



# A New Nearest Neighbor Classifier via Fusing Neighborhood Information

저널 : [Neurocomputing](#), [Volume 143](#), 2 November 2014, Pages 164-169

저자 : [YaojinLin](#), [JinjinLi](#), [MengleiLin](#), [JinkunChen](#)

발표자

18510049 오순묵

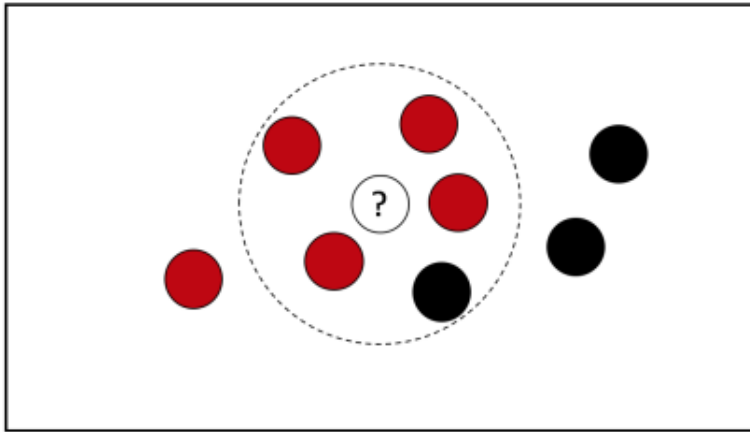
18512084 이도현

18512112 한동기

# 목차

1. 기존 방법론 ( $k$ -NN)
2. 관련 연구
3. 기존 방법론의 장점과 단점
4. 개선 아이디어
5. 실험 디자인
6. 결론
7. 구현

# 1. 기존 방법론 ( $k$ -NN)



- $k$ -NN( $k$ -Nearest Neighbors)
- $k$ 개의 최근접 이웃 데이터를 찾고 다수결 방식으로 분류하는 알고리즘
- 거리함수- Test Sample의  $k$ 개의 이웃을 판별하기 위해 Test Sample과 Train Sample들의 유사도 측정 함수(Similarity function)
- Single Metric for Distance Measure

## 2. 관련 연구 (Distance Metrics)

- Global & Local Distance Metric

Global Distance Metric	Local Distance Metric
<ul style="list-style-type: none"><li>• 모든 데이터 쌍에 대해 일괄적으로 동일한 규칙을 적용하는 distance metric</li><li>• 대표적인 예로 Euclidean Distance</li><li>• HEOM, VDM, HVDM, IVDM 등</li></ul>	<ul style="list-style-type: none"><li>• 지엽적인 데이터 쌍 마다 서로 다른 규칙을 적용하는 distance metric</li><li>• ADEMENN, WA <math>k</math>NN, I <math>k</math>NN 등</li></ul>

## 2. 관련 연구 (Distance Metrics)

- Global Distance Metrics

**HEOM**

$$HEOM(x, y) = \sqrt{\sum_{a=1}^m d_a(x_a, y_a)^2}$$

$$d_a(x, y) = \begin{cases} 1 & \text{if } x \text{ or } y \text{ is unknown} \\ overlap(x, y) & \text{if } a \text{ is nominal} \\ diff_a(x, y) & \text{if } a \text{ is interval} \end{cases}$$

- Interval, Nominal feature를 모두 가지는 데이터에 대한 distance metric
- Interval feature의 경우 정규화 과정을 포함하는 Euclidean distance metric
- Nominal feature의 경우 같은 값이면 1, 다른 값이면 0

**VDM**

$$\begin{aligned} VDM_a(x, y) &= \sum_{c=1}^C \left| \frac{N_{a,x,c}}{N_{a,x}} - \frac{N_{a,y,c}}{N_{a,y}} \right|^q \\ &= \sum_{c=1}^C |P_{a,x,c} - P_{a,y,c}|^q \end{aligned}$$

- Nominal feature에 대한 distance metric
- Feature의 class 분포가 비슷할수록 VDM의 값이 작다

## 2. 관련 연구 (Distance Metrics)

- Global Distance Metrics

### HVDM

$$HVDM(x, y) = \sqrt{\sum_{a=1}^m d_a(x_a, y_a)^2}$$

$$d_a(x, y) = \begin{cases} 1 & \text{if } x \text{ or } y \text{ is unknown} \\ vdm_a(x, y) & \text{if } a \text{ is nominal} \\ diff_a(x, y) & \text{if } a \text{ is interval} \end{cases}$$

- HEOM과 VDM을 결합
- Interval feature의 경우 정규화 과정을 포함하는 Euclidean distance metric
- Nominal feature의 경우 정규화 과정을 포함하는 VDM

### IVDM

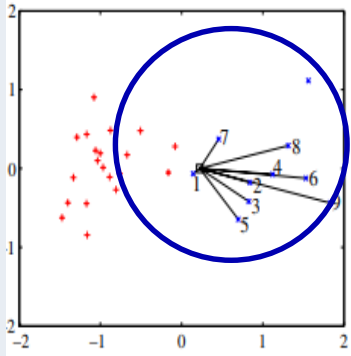
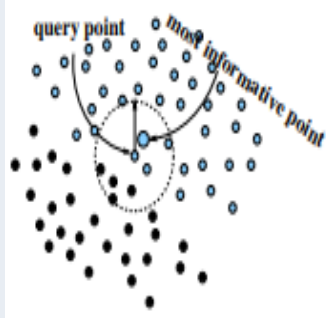
$$IVDM(x, y) = \sum_{a=1}^m d_a(x_a, y_a)^2$$

$$d_a(x, y) = \begin{cases} vdm_a(x, y) & \text{if } a \text{ is nominal} \\ \sum_{c=1}^C |p_{a,c}(x) - p_{a,c}(y)|^2 & \text{if } a \text{ is interval} \end{cases}$$

- VDM의 개념을 Interval feature까지 확장
- Nominal feature의 경우 기존 VDM
- Interval feature의 경우 보간법을 통해 class 분포를 추정하여 적용

## 2. 관련 연구 (Distance Metrics)

- Local Distance Metric

ADAMENN	WA $k$ NN	1 $k$ NN
	$\frac{\sum_{t \in T} (X_t W_t)(Y_t W_t)}{\sqrt{\sum_{t \in T} (X_t W_t)^2} \sqrt{\sum_{t \in T} (Y_t W_t)^2}}$	
<ul style="list-style-type: none"> <li>Training Sample <math>x_i</math>에 대해 반지름 <math>r</math>을 기준으로 원을 그리고, 그 안에 있는 다른 라벨은 제거</li> </ul>	<ul style="list-style-type: none"> <li>각각의 Feature에 대해 Weight를 주고, 이것을 Small Step으로 바꿔 가면서 최적의 Objective Function을 찾는 방법</li> </ul>	<ul style="list-style-type: none"> <li>Different Label과 얼마나 멀리 있는지, Same Label과 얼마나 가까이 있는지를 고려</li> </ul>

### 3. 기존 방법론 장·단점

- 장점

1. 심플하고 효과적인 알고리즘 (원리가 쉽다)
2. 트레이닝 데이터의 수가 늘어나면 정확도 향상
3. 분류 문제와 함수 근사에 탁월

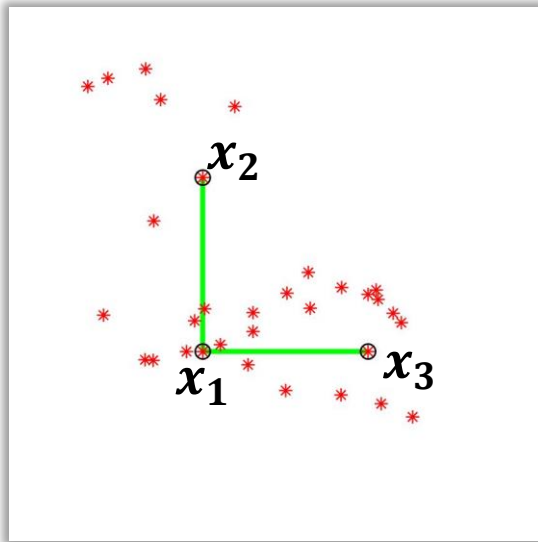
- 단점

1. 거리 함수 의존도가 높다
2. 서로 다른 클래스를 가진 데이터가 뒤섞여 있을 때 분류가 어렵다
3. 하나의 Test Sample을 분류하기 위해 전체 Training 데이터 모두 연산
4. From Single Metric
  - 1) 기본적으로 Single Metric은 모든 데이터의 특성을 반영하는데 한계 존재
  - 2) Single Local Metric은 Noisy Sample에 대해 민감
  - 3) Single Global Metric은 Multimodal Distribution을 반영하지 못함



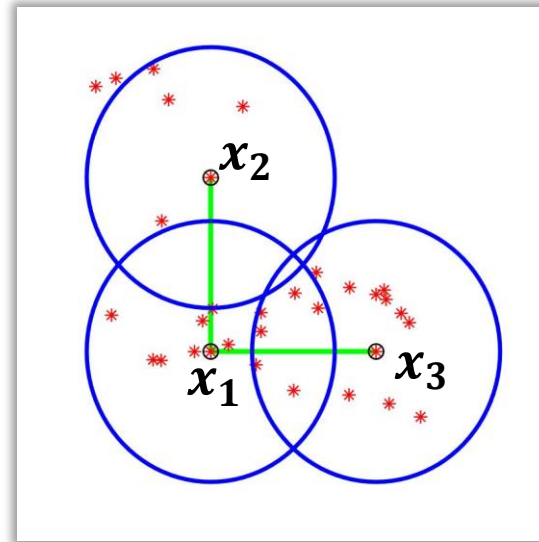
## 4. 개선 아이디어

### (1) Neighborhood Information and It's Influence



Euclidean Distance

$$d(x_1, x_2) = d(x_1, x_3)$$



Distance  
Using Neighborhood Information

$$d(x_1, x_2) \geq d(x_1, x_3)$$

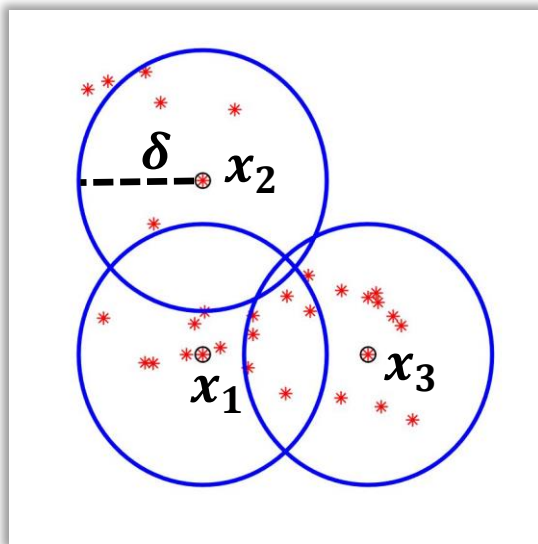


데이터의 neighborhood Information을 활용  
 $k$ -NN 설정시 NN 의 분포가  $d(x_1, x_3)$  방향으로 쏠릴 가능성 높음

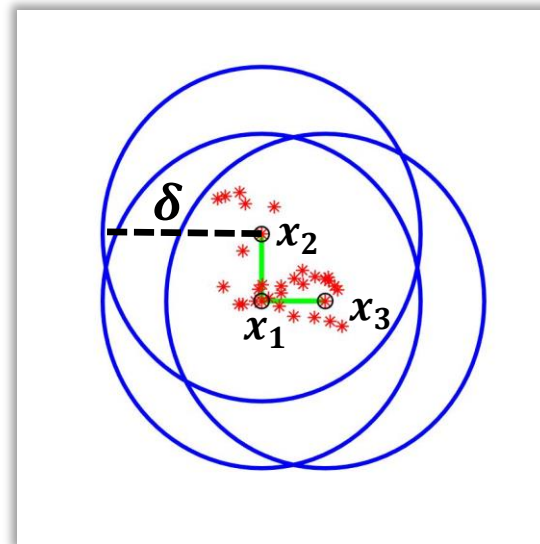
## 4. 개선 아이디어

### (1) Neighborhood Information and It's Influence

Samples with Different Neighborhood



Samples with Similar Neighborhood



$$n_B^\delta(x_i) = \{x_j \mid x_j \in U, \Delta_B(x_i, x_j) \leq \delta\}, \quad \delta \in [0, 1]$$

- 서로 공유하는 Neighborhood가 넓어질수록 Distance Discriminability가 줄어든다
- 적절한  $\delta$  선택 및 multiple metric의 필요성

## 4. 개선 아이디어

### (2) Rank Aggregation

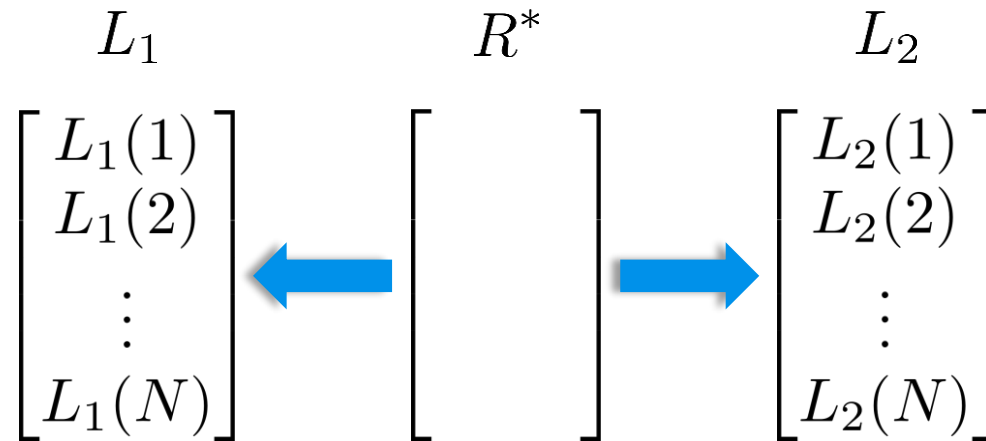
#### Using Multiple Metrics

L1 = List of Euclidian Distance	L2 = List of Distance using Neighborhood Information
$p = 2$ $\Delta_p(x, y) = \left( \sum_{i=1}^N  f(x, a_i) - f(y, a_i) ^p \right)^{\frac{1}{p}}$ $L_1 = \begin{bmatrix} L_1(1) \\ L_1(2) \\ \vdots \\ L_1(N) \end{bmatrix}$	$d(x_i, x_j) = \sum_{l=1}^N \left( \frac{ n_{a_l}^{\delta}(x_i) }{ U } - \frac{ n_{a_l}^{\delta}(x_j) }{ U } \right)^2$ $L_2 = \begin{bmatrix} L_2(1) \\ L_2(2) \\ \vdots \\ L_2(N) \end{bmatrix}$

$L_j(i)$ 는  $j$ 번째 distance metric을 사용했을 때,  
 $i$ 번째로 가까운 데이터의 인덱스를 의미

## 4. 개선 아이디어

### (2) Rank Aggregation



$L_1$ 과  $L_2$  모두를 적절하게 반영하는 Super List를 이용

- Objective Function

$$R^* = \arg \min \sum_{j=1}^2 w_j d(R, L_j)$$

$w_j$ 는  $j$ 번째 distance metric의 가중치

- Spearman Footrule Distance

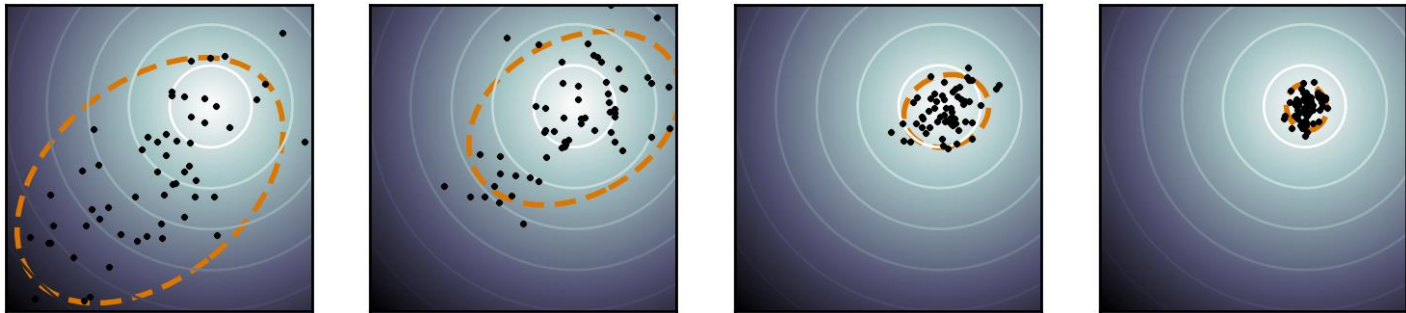
$$d(R, L_j) = \sum_{t \in L_j \cup R} |r^R(t) - r^{L_j}(t)|$$

$r^L(t)$ 는 orderd list  $L$ 의  $t$ 번째 요소 값

## 4. 개선 아이디어

### (2) Rank Aggregation

- cross-entropy Monte Carlo algorithm



1. 임의의 확률 분포에서 변수 샘플링
2. 샘플링 된 변수에 대해 비용 함수 계산
3. 비용 함수 값이 작은 변수들을 가지고 확률 분포 업데이트
4. 1~3 과정을 수렴 할 때까지 반복

## 4. 개선 아이디어

### (3) Algorithm : FN $k$ -NN (Fusing Neighborhood $k$ -Nearest Neighbors)

1. 테스트 샘플  $x$ 와 학습 데이터 셋  $D$ 의 individual distance를 계산하여  $L_1$  생성
2. 테스트 샘플  $x$ 와 학습 데이터 셋  $D$ 의 neighborhood distance를 계산하여  $L_2$  생성
3. cross-entropy Monte Carlo algorithm을 사용하여  $R^*$  생성
4. 학습 데이터 셋  $D$ 에서  $R^*$ 를 기반으로  $k$ 개의 nearest neighbors를 구함
5. 기존  $k$ -NN algorithm을 적용하여 테스트 샘플  $x$ 의 라벨 결정

## 5. 실험 디자인

### (1) Data Set

Dataset	Samples	Features	Classes
Dermatology	366	34	6
Glass	214	9	7
Heart	270	13	2
Hepatitis	155	19	2
ICU	200	20	3
Spam	4601	57	2
Srbct	63	2308	4

❖ From UCI datasets

- Glass, Spam을 제외하고 대부분 질병진단과 관련된 데이터
- 다양한 수의 Sample, Feature, Class
- interval, nominal value 모두 존재
- 기존 연구에서 알고리즘을 평가하는 용도로 자주 사용

# 5. 실험 디자인

## (2) Evaluation Method

- 10-fold cross validation 방식으로 평가
- $\delta = 0.1$  (FNk NN, NEC), linear loss function(LMNN)
- 기존  $k$ -NN과의 비교( $k = 1, 5, 9$ )
- 다양한 classifier와의 비교 ( $k = 10$ )



## 5. 실험 디자인

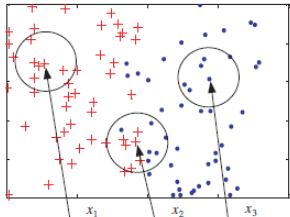
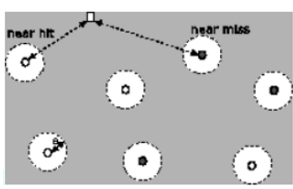
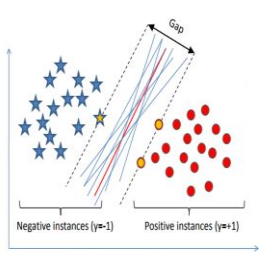
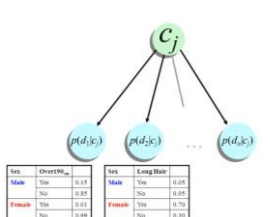
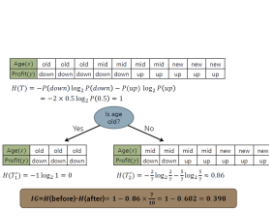
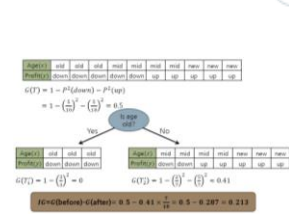
### (3) 기존 $k$ -NN과의 비교

Dataset	$k=1$		$k=5$		$k=9$	
	$k$ NN	FN $k$ NN	$k$ NN	FN $k$ NN	$k$ NN	FN $k$ NN
Dermatology	96.11	<b>96.59</b>	97.14	<b>97.76</b>	96.03	<b>97.94</b>
Glass	66.33	64.27	65.95	<b>67.24</b>	59.36	<b>64.38</b>
Heart	76.67	75.93	81.85	<b>82.22</b>	82.96	<b>83.33</b>
Hepatitis	82.50	<b>83.50</b>	87.17	86.33	85.33	<b>87.33</b>
ICU	86.82	<b>86.89</b>	93.13	93.08	92.61	<b>92.62</b>
Spam	88.57	86.77	88.16	<b>88.62</b>	88.00	<b>88.12</b>
Srbct	89.00	<b>93.33</b>	84.67	<b>96.67</b>	85.33	<b>94.00</b>

- 대체적으로 기존  $k$ -NN보다 좋은 성능
- 특히  $k=9$ 일 경우 모든 데이터 셋에 대해 성능 향상

# 5. 실험 디자인

## (4) 다양한 classifier와의 비교

Neighbor		Vector	Probability	Decision Tree	
NEC	LMNN	LSVM	NBC	C4.5	CART
 $d(x_i, x_j) = \sum_{l=1}^N \left( \frac{ n_{a_l}^{\delta}(x_i) }{ U } - \frac{ n_{a_l}^{\delta}(x_j) }{ U } \right)^2$	 $\theta(x) = \frac{1}{2}(\ x - NM(x)\  - \ x - NH(x)\ )$	 $\mathcal{L}(a, b, a) = \frac{1}{2} \ \tilde{a}\ ^2 - \sum_{i=1}^N \alpha_i [y_i(\tilde{a} \cdot \tilde{x}_i + b) - 1]$	 $\hat{y} = \underset{y}{\operatorname{argmax}} P(y) \prod_{i=1}^n P(x_i   y)$	 $H(X) = \sum_i P(x_i) I(x_i) = - \sum_i P(x_i) \log_2 P(x_i)$ $IG(T, \alpha) = H(T) - H(T \alpha)$	 $G(T) = \sum_{i=1}^n P(i T) P(i T) = 1 - \sum_i P(i T)^2 = 1 - \sum_i \left( \frac{n_i(T)}{n(T)} \right)^2$
거리가 $\delta$ 보다 작은 집합의 개수를 전체 집합의 개수로 나눈 각 Probability parameter들의 차이의 제곱들의 합을 이용하여 거리 함수 도출	Sample의 Nearest Miss와 Nearest Hit를 고려한 Hypothesis margin과 가중치를 이용한 loss 최소화를 통한 Distance metric	경계조건에 속하는 Support Vector들의 거리인 마진의 최대화와 제한 조건에 대한 라그랑주 승수법을 통해 최대 거리를 가지는 Decision boundary 결정	베이즈의 정리와 각 feature들의 조건부 독립이라는 Naïve bayes assumption에 의거 분류하고자 하는 대상의 각 class별 확률을 측정하여 확률이 큰 쪽으로 분류	분류기준이 엔트로피로서 Information Gain의 최대화를 위한 최적화를 통해 분류	분류기준이 Gini Impurity로서 Information Gain의 최대화를 위한 최적화를 통해 분류

## 5. 실험 디자인

### (4) 다양한 classifier와의 비교

Dataset	FNk NN	NEC	CART	LSVM	NBC	C4.5	LMNN
Dermatology	97.38	96.07	92.26	96.55	<u>97.54</u>	95.90	<b>97.94</b>
Glass	64.72	57.61	43.62	57.11	49.64	<b>68.80</b>	<u>68.17</u>
Heart	<b>85.19</b>	80.00	74.07	83.33	<u>84.07</u>	76.30	81.48
Hepatitis	86.83	85.00	<b>91.00</b>	86.17	88.39	<u>89.68</u>	85.33
ICU	<u>92.61</u>	86.29	79.40	92.56	90.50	91.50	<b>93.13</b>
Spam	87.85	81.00	<u>90.55</u>	89.79	79.29	<b>92.91</b>	89.57
Srbct	95.00	87.00	80.33	<b>98.33</b>	<u>95.20</u>	74.00	84.67
Average	87.08	81.85	78.75	86.26	83.51	84.16	85.76

- 평균 분류 정확도가 가장 높음
- 다른 classifier에 비해 모든 데이터에 대해 고른 성능

## 5. 실험 디자인

Dataset	FNk NN	NEC	CART	LSVM	NBC	C4.5	LMNN
Dermatology	97.38	96.07	92.26	96.55	<u>97.54</u>	95.90	<b>97.94</b>
Glass	64.72	57.61	43.62	57.11	49.64	<b>68.80</b>	<u>68.17</u>
Heart	<b>85.19</b>	80.00	74.07	83.33	<u>84.07</u>	76.30	81.48
Hepatitis	86.83	85.00	<b>91.00</b>	86.17	<u>88.39</u>	<u>89.68</u>	85.33
ICU	<u>92.61</u>	86.29	79.40	92.56	90.50	91.50	<b>93.13</b>
Spam	87.85	81.00	<u>90.55</u>	89.79	79.29	<b>92.91</b>	89.57
Srbct	95.00	87.00	80.33	<b>98.33</b>	<u>95.20</u>	74.00	84.67
Average	87.08	81.85	78.75	86.26	83.51	84.16	85.76

Order	FNk NN	NEC	CART	LSVM	NBC	C4.5	LMNN
1	Dermatology	Dermatology	Dermatology	Srbct	Dermatology	Dermatology	Dermatology
2	Srbct	Srbct	Hepatitis	Dermatology	Srbct	Spam	ICU
3	ICU	ICU	Spam	ICU	ICU	ICU	Spam
4	Spam	Hepatitis	Srbct	Spam	Hepatitis	Hepatitis	Hepatitis
5	Hepatitis	Spam	ICU	Hepatitis	Heart	Heart	Srbct
6	Heart	Heart	Heart	Heart	Spam	Srbct	Heart
7	Glass	Glass	Glass	Glass	Glass	Glass	Glass

Dataset	Samples	Features	Description	Analysis
Spam	4601	57	Continuous value57	Feature에 비해 굉장히 많은 Sample
Srbct	63	2308	Continuous value2308	Sample에 비해 굉장히 많은 Feature
Glass	214	9	Continuous value9	Feature가 적다
Heart	270	13	Discrete6, Binary3, Nominal3, Ordered1	Feature가 적다
ICU	200	20	Discrete3, Binary15, Nominal2	두 Dataset이 굉장히 유사, Continuous value의 차이
Hepatitis	155	19	Discrete4, Binary13, Continuous value2	
Dermatology	366	34	Linear value(Nominal33 & Discrete1)	대부분의 분류기들이 쉽게 분류 가능한 선형값

### 공통점

- Dermatology dataset(Linear value만으로 구성된 Features)에 대해서는 모든 분류기에서 정확도가 1위 혹은 2위(only LSVM)
- Feature의 개수가 가장 적은 Glass dataset의 경우 모든 분류기에서 가장 낮은 순위(7위), 그 다음으로 적은 Heart dataset에 대해서도 6위 혹은 5위(NBC & C4.5)

### 차이점

- CART dataset의 경우 Linear value 다음으로 Continuous value를 가지는 데이터셋 우위

## 6. 결론

- $k$ -NN의 정의 및 Distance Function 관련해 진행된 연구 소개
- $k$ -NN의 단점 중 Single Metric를 사용하면서 생기는 단점 지적
- 해결 위해 Neighborhood Information을 반영한 Distance function fusing 알고리즘 제안
- 다양한 데이터 셋에 대한 실험 통해 기존  $k$ -NN , 다양한 classifier과의 비교, 성능 향상 입증

## 7. 구현

- 구현 과정

- individual distance 기반의  $L_1$ 은 Euclidean distance 사용
- neighborhood distance 기반의  $L_2$ 는 논문 수식에 따라 구현
- 두 리스트  $L_1, L_2$ 를 결합하는 Spearman distance 기반의 목적 함수 또한 논문 수식에 따라 구현
- 목적 함수를 최적화하는 Monte Carlo Cross Entropy 알고리즘은 동일 저자의 다른 논문을 참조하여 구현

## 7. 구현

- 구현 결과

	$k = 1$		$k = 5$		$k = 9$		$k = 10$	
	paper	ours	paper	ours	paper	ours	paper	ours
Dermatology	96.56	95.62	97.76	97.55	97.94	97.53	97.38	97.00
Glass	64.27	66.60	67.24	67.25	64.38	65.32	64.72	64.85
Heart	75.93	75.93	82.22	82.59	83.33	82.59	85.19	82.22
Hepatitis	83.50	81.42	86.33	85.08	87.33	85.75	86.83	85.54
Srbct	93.33	87.92	96.67	84.17	94.00	84.31	95.00	83.06

## 7. 구현

- 파라미터 변화

	baseline	$k$	$w$		$N$	
		12	0.1 / 0.9	0.9 / 0.1	100	1500
Dermatology	97.53	97.00	95.89	96.17	96.99	96.73
Glass	65.32	62.47	67.68	61.58	67.25	63.53
Heart	82.59	81.85	80.00	79.63	83.33	82.96
Hepatitis	85.75	84.46	81.83	83.25	83.17	81.88
Srbct	84.31	84.17	84.31	82.08	82.92	84.44



## 7. 구현

- 최적 파라미터 일반화

	k				
	1	5	9	10	12
<b>Dermatology</b>	95.62	97.55	97.53	97	97
<b>Glass</b>	66.6	67.25	65.32	64.85	62.47
<b>Heart</b>	75.93	82.59	82.59	82.22	81.85
<b>Hepatitis</b>	81.42	85.08	85.75	85.54	84.46
<b>Srbct</b>	87.92	84.17	84.31	83.06	84.17
<b>Average</b>	81.498	83.328	83.1	82.534	81.99

	weight1 / weight2		
	0.5 / 0.5	0.1 / 0.9	0.9 / 0.1
<b>Dermatology</b>	97.53	95.89	96.17
<b>Glass</b>	65.32	67.68	61.58
<b>Heart</b>	82.59	80	79.63
<b>Hepatitis</b>	85.75	81.83	83.25
<b>Srbct</b>	84.31	84.31	82.08
<b>Average</b>	83.1	81.942	80.542

	N		
	1000	100	1500
<b>Dermatology</b>	97.53	95.89	96.17
<b>Glass</b>	65.32	67.68	61.58
<b>Heart</b>	82.59	80	79.63
<b>Hepatitis</b>	85.75	81.83	83.25
<b>Srbct</b>	84.31	84.31	82.08
<b>Average</b>	83.1	81.942	80.542

실험 데이터 셋에 대한 평균 정확도를 고려하였을 때  
FN  $k$ NN의 일반화된 설계 파라미터

- $k = 5$
- $L_1$ 과  $L_2$ 의 비율은 동일하게  $w_1 = 0.5$ ,  $w_2 = 0.5$
- $N = 1000$

## 7. 구현

- 결론 및 고찰

- $L_2$  계산 및 목적 함수를 최적화하는 과정에서 대부분의 시간 소모
- MC CE라는 확률적 최적화 방식을 사용하기 때문에 결과의 편차 존재
- 파라미터에 따라 민감한 성능 변화