บทที่ 4

ผลการทดลอง

4.1 การประเมินผลการทดลอง

การประเมินผลการทดลองสามารถทำได้โดยนำอัลกอริทึมเนียเรสท์เนเบอร์ที่ใช้ฟังก์ชันถ่วงน้ำหนักนั้นมาทำการทดสอบเปรียบเทียบกับอัลกอริทึมเนียเรสท์เนเบอร์แบบอื่นในการจำแนกประเภท สำหรับข้อมูลที่จะใช้สำหรับเปรียบเทียบนั้นได้เลือกที่จะใช้ข้อมูลจาก UCI Machine Learning Repository [20] ซึ่งเป็นชุดข้อมูลสำหรับทำการวัดประสิทธิภาพของอัลกอริทึมการเรียนรู้ของเครื่องที่ได้รับการยอมรับกันโดยทั่วไปรวมทั้งสิ้นจำนวน 9 ชุด และทำการสร้างชุดข้อมูลขึ้นเองโดยอาศัยสมการทางคณิตศาสตร์อีก 4 ชุด พร้อมทั้งใช้วิธีการตรวจสอบไขว้มาช่วยเพิ่มความถูกต้องของการประเมินผลด้วย โดยสำหรับข้อมูลที่ไม่ได้ถูกแบ่งกลุ่มสำหรับทำการสอนและทดสอบมาแล้วจะทำการตรวจสอบไขว้ 5 พับ

ชุดข้อมูลจาก UCI Machine Learning Repository ทั้ง 9 ชุด ที่นำมาใช้ในการทดสอบมีรายละเอียดดังที่แสดงอยู่ในตารางต่อไปนี้

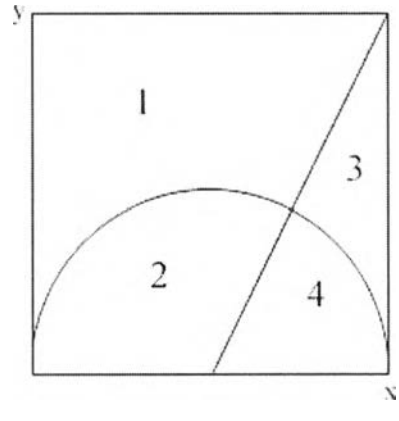
ตารางที่ 1 รายละเอียดเกี่ยวกับชุดข้อมูลจาก UCI Machine Learning Repository ที่นำมาใช้ในการทดสอบ

Dataset	Full Name	#classes	#attributes	#data
glass	Glass Identification Database	7	9	214
image	Image Segmentation Database	7	19	2310
ionosphere	Johns Hopkins University Ionosphere Database	2	34	351
iris	Iris Plants Database	3	4	150
liver	BUPA Liver Disorders Data	2	7	345
scale	Balance Scale Weight & Distance Database	3	4	625
tictactoe	Tic-Tac-Toe Endgame Database	2	9	958
wine	Wine Recognition Data	3	13	178
zoo	Zoo Database	7	17	100

โดยชุดข้อมูลเหล่านี้ทุกตัวไม่มีการแบ่งกลุ่มสำหรับการสอนและทดสอบไว้นอกจากชุดข้อมูล image ซึ่งจะถูกแบ่งกลุ่มสำหรับทำการสอนและทดสอบไว้ 1 ชุด โดยแบ่งเป็นข้อมูลสอน 210 ตัว และข้อมูลทดสอบ 2100 ตัว

สำหรับข้อมูลที่ใช้ทดสอบซึ่งสร้างขึ้นเองนั้นจะทำการสร้างข้อมูลโดยอาศัยสมการทางคณิตศาสตร์โดยใช้สมการเดียวกัน ทั้ง 4 ชุด โดยให้แต่ละชุดมีค่าของข้อมูลที่เป็นสัญญาณรบกวน (noisy data) แตกต่างกันได้แก่ 0%, 10%, 20% และ 30% เพื่อที่จะใช้แสดงให้เห็นถึงผลกระทบของข้อมูลที่มีสัญญาณรบกวนต่อประสิทธิภาพของฟังก์ชันที่ใช้ในการทดสอบแต่ละตัว ซึ่งเราจะทำการเรียกชุดข้อมูลเหล่านี้ว่า genpure, gennoise10%, gennoise20% และ gennoise30% ตามลำดับ โดยในแต่ละชุดนั้น จะมีจำนวนข้อมูลชุดละ 500 ตัว โดยข้อมูลที่ใช้นั้นมีจำนวนคุณสมบัติ 2 ชนิดซึ่งเป็นค่าพิกัดบนแกน x และแกน y โดยที่ $x \in [0,1]$ และ $y \in [0,1]$ และมีจำนวนผลลัพธ์ที่เป็นไปได้ 4 ชนิด ดังต่อไปนี้

1. $2x - y < 1$ และ $(x-0.5)^2 + y^2 > (0.5)^2$
2. $2x - y < 1$ และ $(x-0.5)^2 + y^2 \leq (0.5)^2$
3. $2x - y \geq 1$ และ $(x-0.5)^2 + y^2 > (0.5)^2$
4. $2x - y \geq 1$ และ $(x-0.5)^2 + y^2 \leq (0.5)^2$



รูปที่ 5 ผลลัพธ์ที่เป็นไปได้แต่ละชนิดของข้อมูลที่สร้างขึ้นเอง

ในการเปรียบเทียบประสิทธิภาพในการทำงานนั้นจะอาศัยการประเมินจากความแม่นยำในการจำแนกประเภทโดยจะนำความแม่นยำสำหรับข้อมูลแต่ละชุดมาทำการเฉลี่ยกันตามจำนวนข้อมูลที่ใช้ในการทดสอบ โดยสำหรับข้อมูล ที่ทำการตรวจสอบไขว้แบบหลายพับนั้นจะนับจากจำนวนข้อมูลทดสอบในทุกพับรวมกัน ซึ่งผลรวมนั้นก็จะมีค่าเท่ากับจำนวนข้อมูลทั้งหมดที่มี เช่น ชุดข้อมูล liver ซึ่งมีจำนวนข้อมูล 345 ตัว ก็จะมีน้ำหนักในการเฉลี่ยเท่ากับ 345 แต่สำหรับชุด

ข้อมูล image ซึ่งมีการแบ่งกลุ่มสำหรับเป็นข้อมูลสอนและข้อมูลทดสอบแล้วก็จะทำการนับจากจำนวนข้อมูลสอน นั่นก็คือชุดข้อมูล image จะมีน้ำหนักในการเฉลี่ยเท่ากับ 2100 ซึ่งก็คือจำนวนข้อมูลที่ใช้ในการทดสอบ ส่วนจำนวนข้อมูลสอนซึ่งมีค่าเท่ากับ 210 นั้นจะไม่ถูกนำมาพิจารณา

4.2 ผลที่ได้จากการทดลองและการวิเคราะห์

เราจะทำการเปรียบเทียบผลที่ได้นั่นก็คือความแม่นยำของกรณีต่างๆ โดยเริ่มแรกนั้นเราจะเสนอผลลัพธ์ที่ได้จากการใช้อัลกอริทึมเคเนียเรสท์เนเบอร์และอัลกอริทึมเนียเรสท์เนเบอร์แบบถ่วงน้ำหนักที่ใช้ฟังก์ชันถ่วงน้ำหนักแบบธรรมดาเพื่อใช้สำหรับเป็นตัวเปรียบเทียบ สำหรับผลการทดลองโดยใช้อัลกอริทึมเคเนียเรสท์เนเบอร์แบบดั้งเดิมได้ทำการปรับค่า k เป็น 3, 5 และ 10 ซึ่งเป็นช่วงของค่าปรับแต่งที่มักจะให้ความแม่นยำสูงที่สุดตามการทดลองในตำราและงานวิจัยต่างๆ รวมทั้งในงานวิจัยนี้ และผลที่ได้ก็เป็นดังต่อไปนี้

ตารางที่ 2 เปอร์เซ็นต์ความถูกต้องที่ได้จากอัลกอริทึมเคเนียเรสท์เนเบอร์ดั้งเดิม

Dataset	3-nn	5-nn	10-nn
glass	**68.69**	65.45	61.73
image	89.95	**90.05**	86.24
ionosphere	**86.31**	85.17	81.75
iris	94.67	**96.00**	**96.00**
liver	60.00	59.71	**62.32**
scale	82.08	85.28	**88.64**
tictactoe	**100.00**	99.79	97.91
wine	**96.10**	94.41	95.51
zoo	**95.00**	93.00	87.00
genpure	**95.00**	94.20	94.40
gennoise10%	83.60	83.80	**85.20**
gennoise20%	71.80	74.00	**75.40**
gennoise30%	53.00	57.40	**60.60**
Total	84.63	**85.13**	84.26

ส่วนในตารางที่ 3 แสดงถึงผลลัพธ์เมื่อใช้ฟังก์ชันถ่วงน้ำหนักแบบธรรมดาซึ่งจะใช้เป็นตัวเปรียบเทียบอีกตัวหนึ่ง

ตารางที่ 3 เปอร์เซ็นต์ความถูกต้องที่ได้จากฟังก์ชันถ่วงน้ำหนัก $f(x)=\frac{1}{x^t}$

Dataset	$t=2$	$t=3$	$t=5$	$t=10$
glass	64.52	**70.14**	68.72	68.69
image	90.14	91.33	92.48	**93.00**
ionosphere	65.53	66.39	68.39	**75.77**
iris	95.33	95.33	95.33	**96.00**
liver	61.74	**64.64**	63.19	61.16
scale	**90.40**	90.08	**90.40**	86.72
tictactoe	65.35	70.15	95.51	**98.95**
wine	**98.32**	97.19	97.21	96.10
zoo	82.00	91.00	**94.00**	**94.00**
genpure	95.00	**95.80**	95.40	95.00
gennoise10%	**83.40**	81.40	79.80	77.80
gennoise20%	**73.00**	68.60	64.60	62.80
gennoise30%	**55.20**	52.00	49.80	47.20
Total	79.73	80.55	83.82	**83.89**

ในขั้นตอนต่อไปเราจะเริ่มต้นแสดงผลที่ได้จากฟังก์ชันถ่วงน้ำหนักซึ่งมาจากวิทยานิพนธ์ฉบับนี้ โดยเราจะเริ่มจากการพิจารณาผลลัพธ์ที่ได้จากฟังก์ชันถ่วงน้ำหนักในรูปแบบที่ 1

$f(x)=\frac{1}{x^t}; t \geq n-1$

ตารางที่ 4 เปอร์เซ็นต์ความถูกต้องที่ได้จากฟังก์ชันถ่วงน้ำหนักในรูปแบบที่ 1

Dataset	$t=n-1$	$t=n$	$t=n+1$	$t=n+2$	$t=n+3$	$t=n+5$
glass	67.77	68.24	68.69	68.68	68.68	**69.16**
image	92.38	**92.52**	92.43	92.43	92.48	92.43
ionosphere	86.04	**86.32**	**86.32**	**86.32**	**86.32**	**86.32**
iris	95.33	95.33	95.33	95.33	95.33	**96.00**
liver	**63.19**	**63.19**	61.74	61.45	61.45	61.16
scale	90.08	89.92	**90.40**	87.84	87.04	86.72
tictactoe	98.54	98.64	98.95	98.95	99.06	**99.16**
wine	**96.10**	**96.10**	**96.10**	**96.10**	95.54	95.54
zoo	**94.00**	**94.00**	**94.00**	**94.00**	**94.00**	**94.00**
genpure	83.80	95.00	**95.80**	95.60	95.40	95.40
gennoise10%	77.00	**83.40**	81.40	80.20	79.80	78.60
gennoise20%	63.80	**73.00**	68.60	66.20	64.60	63.80
gennoise30%	**55.80**	55.20	52.00	49.80	49.80	48.20
Total	84.35	**86.28**	85.66	84.99	84.77	84.50

จากผลการทดลองจะเห็นได้ว่าหากเปรียบเทียบโดยอาศัยกรณีดีสุดแล้ว ผลลัพธ์ที่ได้จากฟังก์ชันรูปแบบที่ 1 จะดีกว่าผลที่ได้จากเคเนียเรสท์เนเบอร์และผลที่ได้จากฟังก์ชันถ่วงน้ำหนักในรูปแบบทั่วไปที่ใช้เป็นตัวเปรียบเทียบพอประมาณ โดย ผลลัพธ์จาก 5-เนียเรสท์เนเบอร์ มีความแม่นยำ 85.13% ผลที่ได้จาก $f(x)=\frac{1}{x^{10}}$ มีความแม่นยำ 83.89% และผลลัพธ์ที่ได้จาก $f(x)=\frac{1}{x^{n}}$ มีความแม่นยำ 86.28%

ผลลัพธ์ที่ได้ซึ่งดีกว่ากรณีที่ใช้เปรียบเทียบนั้นสอดคล้องกับสิ่งได้จากทางทฤษฎี โดยจากการทดลองนั้นจะพบได้ว่าค่าของ t น่าจะอยู่ในช่วง $(n-1,n+1)$ สิ่งหนึ่งที่น่าสังเกตคือความแม่นยำที่ได้จากชุดข้อมูลที่สร้างขึ้นมาเองของกรณีที่ t มีค่าต่ำๆของฟังก์ชันในรูปแบบที่ 1 โดยเฉพาะในกรณีที่ $t=n-1$ จะสังเกตได้ว่า เมื่อมีเปอร์เซ็นต์ความผิดพลาดของข้อมูลเพิ่มขึ้น ฟังก์ชันที่มีค่า t ต่ำๆ จะเริ่มให้ผลดีกว่าฟังก์ชันที่มีค่า t สูงๆ ปรากฏการณ์นี้เป็นเหตุเนื่องมาจาก

ฟังก์ชันที่มีค่า t ต่ำๆ นั้นจะมีอัตราการลดลงของฟังก์ชันด้วยความเร็วที่น้อยกว่าฟังก์ชันที่มีค่า t สูงๆ ซึ่งก็คือ ผลที่มาจากข้อมูลแต่ละตัวจะครอบคลุมบริเวณได้ไกลขึ้นนั้นซึ่งเปรียบได้กับอัลกอริทึมเคเนียเรสท์เนเบอร์กรณีที่มีค่า t เพิ่มขึ้น ซึ่งจะเป็นการนำข้อมูลมาพิจารณามากขึ้นทำให้ความน่าจะเป็นที่ข้อมูลซึ่งผิดพลาดจะส่งผลต่อการจำแนกประเภทลดลงไป

อีกประการหนึ่ง หากเราทำการเฉลี่ยค่าของข้อมูลเฉพาะส่วนที่ถูกนำมาจาก UCI Machine Learning Repository เราจะได้ค่าความแม่นยำสำหรับตารางที่ 3 โดยค่าที่มากที่สุดคือกรณีที่ $t = n$ มีความแม่นยำ 90.123% ตามมาด้วย $t = n + 1$ มีความแม่นยำ 90.122% และ $t = n - 1$ มีความแม่นยำ 90.02% ซึ่งหากสอดคล้องตามสมมุติฐานที่ว่าผลลัพธ์ที่ได้จากฟังก์ชันที่ใช้ค่าปรับแต่งขึ้นอยู่กับความผิดพลาดของข้อมูลแล้ว เราก็น่าจะสามารถอธิบายได้ว่าการที่ค่าเฉลี่ยแบบถ่วงน้ำหนักของความแม่นยำจากกรณีที่ $t = n - 1$ มีค่าต่ำกว่าค่าปรับแต่งอื่นนั้น ส่วนหนึ่งจะเป็นผลมาจากชุดข้อมูลที่สร้างขึ้นเองในกรณีที่มีค่าความผิดพลาดของข้อมูลน้อยๆ ซึ่งเป็นลักษณะที่ไม่ตรงกับข้อมูลจากความเป็นจริงซึ่งถูกนำมาจาก UCI Machine Learning Repository นั่นก็คือหากเราพิจารณาสัดส่วนความผิดพลาดของแต่ละชุดข้อมูลเราก็จะสามารถหาค่าปรับแต่งที่เหมาะสมที่สุดกับข้อมูลชุดนั้นๆ ได้ โดยสำหรับกรณีทั่วไปเมื่อเฉลี่ยแล้ว เราจะได้ว่าว่าค่าปรับแต่งที่น่าจะเหมาะที่สุดสำหรับฟังก์ชันในรูปแบบ $f(x) = \frac{1}{x^t}; t \geq n-1$ คือ $t = n$

ในขั้นตอนต่อไปเราจะทำการวิเคราะห์ผลที่ได้จากการประยุกต์ใช้ฟังก์ชัน $h(d)$ ทั้ง 3 รูปแบบ อันได้แก่ $h_1(d)$, $h_2(d)$ และ $h_3(d)$ กับฟังก์ชันในรูปแบบที่ 1 รวมทั้งผลที่เปลี่ยนแปลงไปเมื่อใช้จำนวนข้อมูลโดยรอบสำหรับคำนวณค่า $h(d)$ ต่างๆ กัน โดยผลลัพธ์ที่ได้นั้นมีดังในตารางต่อไปนี้

ตารางที่ 5 เปอร์เซ็นต์ความถูกต้องจากการใช้ $h_1(d)$ กับฟังก์ชันรูปแบบที่ 1 เมื่ออาศัยข้อมูลโดยรอบ 3 ตัว

Dataset	$t=n-1$	$t=n$	$t=n+1$	$t=n+2$	$t=n+3$
glass	62.58	63.98	66.31	**69.14**	68.67
image	88.24	88.95	89.62	90.00	**90.29**
ionosphere	74.92	79.48	83.75	84.62	**86.33**
iris	93.33	**94.67**	**94.67**	**94.67**	94.00
liver	62.32	**63.19**	**63.19**	62.61	62.90
scale	**89.92**	89.76	88.96	88.00	86.72
tictactoe	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
wine	96.62	97.19	**97.75**	**97.75**	**97.75**
zoo	86.00	86.00	87.00	**88.00**	**88.00**
genpure	85.60	**95.80**	**95.80**	**95.80**	95.40
gennoise10%	80.60	**84.00**	81.00	80.80	80.00
gennoise20%	67.80	**71.60**	68.40	66.00	64.40
gennoise30%	**56.20**	54.20	50.20	48.60	48.60
Total	83.09	**84.74**	84.46	84.30	84.15

ตารางที่ 6 เปอร์เซ็นต์ความถูกต้องจากการใช้ $h_1(d)$ กับฟังก์ชันรูปแบบที่ 1 เมื่ออาศัยข้อมูลโดยรอบ 5 ตัว

Dataset	$t = n-1$	$t = n$	$t = n+1$	$t = n+2$	$t = n+3$
glass	63.51	66.31	66.80	67.28	**67.74**
image	89.86	89.86	90.29	90.76	**91.10**
ionosphere	77.77	81.48	84.61	86.32	**87.18**
iris	93.33	**94.67**	94.00	**94.67**	**94.67**
liver	62.03	**63.48**	63.19	63.19	**63.48**
scale	**90.08**	89.92	88.64	87.52	86.40
tictactoe	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
wine	96.62	97.19	**97.75**	**97.75**	97.19
zoo	89.00	88.00	88.00	90.00	**91.00**
genpure	87.80	**96.20**	95.80	95.40	95.20
gennoise10%	80.60	**83.40**	81.00	80.60	80.00
gennoise20%	68.20	**72.20**	68.40	65.80	64.60
gennoise30%	**56.60**	53.80	50.20	48.80	48.60
Total	84.00	**85.24**	84.69	84.53	84.45

ตารางที่ 7 เปอร์เซ็นต์ความถูกต้องจากการใช้ $h_1(d)$ กับฟังก์ชันรูปแบบที่ 1 เมื่ออาศัยข้อมูลโดยรอบ 10 ตัว

Dataset	$t=n-1$	$t=n$	$t=n+1$	$t=n+2$	$t=n+3$
glass	64.92	67.73	**68.21**	67.29	67.30
image	90.38	90.67	90.90	**91.38**	91.33
ionosphere	80.63	83.75	86.31	87.17	**88.60**
iris	94.00	**94.67**	**94.67**	**94.67**	**94.67**
liver	61.45	63.77	**64.06**	**64.06**	62.61
scale	**90.08**	**90.08**	88.64	87.36	86.24
tictactoe	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
wine	**96.63**	**96.63**	**96.63**	**96.63**	**96.63**
zoo	89.00	**91.00**	**91.00**	**91.00**	**91.00**
genpure	88.20	**95.80**	**95.80**	**95.80**	95.40
gennoise10%	80.20	**83.80**	80.80	80.40	79.60
gennoise20%	68.20	**72.60**	68.60	65.60	65.00
gennoise30%	**56.80**	55.00	50.40	49.00	48.60
Total	84.35	**85.81**	85.09	84.79	84.52

ตารางที่ 8 เปอร์เซ็นต์ความถูกต้องจากการใช้ $h_1(d)$ กับฟังก์ชันรูปแบบที่ 1 เมื่ออาศัยข้อมูลโดยรอบ 20 ตัว

Dataset	$t=n-1$	$t=n$	$t=n+1$	$t=n+2$	$t=n+3$
glass	66.79	**69.17**	**69.17**	68.70	**69.17**
image	91.29	91.52	91.38	91.43	**91.67**
ionosphere	85.18	87.17	88.31	**88.88**	**88.88**
iris	94.67	94.67	94.67	**95.33**	**95.33**
liver	62.90	**64.35**	63.48	63.48	63.77
scale	**90.08**	89.12	88.32	87.20	86.24
tictactoe	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
wine	**97.21**	**97.21**	**97.21**	**97.21**	**97.21**
zoo	**91.00**	**91.00**	**91.00**	**91.00**	**91.00**
genpure	88.40	**95.80**	**95.80**	**95.80**	95.60
gennoise10%	79.80	**84.00**	81.20	80.60	80.00
gennoise20%	67.80	**72.80**	68.80	66.00	64.20
gennoise30%	**57.00**	55.40	49.80	49.40	49.00
Total	85.00	**86.30**	85.32	84.99	84.79

ตารางที่ 9 เปอร์เซ็นต์ความถูกต้องจากการใช้ $h_1(d)$ กับฟังก์ชันรูปแบบที่ 1 เมื่ออาศัยข้อมูลโดยรอบ 40 ตัว

Dataset	$t=n-1$	$t=n$	$t=n+1$	$t=n+2$	$t=n+3$
glass	69.65	69.65	69.17	70.56	**71.03**
image	91.86	92.00	92.10	**92.14**	**92.14**
ionosphere	88.88	90.59	**91.16**	90.88	**91.16**
iris	**95.33**	94.67	**95.33**	**95.33**	**95.33**
liver	62.03	**64.35**	64.06	63.48	63.48
scale	**89.92**	88.96	88.16	86.88	86.88
tictactoe	99.90	**100.00**	**100.00**	**100.00**	**100.00**
wine	**97.76**	97.21	97.21	97.21	97.21
zoo	**93.00**	**93.00**	**93.00**	**93.00**	**93.00**
genpure	88.00	95.80	**96.00**	95.80	95.60
gennoise10%	79.80	**84.00**	81.00	80.40	79.80
gennoise20%	68.00	**72.80**	68.60	66.00	64.20
gennoise30%	**57.40**	56.00	50.40	49.40	49.20
Total	85.44	**86.68**	85.76	85.34	85.17

ตารางที่ 10 เปอร์เซ็นต์ความถูกต้องจากการใช้ $h_1(d)$ กับฟังก์ชันรูปแบบที่ 1 เมื่ออาศัยข้อมูลโดยรอบ 80 ตัว

Dataset	$t = n-1$	$t = n$	$t = n+1$	$t = n+2$	$t = n+3$
glass	70.58	70.58	70.58	**71.51**	71.03
image	92.14	92.19	92.14	92.33	**92.38**
ionosphere	91.44	91.73	92.30	**92.59**	**92.59**
iris	95.33	95.33	95.33	**96.00**	**96.00**
liver	**63.77**	63.48	63.19	62.90	63.19
scale	**89.92**	88.48	88.00	87.04	86.72
tictactoe	99.27	99.27	99.48	99.79	**99.90**
wine	**97.21**	**97.21**	**97.21**	**97.21**	96.65
zoo	**93.00**	**93.00**	**93.00**	**93.00**	**93.00**
genpure	88.00	95.80	**96.00**	95.80	95.60
gennoise10%	79.20	**83.80**	81.20	80.40	79.80
gennoise20%	67.60	**73.20**	68.40	66.20	64.40
gennoise30%	**58.00**	56.40	50.80	50.00	49.80
Total	85.64	**86.70**	85.77	85.54	85.33

ตารางที่ 11 เปอร์เซ็นต์ความถูกต้องจากการใช้ $h_1(d)$ กับฟังก์ชันรูปแบบที่ 1 เมื่ออาศัยข้อมูลโดยรอบเป็นข้อมูลสอนที่เหลือทุกตัว

Dataset	$t = n-1$	$t = n$	$t = n+1$	$t = n+2$	$t = n+3$
glass	71.06	71.06	**71.53**	70.59	69.66
image	92.43	92.48	92.48	92.48	**92.52**
ionosphere	92.59	**92.88**	**92.88**	92.59	92.59
iris	**95.33**	**95.33**	**95.33**	**95.33**	**95.33**
liver	62.61	**63.48**	62.90	61.74	61.74
scale	**89.76**	89.44	87.68	87.20	86.40
tictactoe	**34.65**	**34.65**	**34.65**	**34.65**	**34.65**
wine	**97.76**	97.21	96.65	96.65	96.65
zoo	**93.00**	**93.00**	**93.00**	**93.00**	**93.00**
genpure	86.00	88.60	89.60	**90.60**	90.40
gennoise10%	76.40	**78.40**	76.80	75.40	74.60
gennoise20%	68.20	**68.60**	65.20	62.20	60.40
gennoise30%	**58.20**	53.80	51.60	49.80	49.40
Total	76.64	**76.71**	76.09	75.57	75.26

ตารางที่ 12 เปอร์เซ็นต์ความถูกต้องจากการใช้ $h_2(d)$ กับฟังก์ชันรูปแบบที่ 1 เมื่ออาศัยข้อมูลโดยรอบ 3 ตัว

Dataset	$t=n-1$	$t=n$	$t=n+1$	$t=n+2$	$t=n+3$
glass	60.25	63.99	67.29	**67.74**	**67.74**
image	89.52	89.71	89.90	90.19	**90.43**
ionosphere	78.34	80.91	84.89	86.32	**87.18**
iris	92.00	94.00	94.00	94.00	**94.67**
liver	61.16	62.03	62.32	62.32	**62.90**
scale	**89.92**	89.44	88.96	87.84	86.72
tictactoe	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
wine	97.17	**97.75**	**97.75**	**97.75**	**97.75**
zoo	89.00	89.00	**91.00**	**91.00**	**91.00**
genpure	87.60	95.80	**96.00**	95.80	95.20
gennoise10%	80.60	**83.00**	81.20	80.60	79.60
gennoise20%	68.00	**72.00**	68.20	66.00	64.80
gennoise30%	**56.80**	54.60	50.00	48.60	48.40
Total	83.75	**84.99**	84.63	84.39	84.23

ตารางที่ 13 เปอร์เซ็นต์ความถูกต้องจากการใช้ $h_2(d)$ กับฟังก์ชันรูปแบบที่ 1 เมื่ออาศัยข้อมูลโดยรอบ 5 ตัว

Dataset	$t=n-1$	$t=n$	$t=n+1$	$t=n+2$	$t=n+3$
glass	63.99	64.45	**68.68**	68.22	68.22
image	90.48	90.71	91.00	91.24	**91.52**
ionosphere	80.06	82.62	86.31	88.02	**88.88**
iris	93.33	**94.67**	**94.67**	**94.67**	**94.67**
liver	61.45	**63.19**	**63.19**	62.03	61.45
scale	**89.92**	89.76	88.80	87.04	85.92
tictactoe	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
wine	**97.19**	96.63	96.08	96.63	96.63
zoo	92.00	**93.00**	**93.00**	**93.00**	**93.00**
genpure	88.00	95.00	**95.80**	95.60	95.60
gennoise10%	80.20	**83.20**	81.40	80.60	79.40
gennoise20%	68.40	**72.80**	68.00	66.00	64.40
gennoise30%	**56.40**	55.20	50.00	49.40	48.80
Total	84.32	**85.57**	85.09	84.77	84.53

ตารางที่ 14 เปอร์เซ็นต์ความถูกต้องจากการใช้ $h_2(d)$ กับฟังก์ชันรูปแบบที่ 1 เมื่ออาศัยข้อมูลโดยรอบ 10 ตัว

Dataset	$t=n-1$	$t=n$	$t=n+1$	$t=n+2$	$t=n+3$
glass	64.92	68.24	68.70	68.70	**69.17**
image	91.29	91.33	91.62	**91.67**	**91.67**
ionosphere	85.46	87.74	89.16	**90.30**	**90.30**
iris	94.67	94.67	94.67	**95.33**	**95.33**
liver	61.45	63.48	63.19	**63.77**	62.61
scale	**90.08**	89.76	88.80	87.52	86.24
tictactoe	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
wine	96.65	**97.78**	97.21	97.21	97.21
zoo	92.00	**93.00**	**93.00**	**93.00**	**93.00**
genpure	88.60	95.40	**95.80**	95.60	95.60
gennoise10%	80.00	**84.00**	81.20	80.40	79.40
gennoise20%	68.00	**73.00**	68.00	65.60	64.40
gennoise30%	**56.80**	54.80	50.40	49.40	49.40
Total	84.92	**86.24**	85.46	85.14	84.83

ตารางที่ 15 เปอร์เซ็นต์ความถูกต้องจากการใช้ $h_2(d)$ กับฟังก์ชันรูปแบบที่ 1 เมื่ออาศัยข้อมูลโดยรอบ 20 ตัว

Dataset	$t = n-1$	$t = n$	$t = n+1$	$t = n+2$	$t = n+3$
glass	69.18	**71.52**	71.51	71.51	70.56
image	91.95	91.81	91.95	92.00	**92.14**
ionosphere	88.31	89.73	90.02	**90.31**	90.02
iris	94.67	94.67	**95.33**	**95.33**	**95.33**
liver	62.03	**64.06**	62.90	62.90	63.48
scale	**89.92**	88.80	87.84	86.72	86.72
tictactoe	99.58	99.69	99.90	**100.00**	**100.00**
wine	**97.78**	97.21	97.21	97.21	97.21
zoo	**93.00**	**93.00**	**93.00**	**93.00**	**93.00**
genpure	87.80	95.80	**96.00**	95.80	95.60
gennoise10%	80.20	**83.80**	80.80	80.40	80.00
gennoise20%	67.40	**73.20**	68.40	66.20	64.20
gennoise30%	**57.40**	55.40	51.40	49.80	49.60
Total	85.34	**86.54**	85.67	85.30	85.13

ตารางที่ 16 เปอร์เซ็นต์ความถูกต้องจากการใช้ $h_2(d)$ กับฟังก์ชันรูปแบบที่ 1 เมื่ออาศัยข้อมูลโดยรอบ 40 ตัว

Dataset	$t=n-1$	$t=n$	$t=n+1$	$t=n+2$	$t=n+3$
glass	70.59	71.05	70.58	71.04	**71.51**
image	92.19	92.24	92.14	92.29	**92.43**
ionosphere	91.44	91.73	91.73	**92.30**	**92.30**
iris	95.33	95.33	**96.00**	**96.00**	**96.00**
liver	62.03	**63.19**	**63.19**	**63.19**	**63.19**
scale	**89.92**	88.48	87.68	86.72	86.24
tictactoe	99.37	99.58	99.79	99.90	**100.00**
wine	**97.76**	97.21	97.21	97.21	96.65
zoo	**94.00**	**94.00**	**94.00**	**94.00**	**94.00**
genpure	87.80	95.80	**96.00**	95.80	95.60
gennoise10%	79.80	**83.80**	81.20	80.40	79.80
gennoise20%	67.20	**73.20**	68.60	66.20	64.20
gennoise30%	**57.20**	56.40	51.20	49.80	49.60
Total	85.56	**86.77**	85.83	85.50	85.30

ตารางที่ 17 เปอร์เซ็นต์ความถูกต้องจากการใช้ $h_2(d)$ กับฟังก์ชันรูปแบบที่ 1 เมื่ออาศัยข้อมูลโดยรอบ 80 ตัว

Dataset	$t=n-1$	$t=n$	$t=n+1$	$t=n+2$	$t=n+3$
glass	69.66	**70.59**	70.12	70.12	70.12
image	91.95	91.95	92.05	92.14	**92.29**
ionosphere	92.59	92.59	**92.87**	92.59	92.59
iris	**95.33**	**95.33**	**95.33**	**95.33**	**95.33**
liver	**63.19**	62.03	**63.19**	**63.19**	**63.19**
scale	**89.76**	88.32	87.84	86.72	86.56
tictactoe	99.06	99.16	**99.27**	**99.27**	**99.27**
wine	**96.65**	**96.65**	**96.65**	**96.65**	**96.65**
zoo	**94.00**	**94.00**	**94.00**	**94.00**	**94.00**
genpure	87.40	95.40	**95.80**	**95.80**	95.40
gennoise10%	78.60	**84.00**	81.00	80.60	79.80
gennoise20%	67.00	**73.40**	68.40	66.20	64.40
gennoise30%	**58.00**	56.00	51.20	50.20	50.00
Total	85.42	**86.54**	85.72	85.37	85.17

ตารางที่ 18 เปอร์เซ็นต์ความถูกต้องจากการใช้ $h_2(d)$ กับฟังก์ชันรูปแบบที่ 1 เมื่ออาศัยข้อมูลโดยรอบเป็นข้อมูลสอนที่เหลือทุกตัว

Dataset	$t = n-1$	$t = n$	$t = n+1$	$t = n+2$	$t = n+3$
glass	67.33	67.79	68.70	68.70	**69.17**
image	92.38	92.38	92.38	**92.43**	92.29
ionosphere	**89.17**	**89.17**	**89.17**	**89.17**	**89.17**
iris	**95.33**	94.00	94.67	94.67	94.67
liver	61.16	**61.45**	61.16	**61.45**	59.71
scale	**89.76**	89.60	87.84	87.36	86.72
tictactoe	98.95	98.95	99.06	**99.16**	**99.16**
wine	**96.10**	95.54	95.54	95.54	95.54
zoo	**94.00**	**94.00**	**94.00**	**94.00**	**94.00**
genpure	83.60	95.20	**95.80**	**95.80**	95.40
gennoise10%	76.60	**83.80**	81.00	80.60	79.80
gennoise20%	64.60	**73.40**	68.80	66.20	64.40
gennoise30%	55.40	**56.00**	51.20	50.00	49.80
Total	84.41	**86.38**	85.44	85.15	84.75

ตารางที่ 19 เปอร์เซ็นต์ความถูกต้องจากการใช้ $h_3(d)$ กับฟังก์ชันรูปแบบที่ 1 เมื่ออาศัยข้อมูลโดยรอบ 3 ตัว

Dataset	$t=n-1$	$t=n$	$t=n+1$	$t=n+2$	$t=n+3$
glass	63.06	64.44	66.79	**69.14**	68.21
image	88.62	89.43	89.67	90.14	**90.33**
ionosphere	75.48	79.48	83.75	84.90	**86.33**
iris	92.67	**94.67**	**94.67**	94.00	94.00
liver	61.45	**63.19**	**63.19**	62.90	62.61
scale	**89.92**	89.76	88.96	88.00	86.72
tictactoe	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
wine	96.62	97.19	**97.75**	**97.75**	**97.75**
zoo	86.00	87.00	88.00	**91.00**	90.00
genpure	86.60	95.80	**96.20**	95.60	95.20
gennoise10%	80.60	**83.40**	81.20	80.60	79.60
gennoise20%	67.80	**72.20**	68.20	65.80	64.60
gennoise30%	**56.60**	54.40	49.80	48.00	48.20
Total	83.29	**84.93**	84.50	84.32	84.10

ตารางที่ 20 เปอร์เซ็นต์ความถูกต้องจากการใช้ $h_3(d)$ กับฟังก์ชันรูปแบบที่ 1 เมื่ออาศัยข้อมูลโดยรอบ 5 ตัว

Dataset	$t=n-1$	$t=n$	$t=n+1$	$t=n+2$	$t=n+3$
glass	63.98	65.38	**67.28**	66.81	66.35
image	90.05	90.38	90.48	90.86	**91.10**
ionosphere	77.77	81.19	84.90	86.61	**87.18**
iris	92.67	94.00	94.00	**94.67**	**94.67**
liver	62.32	**63.19**	62.90	62.90	62.61
scale	**90.08**	89.92	88.64	87.52	86.24
tictactoe	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
wine	96.62	97.19	97.19	**97.75**	97.19
zoo	88.00	89.00	**91.00**	**91.00**	**91.00**
genpure	88.00	95.60	**95.80**	95.60	95.40
gennoise10%	80.20	**83.40**	81.00	80.60	79.40
gennoise20%	68.00	**72.40**	68.20	65.80	64.40
gennoise30%	**57.00**	54.60	50.20	49.00	48.80
Total	84.06	**85.37**	84.77	84.59	84.32

ตารางที่ 21 เปอร์เซ็นต์ความถูกต้องจากการใช้ $h_3(d)$ กับฟังก์ชันรูปแบบที่ 1 เมื่ออาศัยข้อมูลโดยรอบ 10 ตัว

Dataset	$t=n-1$	$t=n$	$t=n+1$	$t=n+2$	$t=n+3$
glass	64.91	68.19	**68.22**	67.29	67.76
image	90.43	90.81	90.90	**91.29**	91.14
ionosphere	81.20	84.04	86.60	87.74	**89.17**
iris	94.00	**94.67**	**94.67**	**94.67**	**94.67**
liver	61.74	63.48	63.48	**63.77**	62.32
scale	**90.08**	**90.08**	88.80	87.36	86.24
tictactoe	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
wine	**97.21**	96.63	96.63	96.63	**97.21**
zoo	**93.00**	**93.00**	**93.00**	**93.00**	**93.00**
genpure	88.60	**95.60**	**95.80**	**95.60**	**95.60**
gennoise10%	79.80	**84.00**	80.80	80.40	79.40
gennoise20%	68.20	**73.00**	68.00	65.60	64.40
gennoise30%	**57.00**	54.80	50.40	49.40	48.80
Total	84.49	**85.91**	85.07	84.82	84.50

ตารางที่ 22 เปอร์เซ็นต์ความถูกต้องจากการใช้ $h_3(d)$ กับฟังก์ชันรูปแบบที่ 1 เมื่ออาศัยข้อมูลโดยรอบ 20 ตัว

Dataset	$t=n-1$	$t=n$	$t=n+1$	$t=n+2$	$t=n+3$
glass	67.25	**69.65**	69.17	69.63	69.17
image	91.52	91.62	91.62	91.52	**91.86**
ionosphere	86.60	88.02	89.16	**89.73**	**89.73**
iris	94.67	94.67	**95.33**	**95.33**	**95.33**
liver	62.32	**64.06**	62.61	62.61	63.19
scale	**90.08**	88.96	88.00	87.04	86.24
tictactoe	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
wine	**97.78**	97.21	97.21	97.21	97.21
zoo	**92.00**	**92.00**	**92.00**	**92.00**	**92.00**
genpure	87.80	95.80	**96.00**	95.80	95.60
gennoise10%	80.00	**83.80**	80.80	80.20	80.00
gennoise20%	67.60	**72.60**	68.60	66.20	64.00
gennoise30%	**57.60**	55.60	50.40	49.60	49.60
Total	85.16	**86.36**	85.40	85.04	84.90

ตารางที่ 23 เปอร์เซ็นต์ความถูกต้องจากการใช้ $h_3(d)$ กับฟังก์ชันรูปแบบที่ 1 เมื่ออาศัยข้อมูลโดยรอบ 40 ตัว

Dataset	$t=n-1$	$t=n$	$t=n+1$	$t=n+2$	$t=n+3$
glass	70.58	70.59	70.58	**71.97**	71.51
image	91.95	92.00	92.00	**92.14**	92.10
ionosphere	89.74	90.59	91.16	90.88	**91.45**
iris	**95.33**	**95.33**	**95.33**	**95.33**	**95.33**
liver	61.74	63.19	**63.48**	63.19	63.19
scale	**89.92**	88.96	87.84	86.72	86.72
tictactoe	99.79	99.90	**100.00**	**100.00**	**100.00**
wine	**97.76**	97.21	97.21	97.21	97.21
zoo	**94.00**	**94.00**	**94.00**	**94.00**	**94.00**
genpure	87.80	95.80	**96.00**	95.80	95.60
gennoise10%	79.60	**83.80**	81.20	80.40	79.80
gennoise20%	67.20	**73.00**	68.40	66.00	64.20
gennoise30%	**57.20**	56.20	51.00	49.80	49.80
Total	85.43	**86.68**	85.77	85.40	85.22

ตารางที่ 24 เปอร์เซ็นต์ความถูกต้องจากการใช้ $h_3(d)$ กับฟังก์ชันรูปแบบที่ 1 เมื่ออาศัยข้อมูลโดยรอบ 80 ตัว

Dataset	$t = n-1$	$t = n$	$t = n+1$	$t = n+2$	$t = n+3$
glass	70.59	70.59	**71.98**	71.52	71.04
image	91.90	91.95	92.05	**92.29**	**92.29**
ionosphere	92.30	92.59	**92.87**	**92.87**	**92.87**
iris	95.33	95.33	95.33	95.33	**96.00**
liver	62.61	62.61	63.19	**63.48**	62.90
scale	**89.92**	88.32	88.00	86.88	86.56
tictactoe	99.27	99.27	99.27	99.58	**99.79**
wine	**97.21**	**97.21**	**97.21**	96.65	96.65
zoo	**94.00**	**94.00**	**94.00**	**94.00**	**94.00**
genpure	87.60	95.40	95.80	**96.00**	95.60
gennoise10%	78.60	**84.00**	81.00	80.60	79.80
gennoise20%	67.00	**73.40**	68.40	66.20	64.40
gennoise30%	**57.60**	56.20	51.20	50.20	50.00
Total	85.43	**86.61**	85.80	85.56	85.30

ตารางที่ 25 เปอร์เซ็นต์ความถูกต้องจากการใช้ $h_3(d)$ กับฟังก์ชันรูปแบบที่ 1 เมื่ออาศัยข้อมูลโดยรอบเป็นข้อมูลสอนที่เหลือทุกตัว

Dataset	$t=n-1$	$t=n$	$t=n+1$	$t=n+2$	$t=n+3$
glass	70.13	**70.60**	70.12	69.66	69.66
image	92.43	92.43	92.38	92.38	**92.48**
ionosphere	**92.87**	**92.87**	**92.87**	92.59	92.59
iris	**96.00**	95.33	95.33	94.67	94.67
liver	61.45	**62.32**	62.03	61.45	61.45
scale	**89.76**	89.60	87.68	87.20	86.56
tictactoe	98.95	98.95	99.06	99.16	**99.27**
wine	**97.21**	96.65	96.65	96.65	96.65
zoo	**94.00**	**94.00**	**94.00**	**94.00**	**94.00**
genpure	84.00	95.20	**95.80**	**95.80**	95.40
gennoise10%	77.00	**83.80**	81.00	80.60	79.80
gennoise20%	64.80	**73.40**	68.80	66.20	64.40
gennoise30%	55.20	**56.00**	51.20	50.00	49.80
Total	84.80	**86.77**	85.74	85.34	85.10

เพื่อให้ง่ายต่อการเปรียบเทียบเราจึงทำการแสดงผลของกรณีที่มีความถูกต้องสูงที่สุดจากตารางที่ 5-25 ลงในตารางต่อไปนี้

ตารางที่ 26 สรุปเปอร์เซ็นต์ความถูกต้องจากการใช้ $h(d)$ กับฟังก์ชันรูปแบบที่ 1 เมื่ออาศัยข้อมูลโดยรอบจำนวนต่างๆ กัน k ตัว

	$h_1(d)$	$h_2(d)$	$h_3(d)$
$k=3$	84.74	84.99	84.93
$k=5$	85.24	85.57	85.37
$k=10$	85.81	86.24	85.91
$k=20$	86.30	86.54	86.36
$k=40$	86.68	**86.77**	86.68
$k=80$	**86.70**	86.54	86.61
$k=all$	76.71	86.38	**86.77**

จากผลการทดลอง ผลที่ได้จากการใช้ฟังก์ชัน $h_1(d)$ $h_2(d)$ และ $h_3(d)$ กับฟังก์ชันในรูปแบบแรกต่างก็มีค่าใกล้เคียงกัน และเมื่อเปรียบเทียบกับค่าความแม่นยำของฟังก์ชันในรูปแบบแรกก่อนที่จะใช้ฟังก์ชัน $h(d)$ นั้น ปรากฏว่าเมื่อคำนวณค่า $h(d)$ โดยใช้อาศัยจำนวนข้อมูลโดยรอบน้อยๆ แล้วการใช้ฟังก์ชันรูปแบบที่ 1 โดยไม่ใช้ฟังก์ชัน $h(d)$ คูณเข้าไปจะให้ผลดีกว่า แต่ว่าส่วนใหญ่แล้วก็มีความแม่นยำไม่แตกต่างกันมากนัก และเมื่อเพิ่มจำนวนข้อมูลโดยรอบที่ใช้ในการคำนวณค่า $h(d)$ ก็จะเห็นได้ว่าค่าความถูกต้องจะเพิ่มขึ้นเรื่อยๆ จนกระทั่งเหนือกว่าค่าความถูกต้องที่ได้จากฟังก์ชันรูปแบบที่ 1 ซึ่งมีค่าเท่ากับ 86.28% ในที่สุดสำหรับการใช้ฟังก์ชัน $h(d)$ ทั้ง 3 รูปแบบ

เราสามารถตั้งข้อสังเกตจากลักษณะของผลการทดลองได้ดังต่อไปนี้

1. ฟังก์ชัน $h_2(d)$ ซึ่งใช้ค่าเฉลี่ยเลขคณิตของระยะทางที่ยกกำลังด้วยจำนวนมิติแล้วให้ผลลัพธ์ที่ดีที่สุด ตามมาด้วยฟังก์ชัน $h_3(d)$ ซึ่งใช้ค่าเฉลี่ยเลขคณิตของระยะทางแล้วนำมายกกำลังทีหลัง ส่วนฟังก์ชัน $h_1(d)$ ซึ่งคาดว่าจะให้ผลที่ดีนั้น ในการทดลองที่ผ่านมาให้ค่าความถูกต้องต่ำที่สุด

2. ผลลัพธ์ที่ดีที่สุดเมื่อนำฟังก์ชัน $h(d)$ ยังคงเป็นกรณีที่ฟังก์ชันรูปแบบที่ 1 ใช้ค่าปรับแต่ง $t=n$ อยู่ โดยจะสังเกตได้ว่าในตารางที่ 5-25 นั้น กรณีที่ใช้ค่า $t=n$ ค่าความถูกต้องสูงสุดทุกตาราง

3. ค่า k ที่เหมาะสมในการคำนวณค่า $h(d)$ ที่จะเอามาใช้คูณเข้ากับฟังก์ชันถ่วงน้ำหนักนั้นไม่ใช่ค่า k ที่เหมาะสมที่สุดของอัลกอริทึมเคเนียเรสท์เนเบอร์แม้ว่าจะใช้วิธีการคำนวณที่คล้ายกัน โดยจะสังเกตได้จากในตารางที่ 2 ซึ่งค่าความถูกต้องของอัลกอริทึมเคเนียเรสท์เนเบอร์จะมีค่าสูงสุดเมื่อให้ค่า $k = 5$ แต่ในตารางที่ 26 ซึ่งเป็นการสรุปผลจากการใช้ค่า $h(d)$ ทั้งหมดจะพบว่าค่าความแม่นยำจะในช่วงที่ $k > 20$ ซึ่งเป็นค่าที่ค่อนข้างมากถ้าจะใช้กับอัลกอริทึมเคเนียเรสท์เนเบอร์

จากการวิเคราะห์ การที่ผลการทดลองบางส่วนผิดไปจากที่คาดไว้อาจมาจากสาเหตุหลักๆ ดังนี้

1. การใช้ฟังก์ชัน $h_1(d)$, $h_2(d)$ และ $h_3(d)$ ประมาณขอบเขตโดยรอบของข้อมูลหรือขนาดของเซลล์ของแผนภาพโวโรนอยได้ไม่ดีพอ โดยอาจเป็นเหตุเนื่องมาจากในการประมาณนี้ต้องอาศัยค่าปัจจัยคือจำนวนข้อมูลโดยรอบที่จะใช้ในการประมาณและพิจารณาจากข้อมูลโดยรอบที่อยู่ใกล้ที่สุดโดยไม่ได้พิจารณาถึงแนวการวางตัวของข้อมูล ซึ่งนี่เองอาจเป็นเหตุผลที่ผลที่ได้แย่กว่าก่อนใช้เมื่อใช้จำนวนข้อมูลโดยรอบในการคำนวณน้อยๆ

2. การคูณด้วยฟังก์ชัน $f(x)$ ด้วย $h(d)$ อาจประมาณผลของการกระจายตัวแบบสม่ำเสมอได้ไม่ดีพอ โดยจะสังเกตได้ว่าหากค่า x ซึ่งเป็นระยะห่างระหว่างข้อมูลสอนกับจุดที่สนใจมีค่าไม่มากนักเมื่อเทียบค่า d ซึ่งเป็นระยะทางที่เป็นเหมือนรัศมีของบริเวณโดยรอบของข้อมูลสอนนั้น การประมาณว่าข้อมูลใดๆ ที่อยู่ในอาณาเขตนั้นจะอยู่ห่างจากข้อมูลที่สนใจด้วยระยะทางเท่ากับ x เหมือนข้อมูลสอนจะได้ผลไม่ค่อยถูกต้อง

3. การใช้ $h(d)$ คูณเข้าไปจะทำให้ข้อมูลสอนที่ผิดพลาดมีผลต่อการจำแนกประเภทเพิ่มขึ้น โดยหากสังเกตผลลัพธ์ที่ได้จากชุดข้อมูล genpure ซึ่งเป็นชุดข้อมูลที่สร้างขึ้นโดยไม่มีข้อผิดพลาด จะพบได้ว่าสำหรับชุดข้อมูลนี้เมื่อใช้ฟังก์ชัน $h(d)$ จะให้ผลลัพธ์ที่ดีกว่าโดยเฉพาะในกรณีที่ $t = n - 1$ จึงน่าจะสรุปได้ว่าสาเหตุที่ผลการทำงานผิดไปจากที่คาดการณ์ไว้นั้นอาจเกิดจากข้อมูลที่เป็นข้อผิดพลาด

4. จำนวนของข้อมูลโดยรอบที่ใช้ในการพิจารณาในการคำนวณค่า $h_1(d)$ $h_2(d)$ และ $h_3(d)$ น้อยเกินไป โดยจะสังเกตได้จากผลการทดลองกรณีที่ใช้ข้อมูลโดยรอบในการพิจารณา 3 ตัว 5 ตัว และ 10 ตัวว่าเมื่อเพิ่มจำนวนโดยรอบของข้อมูลที่ใช้ในการพิจารณา ค่าความแม่นยำของผลลัพธ์ที่ได้จะเพิ่มขึ้น เพราะการที่ใช้จำนวนข้อมูลโดยรอบในการคำนวณค่า $h(d)$ ย่อมทำให้ค่า $h(d)$ ที่ได้มีความแปรปรวนสูง และหากใช้จำนวนข้อมูลโดยรอบมากเกินไปก็จะทำให้ค่า $h(d)$ ที่ได้ไม่แม่นยำเพราะผลจากข้อมูลที่แตกต่างกันมากจนทำให้ผลรวมแสดงถึงข้อมูลโดยรอบได้แย่ลง และจากกรณีที่ใช้ข้อมูลที่เหลือทั้งหมดในการคำนวณก็จะทำให้ข้อมูลที่เป็นจุดสุด

ขีด (extreme point) ซึ่งเป็นจุดที่อยู่ไกลที่สุดในแนวใดแนวหนึ่ง มีค่าที่ได้สูงมากๆ ซึ่งนี่เองก็เป็นปัญหาเดียวกับการกำหนดค่า k ของอัลกอริทึมเคเนียเรสท์เนเบอร์ที่เป็นจำนวนของข้อมูลโดยรอบที่จะใช้ในการจำแนกประเภท โดยจากผลการทดลองในตารางที่ 26 นั้นจะสังเกตได้ว่าเมื่อใช้ข้อมูลทุกตัวเป็นข้อมูลโดยรอบผลที่ได้จากฟังก์ชัน $h_1(d)$ และ $h_2(d)$ จะมีค่าความถูกต้องน้อยลง ส่วนสำหรับกรณีที่ค่าความถูกต้องของฟังก์ชัน $h_3(d)$ มีค่าเพิ่มขึ้นนั้นคาดว่าจะเป็นผลมาจากลักษณะเฉพาะของข้อมูล โดยหากพิจารณาจากค่าในตารางที่ 24 และตารางที่ 25 ซึ่งเป็นผลจากฟังก์ชัน $h_3(d)$ จะเห็นได้ว่ากรณีที่เมื่อใช้ข้อมูลทั้งหมดเป็นข้อมูลโดยรอบแล้วมีค่าความถูกต้องเพิ่มขึ้นจะมีเฉพาะกรณีที่ $t = n$ เท่านั้น โดยกรณีที่เหลือจะมีค่าความถูกต้องลดลงทั้งหมด

5. การที่ฟังก์ชัน $h_1(d)$ ให้ค่าความถูกต้องต่ำกว่าอีก 2 ฟังก์ชันที่เหลืออาจเป็นผลมาจากลักษณะของการใช้ค่าเฉลี่ยเรขาคณิตซึ่งถ้ามีค่าใดค่าหนึ่งใกล้ศูนย์มากๆ ค่านั้นจะมีผลต่อการเฉลี่ยมากไปด้วย และหากมีค่าใดค่าหนึ่งเป็นศูนย์ก็จะทำให้ค่าเฉลี่ยที่ได้เป็นศูนย์ไปโดยไม่ต้องสนใจค่าอื่นที่นำมาใช้ในการเฉลี่ยด้วยเลย

ในขั้นตอนต่อไปเราจะทำการแสดงและวิเคราะห์ผลลัพธ์ที่ได้จากฟังก์ชันรูปแบบที่สอง โดยในการทดลองทั้งหมดที่เกี่ยวข้องกับรูปแบบย่อยทั้ง 2 แบบของฟังก์ชันรูปแบบที่สอง ซึ่งสามารถที่จะเขียนออกมาเป็นสมการได้ว่า $f(x) = \frac{1}{x^{n-1}} \exp(-cx / \sqrt[n]{h(d)}); c > 0$ และ $f(x) = \frac{1}{x^{n-1}} \exp(-cx / \sqrt[n]{h(d)}); c > 0$ นั้น เราได้ใช้ $h_3(d)$ ซึ่งจะทำให้ $\sqrt[n]{h(d)}$ เป็นค่าเฉลี่ยเลขคณิตของระยะทางธรรมดา สำหรับสาเหตุที่เลือกใช้ค่าเฉลี่ยธรรมดาเนื่องจากการเฉลี่ยหลังจากเปลี่ยนเป็นขนาดบริเวณโดยรอบซึ่งอยู่ใน n มิติแล้วโดยการใช้ $h_2(d)$ แล้วนำมาประมาณค่าเฉลี่ยระยะทางซึ่งมีมิติเดียวจะเป็นการไม่เหมาะสมเพราะค่าที่ต้องการประมาณไม่ใช่ขนาดบริเวณโดยรอบซึ่งมีหลายมิติ และในการใช้ค่าเฉลี่ยเรขาคณิตหรือ $h_1(d)$ จะมีข้อเสียเนื่องจากกรณีที่มีค่าใดอยู่ใกล้กันมากๆ จะทำให้ $h_1(d)$ เข้าใกล้ 0 ด้วย และถ้าหากมีค่าใดค่าหนึ่งเป็น 0 ก็จะทำให้ค่าเฉลี่ยเรขาคณิตเป็น 0 ไปในทันที(เราสามารถลดผลจากปรากฏการณ์นี้ได้เช่นกัน โดยการบวกค่าคงที่ค่าหนึ่งให้ข้อมูลทุกตัวก่อนจะนำไปหาค่าเฉลี่ยกันแล้วจึงนำค่านั้นมาลบออกภายหลัง)

ตารางที่ 27 เปอร์เซ็นต์ความถูกต้องจากฟังก์ชันรูปแบบที่ 2 เมื่อใช้ $\overline{\sqrt[n]{h_3(d)}}$ โดยอาศัยข้อมูลโดยรอบ 3 ตัว

Dataset	$c = 0.1$	$c = 0.2$	$c = 0.3$	$c = 0.4$	$c = 0.5$	$c = 1$
glass	67.77	67.77	67.77	68.24	68.24	**68.70**
image	92.43	**92.48**	**92.48**	**92.48**	**92.48**	92.43
ionosphere	86.04	**86.32**	**86.32**	**86.32**	**86.32**	**86.32**
iris	**95.33**	**95.33**	**95.33**	**95.33**	**95.33**	**95.33**
liver	62.03	**62.61**	62.03	62.32	62.32	63.19
scale	**89.92**	**89.92**	**89.92**	**89.92**	**89.92**	**89.92**
tictactoe	98.54	98.54	98.64	98.64	98.64	**98.95**
wine	**96.10**	**96.10**	**96.10**	**96.10**	**96.10**	**96.10**
zoo	**94.00**	**94.00**	**94.00**	**94.00**	**94.00**	**94.00**
genpure	93.80	95.40	95.60	95.20	**95.80**	95.60
gennoise10%	84.00	**85.40**	**85.40**	**85.40**	84.80	83.80
gennoise20%	74.20	75.20	**75.80**	75.60	74.60	71.60
gennoise30%	**61.20**	**61.20**	60.60	59.80	59.20	54.60
Total	86.63	**86.97**	86.97	86.90	86.78	86.24

ตารางที่ 28 เปอร์เซ็นต์ความถูกต้องจากฟังก์ชันรูปแบบที่ 2 เมื่อใช้ $\overline{\sqrt[n]{h_3(d)}}$ โดยอาศัยข้อมูลโดยรอบ 5 ตัว

Dataset	$c = 0.1$	$c = 0.2$	$c = 0.3$	$c = 0.4$	$c = 0.5$	$c = 1$
glass	67.77	67.77	67.77	68.24	68.24	**68.70**
image	92.43	**92.48**	**92.48**	**92.48**	**92.48**	92.43
ionosphere	86.04	**86.32**	**86.32**	**86.32**	**86.32**	**86.32**
iris	**95.33**	**95.33**	**95.33**	**95.33**	**95.33**	**95.33**
liver	62.03	62.32	62.03	62.32	62.61	**62.90**
scale	**89.92**	**89.92**	**89.92**	**89.92**	**89.92**	89.76
tictactoe	98.54	98.54	98.64	98.64	98.64	**98.95**
wine	**96.10**	**96.10**	**96.10**	**96.10**	**96.10**	**96.10**
zoo	**94.00**	**94.00**	**94.00**	**94.00**	**94.00**	**94.00**
genpure	93.00	95.00	**95.60**	**95.60**	95.40	95.40
gennoise10%	83.40	85.00	**85.80**	**85.80**	85.40	83.80
gennoise20%	73.00	74.80	75.60	**75.80**	75.60	73.00
gennoise30%	**61.40**	**61.40**	60.80	60.60	59.80	56.60
Total	86.46	86.88	87.00	**87.03**	86.93	86.44

ตารางที่ 29 เปอร์เซ็นต์ความถูกต้องจากฟังก์ชันรูปแบบที่ 2 เมื่อใช้ $\sqrt[n]{h_3(d)}$ โดยอาศัยข้อมูลโดยรอบ 10 ตัว

Dataset	$c = 0.1$	$c = 0.2$	$c = 0.3$	$c = 0.4$	$c = 0.5$	$c = 1$
glass	67.77	67.77	67.77	67.77	68.24	**68.70**
image	92.43	**92.48**	**92.48**	**92.48**	**92.48**	**92.48**
ionosphere	86.04	**86.32**	**86.32**	**86.32**	**86.32**	**86.32**
iris	**95.33**	**95.33**	**95.33**	**95.33**	**95.33**	**95.33**
liver	62.03	62.32	62.32	62.03	62.61	**62.90**
scale	**89.92**	**89.92**	**89.92**	**89.92**	**89.92**	89.76
tictactoe	98.54	98.54	98.64	98.64	98.64	**98.95**
wine	**96.10**	**96.10**	**96.10**	**96.10**	**96.10**	**96.10**
zoo	**94.00**	**94.00**	**94.00**	**94.00**	**94.00**	**94.00**
genpure	91.80	94.00	95.20	**95.60**	**95.60**	**95.60**
gennoise10%	82.80	84.60	85.40	**85.60**	85.40	83.60
gennoise20%	70.80	74.60	74.80	75.60	**75.80**	74.40
gennoise30%	**61.20**	**61.20**	61.00	60.80	60.60	58.00
Total	86.16	86.75	86.91	86.98	**87.01**	86.65

ตารางที่ 30 สรุปเปอร์เซ็นต์ความถูกต้องจากฟังก์ชันรูปแบบที่ 2 เมื่อใช้ $\sqrt[3]{h_3(d)}$ โดยอาศัยข้อมูลโดยรอบจำนวนต่างๆ กัน k ตัว

	$c=0.1$	$c=0.2$	$c=0.3$	$c=0.4$	$c=0.5$	$c=1$
$k=3$	86.63	**86.97**	86.97	86.90	86.78	86.24
$k=5$	86.46	86.88	87.00	**87.03**	86.93	86.44
$k=10$	86.16	86.75	86.91	86.98	**87.01**	86.65

โดยจากการทดลองเราค้นพบว่าช่วงของค่า c ที่เหมาะสมที่สุดน่าจะอยู่ในช่วง [0.1,0.5] โดยกรณีที่มีค่าความถูกต้องสูงที่สุดในการทดลองนี้คือ $c=0.4$ เมื่อพิจารณาข้อมูลโดยรอบ 5 ตัว ซึ่งมีค่าความถูกต้อง 87.03% ซึ่งให้ค่าความถูกต้องสูงกว่ากรณีที่ใช้ฟังก์ชัน $f(x)=\frac{1}{x''}$ ซึ่งมีค่าความถูกต้อง 86.28% และนับว่าสูงที่สุดจากผลการทดลองที่ผ่านมา และนอกจากนี้เราจะสามารถสังเกตจากผลการทดลองได้ว่าเมื่อเราทำการเปลี่ยนจำนวนข้อมูลโดยรอบที่ใช้คำนวณค่า $h(d)$ ถึงแม้ว่าค่า c ที่ให้ค่าความถูกต้องสูงสุดจะเปลี่ยนไปบ้างแต่ค่าความถูกต้องสูงสุดที่ได้จะไม่แตกต่างกันมากนัก

ตารางที่ 31 เปอร์เซ็นต์ความถูกต้องจากฟังก์ชันรูปแบบที่ 2 เมื่อใช้ $\sqrt[n]{h_3(d)}$ โดยอาศัยข้อมูลโดยรอบ 3 ตัว

Dataset	$c = 0.2$	$c = 0.3$	$c = 0.4$	$c = 0.5$	$c = 1$	$c = 2$
glass	67.76	67.76	68.23	68.69	**69.63**	68.24
image	92.52	92.57	92.48	92.57	**92.62**	92.19
ionosphere	87.18	86.60	86.89	86.89	88.60	**89.16**
iris	**95.33**	**95.33**	**95.33**	**95.33**	**95.33**	**95.33**
liver	62.03	61.74	**62.61**	**62.61**	62.32	61.74
scale	**89.92**	**89.92**	**89.92**	**89.92**	89.76	88.64
tictactoe	98.54	98.64	98.64	98.64	98.95	**99.16**
wine	96.10	96.10	96.10	96.10	96.10	**97.21**
zoo	**94.00**	**94.00**	**94.00**	**94.00**	**94.00**	**94.00**
genpure	95.00	95.40	95.60	95.60	**95.80**	95.60
gennoise10%	85.80	**86.20**	85.60	84.60	83.60	81.00
gennoise20%	**76.00**	75.60	75.80	75.20	71.20	67.40
gennoise30%	**60.80**	**60.80**	60.00	58.80	53.20	49.00
Total	87.03	**87.04**	87.01	86.85	86.26	85.27

ตารางที่ 32 เปอร์เซ็นต์ความถูกต้องจากฟังก์ชันรูปแบบที่ 2 เมื่อใช้ $\sqrt[n]{h_3(d)}$ โดยอาศัยข้อมูลโดยรอบ 5 ตัว

Dataset	$c = 0.2$	$c = 0.3$	$c = 0.4$	$c = 0.5$	$c = 1$	$c = 2$
glass	**68.24**	67.76	67.76	68.23	67.76	70.10
image	92.52	92.48	92.52	92.43	**92.62**	92.38
ionosphere	86.89	87.18	86.60	86.89	88.60	**89.73**
iris	**95.33**	**95.33**	**95.33**	**95.33**	**95.33**	**95.33**
liver	61.74	62.03	**62.90**	**62.90**	61.45	**62.90**
scale	**89.92**	**89.92**	**89.92**	**89.92**	89.76	88.00
tictactoe	98.54	98.64	98.64	98.64	98.95	**99.16**
wine	96.10	96.10	96.10	96.10	96.10	**96.65**
zoo	**94.00**	**94.00**	**94.00**	**94.00**	**94.00**	**94.00**
genpure	95.20	95.60	95.60	95.60	95.60	**95.80**
gennoise10%	85.20	85.60	**86.00**	85.80	84.20	81.40
gennoise20%	75.20	**76.00**	**76.00**	75.40	73.00	68.40
gennoise30%	**61.20**	60.80	60.40	59.80	55.00	49.60
Total	86.95	87.05	**87.08**	86.98	86.44	85.56

ตารางที่ 33 เปอร์เซ็นต์ความถูกต้องจากฟังก์ชันรูปแบบที่ 2 เมื่อใช้ $\sqrt[n]{h_3(d)}$ โดยอาศัยข้อมูลโดยรอบ 10 ตัว

Dataset	$c = 0.2$	$c = 0.3$	$c = 0.4$	$c = 0.5$	$c = 1$	$c = 2$
glass	68.24	68.24	68.24	68.24	68.23	**70.10**
image	**92.52**	**92.52**	92.43	92.43	92.48	92.48
ionosphere	87.18	87.46	87.74	87.74	88.03	**89.45**
iris	**95.33**	**95.33**	**95.33**	**95.33**	**95.33**	**95.33**
liver	62.03	61.74	62.03	62.32	62.32	**63.48**
scale	**89.92**	**89.92**	**89.92**	**89.92**	89.76	88.64
tictactoe	98.64	98.64	98.64	98.75	98.95	**99.16**
wine	**96.10**	**96.10**	**96.10**	**96.10**	**96.10**	**96.10**
zoo	**94.00**	**94.00**	**94.00**	**94.00**	**94.00**	**94.00**
genpure	95.00	95.20	95.40	95.40	**95.80**	**95.80**
gennoise10%	84.80	85.20	85.80	**86.00**	84.20	83.00
gennoise20%	75.20	**75.80**	**75.80**	75.60	74.40	70.20
gennoise30%	**61.20**	**61.20**	60.60	60.60	57.60	52.60
Total	86.95	87.04	87.05	**87.08**	86.73	86.10

ตารางที่ 34 สรุปเปอร์เซ็นต์ความถูกต้องจากฟังก์ชันรูปแบบที่ 2 เมื่อใช้ $\overline{\sqrt[n]{h_3(d)}}$ โดยอาศัยข้อมูลโดยรอบจำนวนต่างๆ กัน k ตัว

	$c=0.2$	$c=0.3$	$c=0.4$	$c=0.5$	$c=1$	$c=2$
$k=3$	87.03	**87.04**	87.01	86.85	86.26	85.27
$k=5$	86.95	87.05	**87.08**	86.98	86.44	85.56
$k=10$	86.95	87.04	87.05	**87.08**	86.73	86.10

ในการประมาณการกระจายตัวแบบสม่ำเสมอของข้อมูลโดยอาศัยฟังก์ชันในรูปแบบที่สอง เราจะเห็นได้ว่าความแม่นยำของการจำแนกประเภทจะเพิ่มขึ้นเพียงเล็กน้อยเมื่อเทียบกับการใช้ค่าเฉลี่ยของค่า $\sqrt[n]{h(d)}$ ค่า c ที่เหมาะสมน่าจะอยู่ในช่วง [0.1,0.5] เช่นกัน

อย่างไรก็ตามจะสังเกตเห็นได้ว่าสำหรับข้อมูลเกือบทุกชุดนอกจากชุดที่สร้างขึ้นมาแล้วทำให้มีข้อผิดพลาดนั้น ค่าปรับแต่งของฟังก์ชันที่ให้ค่าความถูกต้องสูงที่สุดส่วนใหญ่จะมาจากกรณีที่ $c=1$ หรือ $c=2$ โดยหากคิดค่าเฉลี่ยของค่าความถูกต้องเฉพาะชุดข้อมูลที่นำมาจาก UCI Machine Learning Repository เราจะได้ว่าจากในตารางทั้ง 3 นั้น ในตารางที่ 31 จะมีค่าความถูกต้องสูงที่สุดเมื่อ $c=1$ และในตารางที่ 32 และ 33 จะมีค่าความถูกต้องสูงที่สุดเมื่อ $c=2$ ซึ่งจากผลการทดลองของฟังก์ชันรูปแบบที่สองที่ไม่ใช้การประมาณการกระจายตัวแบบสม่ำเสมอของข้อมูลก็พบลักษณะที่ว่านี้เช่นกัน โดยสาเหตุของปรากฏการณ์นี้อาจมาจากการที่ค่าของ c ที่เหมาะสมจะเปลี่ยนไปตามจำนวนคุณสมบัติของข้อมูลด้วย

ตารางที่ 35 เปอร์เซ็นต์ความถูกต้องที่ได้จากฟังก์ชันถ่วงน้ำหนักในรูปแบบที่ 3

Dataset	$t = 0.2$	$t = 0.5$	$t = 1$	$t = 2$	$t = 5$	$t = 10$
Glass	62.62	**69.66**	68.24	68.69	68.69	67.76
image	91.67	**93.05**	92.52	92.38	92.48	92.48
ionosphere	70.09	81.47	86.32	**87.46**	86.61	86.61
iris	**96.00**	95.33	95.33	**96.00**	**96.00**	**96.00**
liver	59.71	**64.64**	63.19	60.29	60.00	59.71
scale	88.32	**90.40**	89.92	87.04	86.72	86.72
tictactoe	65.35	89.66	98.64	99.90	**100.00**	**100.00**
wine	**97.76**	96.65	96.10	95.54	95.54	95.54
zoo	92.00	**94.00**	**94.00**	**94.00**	**94.00**	**94.00**
genpure	51.60	83.80	95.00	**95.60**	95.00	95.20
gennoise10%	46.60	77.00	**83.40**	80.20	77.80	77.60
gennoise20%	46.40	63.80	**73.00**	66.20	62.80	62.40
gennoise30%	41.00	**55.20**	**55.20**	49.80	47.20	47.00
Total	71.60	83.24	**86.28**	85.03	84.35	84.26

สำหรับฟังก์ชันในรูปแบบที่ 3 ซึ่งเขียนเป็นสมการได้ว่า $f(x) = \frac{1}{x^{tn}}; t \geq 1$ จากการทดลองพบว่าค่าปรับแต่งที่ให้ค่าความถูกต้องสูงที่สุดก็คือ $t = 1$ ซึ่งเมื่อทำการแทนค่าลงไปในสมการของฟังก์ชัน $f(x) = \frac{1}{x^{tn}}; t \geq 1$ ก็จะได้ว่า $f(x) = \frac{1}{x^{n}}$ ซึ่งก็ตรงกับกรณีเฉพาะกรณีหนึ่งของฟังก์ชันในรูปแบบที่ 1 นั่นก็คือ $f(x) = \frac{1}{x^{t}}; t \geq n-1$ เมื่อ $t = n$ ซึ่งเป็นกรณีเดียวที่ได้รับการสนับสนุนทางทฤษฎีจากฟังก์ชันทั้ง 2 รูปแบบ