) out of these N_{l} (N_{e}, resp.) instances have the positive class label. These probabilities can be represented as

$$P'_{\mathrm{lps}}(x,y,d,N_{\mathrm{l}},N_{\mathrm{lp}})=B\left(N_{\mathrm{lp}};N_{\mathrm{l}},\frac{P'_{\mathrm{lp}}(x,y,d)}{P'_{\mathrm{l}}(x,y,d)}\right), \tag{39}$$

$$P'_{\mathrm{eps}}(x,y,d,N_{\mathrm{e}},N_{\mathrm{ep}})=B\left(N_{\mathrm{ep}};N_{\mathrm{e}},\frac{P'_{\mathrm{ep}}(x,y,d)}{P'_{\mathrm{e}}(x,y,d)}\right), \tag{40}$$

where $P'_{\mathrm{lp}}(x,y,d)$ ($P'_{\mathrm{ep}}(x,y,d)$, resp.) is the probability that an arbitrary training instance affected by noise occurs with the distance less than (equal to, resp.) d from $\tau(x,y)$ and has the positive class label. $P'_{\mathrm{lp}}(x,y,d)$ and $P'_{\mathrm{ep}}(x,y,d)$ are given by

$$P'_{\mathrm{lp}}(x,y,d)=\sum_{u'=0}^{n}\sum_{v'=0}^{l}P'_{\mathrm{p}'}(u',v')\sum_{e=0}^{d-1}P_{\mathrm{dis}}(x,y,u',v',e), \tag{41}$$

$$P'_{\mathrm{ep}}(x,y,d)=\sum_{u'=0}^{n}\sum_{v'=0}^{l}P'_{\mathrm{p}'}(u',v')P_{\mathrm{dis}}(x,y,u',v',d). \tag{42}$$

Then, we can represent $P'_{\mathrm{sp}}(x,y,d,N_{\mathrm{l}},N_{\mathrm{e}})$ as

$$P'_{\mathrm{sp}}(x,y,d,N_{\mathrm{l}},N_{\mathrm{e}})=\sum_{N_{\mathrm{lp}}=0}^{N_{\mathrm{l}}}\left(P'_{\mathrm{lps}}(x,y,d,N_{\mathrm{l}},N_{\mathrm{lp}})\right.$$
$$\left.\sum_{N_{\mathrm{ep}}=0}^{N_{\mathrm{e}}}P'_{\mathrm{eps}}(x,y,d,N_{\mathrm{e}},N_{\mathrm{ep}})P_{\mathrm{ksp}}(N_{\mathrm{l}},N_{\mathrm{e}},N_{\mathrm{lp}},N_{\mathrm{ep}})\right), \tag{43}$$

where $P_{\mathrm{ksp}}(N_{\mathrm{l}},N_{\mathrm{e}},N_{\mathrm{lp}},N_{\mathrm{ep}})$ is the probability that, when N_{lp} out of N_{l} training instances with the distance less than d from $\tau(x,y)$ and N_{ep} out of N_{e} instances with the distance d have the positive label, k-NN classifies $\tau(x,y)$ as positive, and this probability was obtained in Eq. (22).

Next, we compute the expected accuracy of k-NN when each noise affects only test instances but not training instances. In this case, the occurrence probability for an arbitrary noisy test instance in $\Psi'(x', y')$ with the positive class label is $P'_{\mathrm{p}'}(x', y')$ given in Eq. (32). Also, that for the negative class label is $P'_{\mathrm{n}'}(x', y')$ given in Eq. (33). Hence, after N noise-free training instances, the expected accuracy of k-NN for noisy test instances can be represented as

$$\mathscr{A}'' = \sum_{x'=0}^{n} \sum_{y'=0}^{l} \{P'_{\mathrm{n}'}(x', y')(1 - P_{\mathrm{pos}}(x', y')) + P'_{\mathrm{p}'}(x', y')P_{\mathrm{pos}}(x', y')\}, \tag{44}$$

where $P_{\mathrm{pos}}(x', y')$ is the probability that k-NN classifies an arbitrary test instance in $\Psi'(x', y')$ as positive for noise-free N training instances, and this probability was obtained in Eq. (7).

Finally, we compute the expected accuracy of k-NN when each noise affects both test and training instances. In this case, the expected accuracy of k-NN can be obtained as

$$\mathscr{A}''' = \sum_{x'=0}^{n} \sum_{y'=0}^{l} \{P'_{\mathrm{n}'}(x', y')(1 - P'_{\mathrm{pos}}(x', y')) + P'_{\mathrm{p}'}(x', y')P'_{\mathrm{pos}}(x', y')\}. \tag{45}$$

4.2. Predicted behavior

In the previous subsection, we gave a formal description of k-NNs behavior as the accuracy function of domain characteristics including the amount of each type of noise, but the implications of this analysis are not obvious. However, using the accuracy function, we can explore the average-case behavior of k-NN in the noisy domain. In this subsection, we explore the implications of the analysis by predicting the effects of each type of noise on k-NN, such as the effect of irrelevant attribute noise, the effects of relevant attribute noise and class noise on k-NNs accuracy against the value of k, k-NNs accuracies as a function of noise level for relevant attribute noise and class noise, and the effects of relevant attribute noise and class noise on the optimal value of k. Unless otherwise stated, our exploration deals with each noise type affecting only training instances but not test instances.

4.2.1. Effect of irrelevant attribute noise

First, our analysis explores the effect of irrelevant attribute noise.

Fig. 6 shows the expected accuracy of 1-NN as a function of the level of irrelevant attribute noise. In this study, we used $p = \frac{1}{2}$ and $q = \frac{1}{3}$ as the probability for each attribute, $N = 64$ as the number of training instances, $n = 5$ as the number of relevant attributes, and $\sigma_{\mathrm{r}} = 0$ and $\sigma_{\mathrm{c}} = 0$ as the levels for relevant attribute noise and class noise. However, we varied the number of irrelevant attributes and the threshold value. For each concept, the expected accuracy of 1-NN decreases little with an increase in the level of irrelevant attribute noise. Thus, 1-NNs accuracy is only slightly affected by irrelevant attribute noise.

Especially for $q = \frac{1}{2}$, the amount of irrelevant attribute noise makes no difference. That is, we can straightforwardly obtain the following claim.

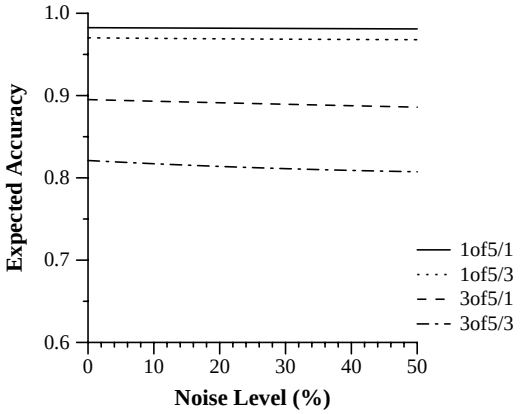

Fig. 6. The effect of irrelevant attribute noise on 1-NNs accuracy, where the number of training instances is fixed at 64.

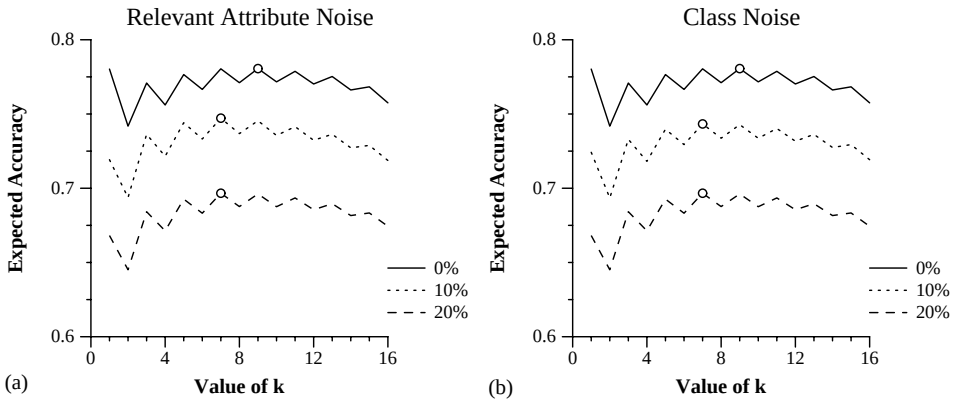

Fig. 7. The expected accuracy of k-NN for a 3-of-5/2 concept as a function of the value of k, for several levels for (a) relevant attribute noise and (b) class noise. Each circle denotes the optimal value of k. The number of training instances is fixed at 32.

Claim 6. *When* $q=\frac{1}{2}$, *each expected accuracy of k-NN given in Eqs.* (34), (44), *and* (45) *has the same for any* σ_i.

4.2.2. Accuracy against value of k

Next, we investigate the predicted behavior of k-NN against the value of k for several levels of relevant attribute noise and class noise.

Fig. 7(a) shows the effects of the value of k for several levels for relevant attribute noise. In this study, we dealt with a 3-of-5/2 target concept, and used $N=32$ as the number of training instances, $p=q=\frac{1}{2}$ as the probability for each attribute, and $\sigma_i=\sigma_c=0$ as the noise rates for irrelevant attribute noise and class noise. Each circle indicates the highest accuracy according to the value of k. The expected accuracy of

k-NN markedly decreases for each value of k with an increase in the level for relevant attribute noise. That is, for the 3-of-5/2 concept, k-NN is strongly sensitive to relevant attribute noise, regardless of the value of k. Also, k-NNs accuracy drops off for each noise level when k is an even number. That is, the expected accuracy of k-NN for an even number of k is lower than both accuracies of $(k-1)$-NN and $(k+1)$-NN for each noise level. This negative influence is crucial especially for a small even number of k.

Fig. 7(b) shows the corresponding effect of the value of k for class noise. Again, this study dealt with a 3-of-5/2 target concept, and used $N=32$, $p=q=\frac{1}{2}$, and $\sigma_r=\sigma_i=0$ as the level for each attribute noise. Each circle expresses the optimal value of k. The behavior of k-NN for class noise is almost the same as that for relevant attribute noise. That is, k-NN is strongly sensitive to class noise for the 3-of-5/2 concept, regardless of the value of k.

4.2.3. *Effect against noise level*

We further investigate the effects of relevant attribute noise and class noise on the expected accuracies of 1-NN and k-NN. Each curve for the optimal k-NN was obtained by collecting the expected accuracy of k-NN with the optimal value of k at each noise level.

Fig. 8(a) shows the effect of relevant attribute noise for 1-of-5/2 and 3-of-5/2 target concepts. We used $N=32$ as the number of training instances, $p=q=\frac{1}{2}$ as the probability for each attribute, and $\sigma_i=\sigma_c=0$ as the noise rate for irrelevant attribute noise and class noise, but varied the noise level for relevant attribute noise. When the noise level is 0%, the accuracy of 1-NN is comparable to that for the optimal k-NN, for both target concepts. However, the expected accuracy of 1-NN decreases almost linearly with an increase in the noise level. In contrast, the expected accuracy of the optimal k-NN exhibits slower degradation. For the 1-of-5/2 concept, the accuracy of the optimal k-NN is *not* greatly changed with the noise level. Moreover, from

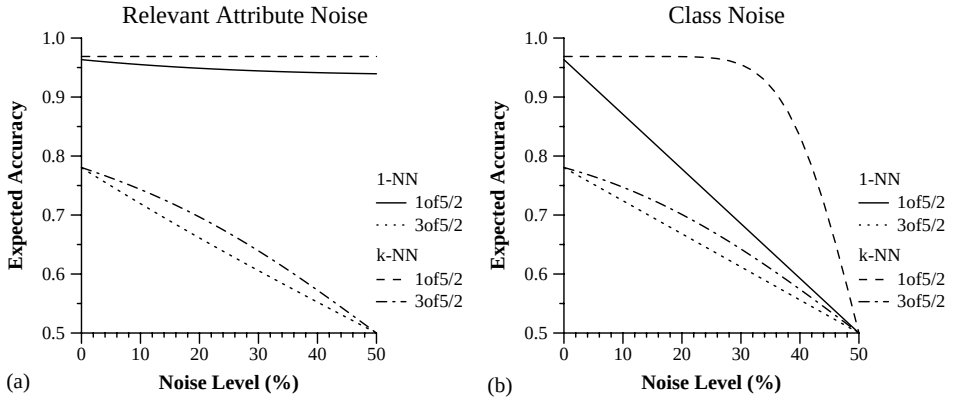

Fig. 8. The effects on the expected accuracies of 1-NN and the optimal k-NN of (a) relevant attribute noise and (b) class noise. The number of training instances is fixed at 32.

Fig. 8(a), we can observe that both accuracies of 1-NN and the optimal k-NN are $\frac{1}{2}$ for the 3-of-5/2 concept when the level for relevant attribute noise is 50%. When we have $p=q=\frac{1}{2}$, $\sigma_r=\frac{1}{2}$, and $\sigma_c=0$, an arbitrary instance affected by relevant attribute noise has the positive class label with the probability of $\frac{1}{2}$ for any m-of-n/l concepts where $m=(n+1)/2$. Hence, this observation can be generalized as the following claim.

Claim 7. *When we have* $\sigma_r=\frac{1}{2}$, $\sigma_c=0$, $p=q=\frac{1}{2}$, *and* $m=(n+1)/2$, *each expected accuracy of k-NN given in Eqs.* (34), (44), *and* (45) *is always* $\frac{1}{2}$.

Fig. 8(b) shows the corresponding effect of class noise. Again, we used $N=32$ and $p=q=\frac{1}{2}$. Here, we used $\sigma_i=\sigma_r=0$ as the noise rate for each attribute noise, but varied the noise level for class noise. 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 effect of class noise on the accuracy differs entirely from one of relevant attribute noise for the 1-of-5/2 concept. The expected accuracy of 1-NN linearly decreases to 0.5. In contrast, the optimal k-NNs accuracy does *not* substantially change until about a 30% noise level, whereafter it rapidly decreases to 50%. Also, Fig. 8(b) shows that both expected accuracies of 1-NN and the optimal k-NN are $\frac{1}{2}$ for each concept when the class noise level is 50%. For a 50% class noise, an arbitrary instance affected by class noise has the positive class label with the probability of $\frac{1}{2}$. That is, we can generalize this observation for any domain characteristics with the exception of σ_c as the following claim.

Claim 8. *When* $\sigma_c=\frac{1}{2}$, *each expected accuracy of k-NN given in Eqs.* (34), (44), *and* (45) *is always* $\frac{1}{2}$.

As can be seen in Fig. 8(b), 1-NNs accuracy decreases linearly with an increase in the class noise level. This observation can be generalized as the following claim.

Claim 9. *Assume class noise affects only training instances but not test instances. When we have* $k=1$ *and* $\sigma_r=\sigma_i=0$, *the expected accuracy*, $\mathscr{A}'$, *given in Eq.* (34) *can be expressed as*

$$\mathscr{A}' = \mathscr{A} + (1 - 2\mathscr{A})\sigma_c, \tag{46}$$

where $\mathscr{A}$ *denotes the corresponding expected accuracy of* 1-*NN for* $\sigma_c=0$.

This claim shows that the expected accuracy of 1-NN changes linearly with the inclination of $(1-2\mathscr{A})$ with an increase in the level for class noise affecting only training instances, when $\sigma_r=\sigma_i=0$.

Moreover, when class noise affects only test instances, this property holds for any value of k. That is, we have the following claim.

Claim 10. *Assume class noise affects only test instances but not training instances. When we have* $\sigma_r=\sigma_i=0$, *the expected accuracy of k-NN*, $\mathscr{A}''$, *given in Eq.* (44) *can*

be represented as

$$\mathscr{A}'' = \mathscr{A} + (1 - 2\mathscr{A})\sigma_c, \tag{47}$$

where $\mathscr{A}$ *denotes the corresponding expected accuracy of k-NN for* $\sigma_c = 0$.

Furthermore, when class noise affects both test and training instances, the effect of class noise on 1-NNs accuracy can be shown as the following claim.

Claim 11. *Assume class noise affects both training and test instances. When we have* $k = 1$ *and* $\sigma_r = \sigma_i = 0$, *the expected accuracy,* $\mathscr{A}'''$, *given in Eq.* (45) *can be expressed as*

$$\mathscr{A}''' = \mathscr{A} - 2\sigma_c(1 - \sigma_c)(2\mathscr{A} - 1), \tag{48}$$

where $\mathscr{A}$ *represents the corresponding expected accuracy of* 1*-NN for* $\sigma_c = 0$.

4.2.4. Optimal value of k

Finally, we explore the relationship between the optimal value of k and the number of training instances in the noisy domain. This study dealt with a 3-of-5/2 target concept.

Fig. 9(a) shows the effect of relevant attribute noise on the optimal value of k as a function of the number of training instances N. We used $p = q = \frac{1}{2}$ as the probability for each attribute, and $\sigma_i = \sigma_c = 0$ as the noise rates for irrelevant attribute noise and class noise, but varied the noise level for relevant attribute noise and 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. That is, the optimal value of k is strongly sensitive to the size of the training set after the optimal k greater than $k = 1$ with an increase in N. Moreover, the predicted behavior about the optimal value of

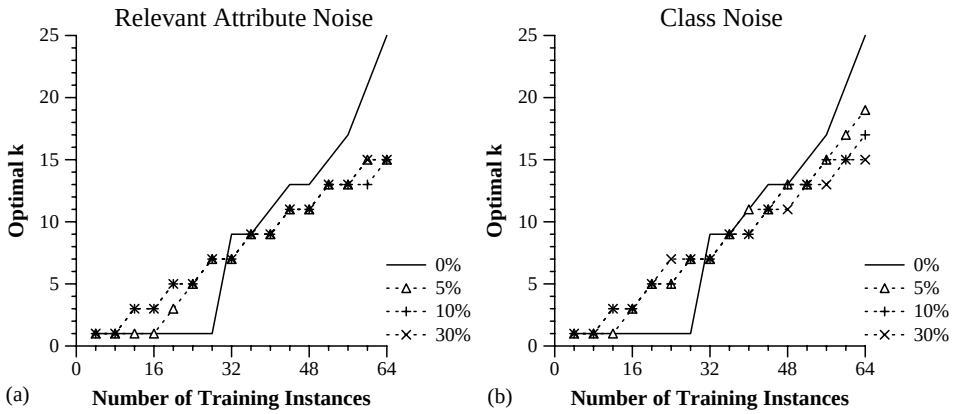

Fig. 9. The optimal value of k as a function of the number of training instances for several noise levels of (a) relevant attribute noise and (b) class noise. The target is a 3-of-5/2 concept.

k is almost the same for 5%, 10%, and 30% levels of noise, regardless of N. Thus, relevant attribute noise does not significantly affect the optimal value of d.

Fig. 9(b) shows the corresponding effect of class noise. We used $p=q=\frac{1}{2}$ and $\sigma_r=\sigma_i=0$, but varied the class noise level and the number of training instances. As before, the optimal value of k almost linearly increases as the number of training instances increases for each level of class noise. That is, the optimal k is strongly sensitive to the size of the training set, especially for a large number of N after the optimal k greater than $k=1$. Also, class noise does not mostly affect the optimal value of k, regardless of the number of training instances.

5. Concluding remarks

In this paper, we have presented average-case analyses of the k-NN classifier (k-NN) employed in most instance-based learning algorithms. As the target concept, we dealt with the m-of-n/l target concept. Our analyses were individually provided for the noise-free domain and for the noisy domain including three types of noise: relevant attribute noise, irrelevant attribute noise, and class noise.

First, in the noise-free domain, we have formally represented the expected accuracy of k-NN as a function of domain characteristics, including the size of training set, the number of relevant and irrelevant attributes, the probability of each attribute, the threshold value, and k. To explore the implications of this analysis, we plotted the predicted behavior of k-NN for artificial domains. The predicted behavior explored involves the effects of each domain characteristic on the expected accuracy of k-NN, the required number of training instances to achieve a certain accuracy, and the optimal value of k against the size of training set.

Next, we extended the analysis to handle three types of noise, and expressed the expected accuracy of k-NN as a function of domain characteristics including the amount of each type of noise. We also investigated the behavioral implications of the analysis by predicting the effects of each type of noise on k-NNs accuracy and on the optimal value of k.

One issue we have not addressed is the tractability of our analysis. Although our analysis presented a formal description of k-NNs behavior as the accuracy function, the function's form was complicated and the calculations needed to predict behavior could take extremely long for large numbers of training instances and attributes. The difficulty resulted from the analyses' reliance on the exact calculation of probabilities for all possible combinations of events. Some researchers pointed out this computational limitation of average-case analysis, and proposed approaches to realize tractable average-case analysis of induction. Golea and Marchand [12] presented an average-case analysis of perceptrons with binary weights by using an approximation of the expected accuracy, and Langley and Sage [18] analyzed a Bayesian classifier using this technique. Also, Reischuk and Zeugmann [27] gave upper and lower bounds on the mind change complexity which is closely related to the number of prediction errors for conjunctive learning algorithms. Moreover, by introducing a new domain characteristic which is the number of instance pairs belonging to the same class with a

certain distance, we gave a simple expression of the expected accuracy of 1-NN, and showed the predicted behavior of 1-NN for large training and attribute sets where empirical approaches such as Monte Carlo simulations are difficult [23]. Following these approaches, we should overcome the computational drawback of the current average-case analyses of k-NN.

Another direction for future research involves using average-case analysis to better understand the behavior of k-Nearest neigbhor and other induction algorithms in natural domains. This would require extending the analysis to handle non-Boolean attributes and a broader range of target concepts. Recently, we presented an average-case analysis of 1-NN for any target concept defined over discrete attribute domains [23]. In the near future, we would like to extend this analysis to k-NN and to apply other induction algorithms.

Acknowledgements

We wish to express our gratitude to Setsuo Arikawa, who supervised the Ph.D. theses of both authors. This paper is based on the first author's Ph.D. thesis, and owes much to his valuable advice and comments.

We are also indebted to Thomas Zeugmann, Ken Satoh, and Pat Langley for constructive comments and helpful suggestions.

Appendix A. Proof of Claim 9

Proof. When we have $k=1$, $P_{\mathrm{sp}}(x,y,d,N_{\mathrm{l}},N_{\mathrm{e}})$ given in Eq. (21) can be represented as

$$\begin{aligned} P_{\mathrm{sp}}(x,y,d,N_{\mathrm{l}},N_{\mathrm{e}}) &= \sum_{N_{\mathrm{ep}}=0}^{N_{\mathrm{e}}} P_{\mathrm{eps}}(x,y,d,N_{\mathrm{e}},P_{\mathrm{ep}})P_{\mathrm{ksp}}(0,N_{\mathrm{e}},0,N_{\mathrm{ep}}) \\ &= \sum_{N_{\mathrm{ep}}=0}^{N_{\mathrm{e}}} B\left(N_{\mathrm{ep}};N_{\mathrm{e}},\frac{P_{\mathrm{ep}}(x,y,d)}{P_{\mathrm{e}}(x,y,d)}\right)\frac{N_{\mathrm{ep}}}{N_{\mathrm{e}}} \\ &= \frac{P_{\mathrm{ep}}(x,y,d)}{P_{\mathrm{e}}(x,y,d)}. \end{aligned} \tag{A.1}$$

In a similar way, $P'_{\mathrm{sp}}(x,y,d,N_{\mathrm{l}},N_{\mathrm{e}})$ given in Eq. (43) can be expressed as

$$P'_{\mathrm{sp}}(x,y,d,N_{\mathrm{l}},N_{\mathrm{e}}) = \frac{P'_{\mathrm{ep}}(x,y,d)}{P'_{\mathrm{e}}(x,y,d)}. \tag{A.2}$$

On the other hand, when $\sigma_{\mathrm{r}}=\sigma_{\mathrm{i}}=0$, $P'_{\mathrm{p}'}(x,y)$ given in Eq. (32) can be represented as

$$P'_{\mathrm{p}'}(x,y) = \begin{cases} \sigma_{\mathrm{c}}P_{\mathrm{occ}}(x,y) & \text{if } 0 \leqslant x < m, \\ (1-\sigma_{\mathrm{c}})P_{\mathrm{occ}}(x,y) & \text{if } m \leqslant x \leqslant n, \end{cases} \tag{A.3}$$

Hence, we can express $P'_{\mathrm{ep}}(x,y,d)$ given in Eq. (42) as

$$\begin{aligned} P'_{\mathrm{ep}}(x,y,d) &= \sigma_{\mathrm{c}} \sum_{u=0}^{m-1} \sum_{v=0}^{l} P_{\mathrm{occ}}(u,v) P_{\mathrm{dis}}(x,y,u,v,d) \\ &\quad + (1-\sigma_{\mathrm{c}}) \sum_{u=m}^{n} \sum_{v=0}^{l} P_{\mathrm{occ}}(u,v) P_{\mathrm{dis}}(x,y,u,v,d) \\ &= \sigma_{\mathrm{c}}(P_{\mathrm{e}}(x,y,d) - P_{\mathrm{ep}}(x,y,d)) + (1-\sigma_{\mathrm{c}}) P_{\mathrm{ep}}(x,y,d) \\ &= \sigma_{\mathrm{c}} P_{\mathrm{e}}(x,y,d) + (1-2\sigma_{\mathrm{c}}) P_{\mathrm{ep}}(x,y,d). \end{aligned} \tag{A.4}$$

Since we have clearly $P'_{\mathrm{e}}(x,y,d) = P_{\mathrm{e}}(x,y,d)$ when $\sigma_{\mathrm{r}} = \sigma_{\mathrm{i}} = 0$, the following equation holds:

$$P'_{\mathrm{sp}}(x,y,d,N_{\mathrm{l}},N_{\mathrm{e}}) = \sigma_{\mathrm{c}} + (1-2\sigma_{\mathrm{c}}) P_{\mathrm{sp}}(x,y,d,N_{\mathrm{l}},N_{\mathrm{e}}). \tag{A.5}$$

Also, for $\sigma_{\mathrm{r}} = \sigma_{\mathrm{i}} = 0$, we straightforwardly have

$$P'_{\mathrm{num}}(x,y,d,N_{\mathrm{l}},N_{\mathrm{e}}) = P_{\mathrm{num}}(x,y,d,N_{\mathrm{l}},N_{\mathrm{e}}). \tag{A.6}$$

From these Eqs. (A.5) and (A.6), when $k=1$ and $\sigma_{\mathrm{r}} = \sigma_{\mathrm{i}} = 0$, we can obtain the following equation:

$$P'_{\mathrm{pos}}(x,y) = \sigma_{\mathrm{c}} + (1-2\sigma_{\mathrm{c}}) P_{\mathrm{pos}}(x,y). \tag{A.7}$$

Hence, we have the desired equation:

$$\begin{aligned} \mathscr{A}' &= \sum_{y=0}^{l} \left[\sum_{x=0}^{m-1} P_{\mathrm{occ}}(x,y)\{1 - (\sigma_{\mathrm{c}} + (1-2\sigma_{\mathrm{c}}) P_{\mathrm{pos}}(x,y))\} \right. \\ &\quad \left. + \sum_{x=m}^{n} P_{\mathrm{occ}}(x,y)(\sigma_{\mathrm{c}} + (1-2\sigma_{\mathrm{c}}) P_{\mathrm{pos}}(x,y)) \right] \\ &= \mathscr{A} + (1-2\mathscr{A})\sigma_{\mathrm{c}}. \qquad \square \end{aligned} \tag{A.8}$$

Appendix B. Proof of Claim 10

Proof. When $\sigma_{\mathrm{r}} = \sigma_{\mathrm{i}} = 0$, Eq. (A.3) holds and we can express $P'_{\mathrm{n}'}(x,y)$ given in Eq. (33) as

$$P'_{\mathrm{n}'}(x,y) = \begin{cases} (1-\sigma_{\mathrm{c}}) P_{\mathrm{occ}}(x,y) & \text{if } 0 \leqslant x < m, \\ \sigma_{\mathrm{c}} P_{\mathrm{occ}}(x,y) & \text{if } m \leqslant x \leqslant n. \end{cases} \tag{B.1}$$

Hence, the expected accuracy of k-NN given in Eq. (44) can be rewritten by

$$\begin{aligned}\mathscr{A}'' = \sum_{y=0}^{l} \Bigg[\sum_{x=0}^{m-1} \{(1-\sigma_c)P_{occ}(x,y)(1-P_{pos}(x,y)) + \sigma_c P_{occ}(x,y)P_{pos}(x,y)\} \\ + \sum_{x=m}^{n} \{\sigma_c P_{occ}(x,y)(1-P_{pos}(x,y)) + (1-\sigma_c)P_{occ}(x,y)P_{pos}(x,y)\} \Bigg] \\ = \mathscr{A} + (1-2\mathscr{A})\sigma_c. \qquad \square \end{aligned} \tag{B.2}$$

Appendix C. Proof of Claim 11

Proof. When $k=1$ and $\sigma_r=\sigma_i=0$, we have Eqs. (A.3), (B.1), and (A.7). Hence, the expected accuracy of 1-NN given in Eq. (45) can be rewritten by

$$\begin{aligned}\mathscr{A}''' = \sum_{x=0}^{m-1} \sum_{y=0}^{l} \Big[(1-\sigma_c)P_{occ}(x,y)\{1-(\sigma_c+(1-2\sigma_c)P_{pos}(x,y))\} \\ + \sigma_c P_{occ}(x,y)(1-2\sigma_c)P_{pos}(x,y) \Big] \\ + \sum_{x=m}^{n} \sum_{y=0}^{l} \Big[\sigma_c P_{occ}(x,y)\{1-(\sigma_c+(1-2\sigma_c)P_{pos}(x,y))\} \\ + (1-\sigma_c)P_{occ}(x,y)(1-2\sigma_c)P_{pos}(x,y) \Big] \\ = \mathscr{A} + 2\sigma_c(1-\sigma_c)(1-2\mathscr{A}). \qquad \square \end{aligned} \tag{C.1}$$

References

[1] D. Aha, D. Kibler, Noise-tolerant instance-based learning algorithms, in: Proc. 11th Internat. Joint Conf. on Artificial Intelligence, Morgan Kaufmann, San Francisco, CA, 1989, pp. 794–799.
[2] D. Aha, D. Kibler, M. Albert, Instance-based learning algorithms, Mach. Learning 6 (1991) 37–66.
[3] M. Albert, D. Aha, Analyses of instance-based learning algorithms, in: Proc. 9th National Conf. on Artificial Intelligence, AAAI Press, Menlo Park, CA, 1991, pp. 553–558.
[4] T. Bailey, A. Jain, A note on distance-weighted k-nearest neighbor rules, IEEE Trans. Systems Man Cybernet. 8 (4) (1978) 311–313.
[5] T. Cover, Estimation by the nearest neighbor rule, IEEE Trans. Inform. Theory 14 (1) (1968) 50–55.
[6] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Trans. Inform. Theory 13 (1) (1967) 21–27.
[7] O. Creecy, B. Masand, S. Smith, D. Waltz, Trading MIPS and memory for knowledge engineering, Comm. Assoc. Comput. Mach. 35 (1992) 48–63.
[8] B. Dasarathy (Ed.), Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques, IEEE Press, Los Altos, CA, 1991.

[9] R. Duda, P. Hart (Eds.), Pattern Classification and Scene Analysis, Wiley, New York, 1973.
[10] S. Dudani, The distance-weighted k-nearest-neighbor rule, IEEE Trans. Systems Man Cybernet. 6 (4) (1976) 325–327.
[11] E. Fix, J. Hodges, Discriminatory analysis: nonparametric discrimination: consistency properties, Technical Report 4, USAF School of Aviation Medicine, Project 21-49-004, 1951.
[12] M. Golea, M. Marchand, On learning perceptrons with binary weights, Neural Comput. 5 (1993) 767–782.
[13] D. Haussler, Probably approximately correct learning, in: Proc. 8th National Conf. on Artificial Intelligence, AAAI Press. Menlo Park, CA, 1990, pp. 1101–1108.
[14] D. Hirschberg, M. Pazzani, Average-case analysis of learning k-CNF concept, in: Proc. 9th Internat. Conf. on Machine Learning, Morgan Kaufmann, San Francisco, CA, 1992, pp. 206–211.
[15] W. Iba, P. Langley, Induction of one-level decision trees, in: Proc. 9th Internat. Conf. on Machine Learning, Morgan Kaufmann, San Francisco, CA, 1992, pp. 233–240.
[16] P. Langley, W. Iba, Average-case analysis of a nearest neighbor algorithm, in: Proc. 13th Internat. Joint Conf. on Artificial Intelligence, Morgan Kaufmann, San Francisco, CA, 1993, pp. 889–894.
[17] P. Langley, W. Iba, K. Thompson, An analysis of bayesian classifiers, in: Proc. 10th National Conf. on Artificial Intelligence, AAAI Press, Menlo Park, CA, 1992, pp. 223–228.
[18] P. Langley, S. Sage, Tractable average-case analysis of naive bayesian classifiers, in: Proc. 16th Internat. Conf. on Machine Learning, Morgan Kaufmann, San Francisco, CA, 1999, pp. 220–228.
[19] P. Murphy, M. Pazzani, ID2-*of*-3: constructive induction of *m-of-n* concepts for discriminators in decision trees, in: Proc. 8th Internat. Workshop on Machine Learning, Morgan Kaufmann, Los Altos, CA, 1991, pp. 183–187.
[20] S. Okamoto, K. Satoh, An average-case analysis of k-nearest neighbor classifier, in: Proc. 1st Internat. Conf. on Case-Based Reasoning, Lecture Notes in Artificial Intelligence, Vol. 1010, Springer, Berlin, 1995, pp. 253–264.
[21] S. Okamoto, N. Yugami, Theoretical analysis of the nearest neighbor classifier in noisy domains, in: Proc. 13th Internat. Conf. on Machine Learning, Morgan Kaufmann, San Francisco, CA, 1996, pp. 355–363.
[22] S. Okamoto, N. Yugami, An average-case analysis of the k-nearest neighbor classifier for noisy domains, in: Proc. 15th Internat. Joint Conf. on Artificial Intelligence, Morgan Kaufmann, San Francisco, CA, 1997, pp. 238–243.
[23] S. Okamoto, N. Yugami, Generalized average-case analyses of the nearest neighbor algorithm, in: Proc. 17th Internat. Conf. on Machine Learning, Morgan Kaufmann, San Francisco, CA, 2000, pp. 695–702.
[24] M. Pazzani, W. Sarrett, A framework for average case analysis of conjunctive learning algorithms, Mach. Learning 9 (1992) 349–372.
[25] L. Pitt, L. Valiant, Computational limitations on learning from examples, J. Assoc. Comput. Mach. 35 (4) (1988) 965–984.
[26] J. Rachlin, S. Kasif, S. Salzberg, D. Aha, Toward a better understanding of memory-based reasoning systems, in: Proc. 11th Internat. Conf. on Machine Learning, Morgan Kaufmann, Los Altos, CA, 1994, pp. 242–250.
[27] R. Reischuk, T. Zeugmann, A complete and tight average-case analysis of learning monomials, in: Proc. 16th Internat. Symp. on Theoretical Aspects of Computer Science, Lecture Notes in Computer Science, Vol. 1563, Springer, Berlin, 1999, pp. 414–423.
[28] S. Salzberg, A. Delcher, D. Heath, S. Kasif, Learning with a helpful teacher, in: Proc. 12th Internat. Joint Conf. on Artificial Intelligence, Morgan Kaufmann, San Francisco, CA, 1991, pp. 705–711.
[29] S. Salzberg, A. Delcher, D. Heath, S. Kasif, Best-case results for nearest neighbor learning, IEEE Trans. Pattern Anal. Mach. Intell. 17 (6) (1995) 599–610.
[30] C. Stanfill, D. Waltz, Toward memory-based reasoning, Comm. Assoc. Comput. Mach. 29 (12) (1986) 1213–1228.
[31] L. Valiant, A theory of the learnable, Comm. Assoc. Comput. Mach. 27 (1984) 1134–1142.
[32] D. Waltz, Massively parallel AI, in: Proc. 8th National Conf. on Artificial Intelligence, AAAI Press, Menlo Park, CA, 1990, pp. 1117–1122.
[33] D. Wettschereck, D. Aha, T. Mohri, A review and empirical comparison of feature weighting methods for a class of lazy learning algorithms, Artif. Intell. Rev. 11 (1997) 273–314.