Algoritmo KNN basado en Información Mutua para Clasificación de Patrones con Valores Perdidos

Pedro J. García-Laencina, Rafael Verdú-Monedero, Jorge Larrey-Ruiz, Juan Morales-Sánchez, José-Luis Sancho-Gómez
e-mail: {pedroj.garcia,rafael.verdu,jorge.larrey,juan.morales,josel.sancho}@upct.es
Dpto. Tecnologías de la Información y las Comunicaciones.
Universidad Politécnica de Cartagena.
Plaza del Hospital, 1. 30202 Cartagena (Murcia), Spain.

***Abstract*— Incomplete data is a common drawback in real-life classification problems. Missing values in data sets may have different origins such as death of patients, equipment malfunctions, refusal of respondents to answer certain questions, and so on. This work[1] presents an effective and robust approach for classification with unknown input data. In particular, an enhanced version of the K-Nearest Neighbours algorithm using Mutual Information is proposed. Results on two classification datasets shows the usefullness of this approach.**

I. Introducción

La presencia de datos incompletos es un inconveniente muy común en aplicaciones reales de clasificación estadística de patrones[2]. Hay infinidad de posibles orígenes para la ausencia de datos en los vectores de entrada. Como ejemplos, en una red de sensores algunos datos pueden estar incompletos debido a fallos mecánicos y/o electrónicos durante el proceso de adquisición; en un problema de diagnosis médica algunas pruebas no son posibles de realizar debido a que no son apropiadas para determinados pacientes o bien el hospital carece del equipamiento médico necesario. El correcto tratamiento de los valores perdidos es una etapa fundamental, ya que en caso contrario se pueden cometer grandes errores o falsos resultados durante el procesamiento de los mismos. Uno de los procedimientos más recomendados es la imputación de datos. Se entiende por imputación al proceso de estimar y rellenar valores perdidos usando toda la información disponible.

Este artículo presenta un método robusto para imputación y clasificación de patrones con valores perdidos. En concreto, se propone un versión mejorada del conocido algoritmo de los K vecinos más cercanos (K-Nearest Neighbours, KNN) basada en la Información Mutua (IM). Este método encuentra los K vecinos más cercanos mediante una eficiente métrica de distancia basada en la MI que considera la relevancia de cada característica de entrada para el problema de clasificación. Dicha distancia es utilizada tanto para imputación como para clasificación. En el caso de imputación de datos, el método propuesto obtiene una imputación orientada y dirigida a resolver la tarea de clasificación. Por otro lado, el clasificador KNN basado en la IM es un procedimiento robusto que permite mejorar las prestaciones en terminos de clasificación. El resto del artículo se estructura de la siguiente forma: en la Sección II, se presenta la notación empleada; la Sección III analiza el algoritmo estandar KNN para clasificación e imputación de patrones incompletos; mientras que la Sección IV describe el método propuesto para imputación usando una médida de distancia basada en la IM; la Sección V se presenta el algoritmo KNN para clasificación, y su versión extendida basada en la IM; la Sección VI muestra los resultados obtenidos en dos problemas de clasificación; y finalmente, se exponen las conclusiones principales y futuros trabajos.

[1]Este trabajo está parcialmente financiado por el Ministerio de Educación y Ciencia a través del proyecto TEC2006-13338/TCM.

[2]Los términos patrón, vector de entrada y caso son usados como sinónimos.

II. Clasificación de Patrones Incompletos

Un problema de clasificación consiste en asignar una determinada clase a cada patrón a partir del vector de características asociado al mismo. Considerar un problema de clasificación $\mathcal{D}$ caracterizado por un conjunto de N patrones,

$$\mathcal{D} = \{(\mathbf{x}^i, \mathbf{m}^i, t^i)\}_{i=1}^{N}$$

donde $\mathbf{x}^i$ es el i-ésimo vector de entrada compuesto por n características ($\mathbf{x}^i = \{x_j^i\}_{j=1}^n$), $\mathbf{m}^i$ es un vector de variables binarias tal que m_j^i es igual a 1 si x_j^i está incompleto o 0 en caso contrario; y t^i es la etiqueta de la clase a la que pertenece dicho patrón. Este trabajo asume que los valores perdidos son del tipo MCAR (Missing Completely At Random), i.e., la probabilidad de que un patrón presente un valor perdido en una característica no depende ni del resto de características ni de los valores de la propia característica incompleta [1].

En general, la clasificación de patrones incompletos conlleva la resolución de dos problemas distintos: tratamiento de valores perdidos y clasificación de patrones. Este artículo está centrado en soluciones basadas en imputación de datos, cuyo objetivo es completar la información desconocida (valores perdidos) con estimaciones obtenidas a partir del conjunto de datos conocidos. Una vez todos los valores perdidos de los patrones incompletos han sido imputados, una máquina de decisión es entrenada para resolver el problema de clasificación.

III. Imputación mediante Algoritmo KNN

Este artículo usa el algoritmo KNN para imputar los valores perdidos de las características incompletas (KNNimpute) [2],

[3]. La mayor ventaja de este método es la gran versatilidad que presenta para manejar patrones con múltiples valores perdidos. Además, KNNimpute puede estimar eficientemente tanto características numéricas como categóricas. Como desventaja destaca el hecho de que el algoritmo KNN busca los patrones más cercanos en todo el conjunto de datos. Esta limitación puede ser crítica en grandes bases de datos.

A. Algoritmo KNNimpute

Dado un patrón incompleto $\mathbf{x}$, el algoritmo KNNimpute selecciona los K_I casos más cercanos del conjunto de entrenamiento con valores conocidos en las características a ser imputadas (i.e., características con valores perdidos en $\mathbf{x}$), de tal forma que se minimize una métrica de distancia. Se usa K_I para referir a los K vecinos más cercanos usados para imputación. Aunque los K_I casos más cercanos pueden tambien seleccionarse de los patrones sin ningún valor perdido, es recomendable usar la anterior opción [3]. El valor óptimo de K_I puede ser obtenido mediante validación cruzada [4].

Una vez los K_I vecinos más cercanos han sido obtenidos, un valor para substituir el dato incompleto es estimado. La manera de calcular dicho valor depende del tipo de característica de entrada, por ejemplo, la moda es usada para variables discretas (características categóricas con un número discreto de posibles valores), mientras que la media es comunmente empleada para variables continuas (características numéricas con un rango continuo de posibles valores). KNNimpute proporciona un robusto procedimiento para la estimación de valores perdidos [2], [3].

En este trabajo, se ha usado una métrica de distancia heterogenea (Heterogeneous Euclidean-Overlap Metric, HEOM) [3], que permite manejar datos incompletos tanto en características numéricas como categóricas. La distancia HEOM entre dos patrones $\mathbf{x}^a$ y $\mathbf{x}^b$ viene dada por,

$$d_H(\mathbf{x}^a, \mathbf{x}^b) = \sqrt{\sum_{j=1}^{n} d_j(x_j^a, x_j^b)^2}, \tag{1}$$

siendo $d_j(x_j^a, x_j^b)$ la distancia entre $\mathbf{x}^a$ y $\mathbf{x}^b$ evaluada en la j-ésima característica de entrada:

$$d_j(x_j^a, x_j^b) = \begin{cases} 1, & (1-m_j^a)(1-m_j^b) = 0 \\ d_O(x_j^a, x_j^b), & x_j \text{ es una variable discreta} \\ d_N(x_j^a, x_j^b), & x_j \text{ es una variable continua.} \end{cases} \tag{2}$$

Cuando existen valores perdidos, se asigna un valor distancia igual a 1 (i.e. distancia máxima). La distancia "overlap" d_O asigna un valor de distancia igual a 0 si las características categóricas x_j^a y x_j^a son iguales, en caso contrario si asigna una distancia igual a 1. La distancia normalizada d_N viene dada por

$$d_N(x_j^a, x_j^b) = \frac{\left|x_j^a - x_j^b\right|}{max(x_j) - min(x_j)}, \tag{3}$$

donde $max(x_j)$ and $min(x_j)$ son respectivamente los valores máximo y mínimo observados en el conjunto de entrenamiento para la característica numérica x_j.

IV. Imputación mediante Algoritmo KNN basado en Información Mutua

Evaluar la importancia/relevancia de las características de entrada antes de construir un modelo de decisión o regresión es un procedimiento muy útil que puede mejorar sensiblemente las prestaciones finales del modelo. Una forma de medir la relevancia de cada característica de entrada para un problema de clasificación o aproximación es calcular la Información Mutua (IM) entre cada característica y la tarea objetivo [5], [6]. Considerar un problema de clasificación, donde α_i representa la IM medida entre la j-ésima característica de entrada (x_j) y la salida deseada de clasificación (t), i.e., $\alpha_j = I(x_j, t)$. Cuanto mayor sea el valor de α_j, más relevante es x_j para resolver la tarea de clasificación.

La IM es una medida no-paramétrica y no-lineal de relevancia derivada de la teoría de la información [5]. La IM entre dos variables aletorias Z_1 y Z_2, $I(Z_1, Z_2)$, es una medida de como Z_1 depende de Z_2 y viceversa, y se puede definir a partir del concepto de entropía $H(\cdot)$ [5]:

$$\begin{aligned} I(Z_1, Z_2) &= H(Z_1) + H(Z_2) - H(Z_1, Z_2) = \\ &= H(Z_2) - H(Z_2|Z_1), \end{aligned} \tag{4}$$

donde $H(Z_2|Z_1)$ es la entropia condicionada de Z_2 dado Z_1, y representa la incertidumbre que se tiene sobre Z_2 cuando se conoce el valor de Z_1 [5]. $I(Z_1, Z_2)$ mide la reducción en la incertidumbre en Z_1 cuando se conoce el valor de Z_1 y viceversa. Si las variables Z_1 y Z_2 son independientes, $H(Z_1, Z_2) = H(Z_1) + H(Z_2)$, y $H(Z_2|Z_1) = H(Z_2)$, i.e., la IM entre dos variables independientes es cero. Cuanto mayor sea la medida de IM entre dos variables, mayor será la dependencia existente entre ambas.

A. Algoritmo MI-KNNimpute

Siguiendo el concepto de relevancia de características para clasificación basado en la IM, un eficiente método de imputación mediante KNN es propuesto usando una métrica de distancia basada en la IM. El método propuesto es conocido como MI-KNNimpute. La métrica de distancia basada en la IM entre dos patrones $\mathbf{x}^a$ y $\mathbf{x}^b$ viene dada por

$$d_I(\mathbf{x}^a, \mathbf{x}^b) = \sqrt{\sum_{j=1}^{n} \frac{\alpha_j d_j(x_j^a, x_j^b)^2}{\sum_{j'=1}^{n} \alpha_{j'}}}, \tag{5}$$

donde d_j es la distancia heterogenea definida en (2), y $\alpha_j = I(x_j, t)$. La estimación de la IM empleada en este trabajo se ha obtenido mediante un procedimiento basado en ventanas de Parzen [6], considerando únicamente el conjunto de casos completos en el atributo de interés. Al contrario que el algoritmo estándar KNNimpute, el método presentado selecciona los K_I vecinos más cercanos considerando la importancia de las características de entrada para la clasificación. Estos K_I casos son conocidos como los *vecinos más relevantes-cercanos*. Por tanto, MI-KNNimpute proporciona una estimación de valores perdidos orientada a resolver la tarea de clasificación.

Considerar que la j-ésima característica del i-ésimo vector de entrada (x_j^i) es desconocida (i.e., $m_j^i = 1$) y x_j es una

variable continua. En este caso, el valor imputado $\tilde{x}_j^i$ es calculado por una media ponderada,

$$\tilde{x}_j^i = \frac{\sum_{k=1}^{K_I} \frac{v_j^k}{d_I(\mathbf{x}^i,\mathbf{v}^k)^2}}{\sum_{k=1}^{K_I} \frac{1}{d_I(\mathbf{x}^i,\mathbf{v}^k)^2}} \tag{6}$$

donde $\mathbf{v}^1$, $\mathbf{v}^2$, ..., $\mathbf{v}^{K_I}$ son los K_I vecinos más relevantes-cercanos de $\mathbf{x}^i$ ordenados crecientemente según la distancia $d_I(\mathbf{x}^i, \mathbf{v}^k)$. Así $\mathbf{v}^1$ es el vecino más relevante-cercano de todos los K_I vecinos. De esta forma, los vecinos más cercanos (i.e. menor distancia) contribuyen en mayor medida al valor estimado para la imputación. En el caso de que x_j sea una variable discreta, el valor imputado $\tilde{x}_j^i$ es la moda de $\{v_j^k\}_{k=1}^{K_I}$.

V. Clasificación mediante Algoritmo KNN

La idea básica sobre la que se fundamenta un clasificador KNN (KNNclassify) es que un nuevo patrón $\mathbf{x}$ se va a asignar a la clase más frecuente a la que pertenecen sus K_C vecinos más cercanos. Se usa K_C para referir a los K vecinos más cercanos usados para clasificación. En caso de que se produzca un empate entre dos o más clases, conviene tener una regla heurística para su ruptura como son seleccionar la clase que contenta al vecino más próximo, seleccionar la clase con distancia media menor, etc. En la versión estándar de un clasificador KNN, los K_C vecinos más cercanos tienen implícitamente igual importancia en la decisión, sin considerar las respectivas distancias a $\mathbf{x}$. De esta forma, también es conveniente considerar una ponderación de cada uno de los K_C vecinos [7], de tal forma que se tenga en cuenta la distancia de cada vecino a $\mathbf{x}$. En particular, se ha ponderado cada vecino de manera inversamente proporcional al cuadrado de la distancia del mismo a $\mathbf{x}$, y se asigna a aquella clase cuya suma de ponderaciones sea menor [7].

Siguiendo un enfoque basado en el uso de la IM para ponderar la métrica de distancia, se presenta un clasificador KNN basado en la IM (MI-KNNclassify). Considerar que $d_I(\cdot)$ es la métrica de distancia definida en (5), y $\{\mathbf{v}^k\}_{k=1}^{K_C}$ son los K_C vecinos más relevantes-cercanos según un orden creciente de las correspondientes distancias. MI-KNNclassify asigna a cada vecino $\mathbf{v}^k$, un peso $\beta_k(\mathbf{x})$ dado por,

$$\beta_k(\mathbf{x}) = \begin{cases} \frac{d_I(\mathbf{v}^{K_C},\mathbf{x}) - d_I(\mathbf{v}^k,\mathbf{x})}{d_I(\mathbf{v}^{K_C},\mathbf{x}) - d_I(\mathbf{v}^1,\mathbf{x})}, & \text{si } d_I(\mathbf{v}^{K_C},\mathbf{x}) \neq d_I(\mathbf{v}^1,\mathbf{x}) \\ 1, & \text{otro caso.} \end{cases} \tag{7}$$

Así, β_k es calculada teniendo en cuenta la relevancia de las características de entrada para la tarea de clasificación, y $\mathbf{x}$ es asignado a aquella clase que cuyos pesos sumen un mayor valor. Además de esto, MI-KNNclassify es robusto a la presencia de características de entrada irrelevantes.

VI. Resultados Experimentales

Con el objetivo de comparar el método propuesto con el procedimiento KNN estándar, se ha escogido un conjunto de datos completo, *Telugu*[3], y un conjunto incompleto, *Hepatitis*[4].

[3]http://www.isical.ac.in/∼sushmita/patterns/

[4]http://http://archive.ics.uci.edu/ml/

La principal razón de usar un problema de clasificación completo es medir la influencia de insertar artificialmente distintos porcentajes de valores perdidos sobre la tasa de acierto de clasificación. Los resultados experimentales de los distintos métodos usados en este trabajo han sido obtenidos empleando los mismos conjuntos de entrenamiento, validación y test. Dichos conjuntos han sido obtenidos por el método de validación cruzada estratificada con 10 particiones [4]. Para cada partición del problema completo *Telugu*, se inserta un porcentaje de valores perdidos (en este trabajo se ha considerado: 5 %, 10 %, 20 %, 30 % y 40 %) en las características seleccionadas de una manera completamente aleatoria [1]. Así, cada partición presenta un mismo número de valores perdidos. En ambos escenarios (*Telugu* y *Hepatitis*), primero, el conjunto de test es clasificado por KNNclassify y MI-KNNclassify sin imputación de valores perdidos. A continuación, se evalua la influencia sobre la tasa de acierto de clasificación de emplear una imputación previa mediante KNNimpute y MI-KNNimpute. Los valores óptimos de K para imputación (K_I) y para clasificación (K_C) se escogen por validación cruzada.

A. Problema Telugu

Telugu es un problema de reconocimiento de las seis vocales de la lengua india telugú. Este conjunto de datos consta de 871 casos con tres atributos reales (tres primeras frecuencias formantes). Antes de insertar artificialmente valores perdidos, se ha medido la IM entre cada característica y la tarea de clasificación: $\alpha_1 = 1{,}043$, $\alpha_2 = 1{,}381$, y $\alpha_3 = 0{,}829$. Los valores perdidos son insertados aleatoriamente en las características x_1 y x_2, ya que son las más relevantes para la clasificación (mayores valores de α_j) y por tanto la presencia de valores perdidos en estas variables afectará en mayor medida a la tasa de acierto. La Tabla I muestra la tasa de acierto obtenida sin realizar una imputación previa. Se han probado los siguientes valores para K_C y K_I: 1, 5, 10, 15, 20, 30, 40 y 50. Para cada porcentaje de valores perdidos, K_C ha sido escogido por validación cruzada. En la Tabla I podemos ver como MI-KNNclassify proporciona mejores resultados que el clasificador KNN en todos los experimentos.

	Valores perdidos (%) en x_1 y x_2				
	5 %	10 %	20 %	30 %	40 %
KNNclassify	83.69	82.32	76.11	72.77	68.98
MI-KNNclassify	84.61	83.23	78.30	75.53	70.00

TABLA I

TASA DE ACIERTO (%) EN EL CONJUNTO DE TEST PARA EL PROBLEMA *Telugu* USANDO KNNCLASSIFY Y MI-KNNCLASSIFY (SIN IMPUTACIÓN PREVIA EN x_1 Y x_2).

A continuación, se analiza si una imputación de valores perdidos (con KNNimpute y MI-KNNimpute) proporciona mejores resultados de clasificación. Los distintos valores de K_I y K_C han sido escogidos por validación cruzada (usando la tasa de acierto obtenida en el conjunto de validación). Como se puede comprobar en la Tabla II, una imputación de valores perdidos ayuda a mejorar las prestaciones del clasificador en todas las simulaciones. Para bajos porcentajes

de valores perdidos (desde $5\,\%$ hasta $40\,\%$), el procedimiento conjunto MI-KNNclassify y MI-KNNimpute mejora el resto de alternativas. La ventaja de usar MI-KNNimpute no es tan evidente para porcentajes elevados de valores perdidos, como podemos ver con un $40\,\%$. En este caso, KNNimpute es una mejor solución para imputación. Esto es debido a que el valor estimado de la IM no es preciso (se tiene mucha menos información para calcular los valores de IM, i.e., α_j).

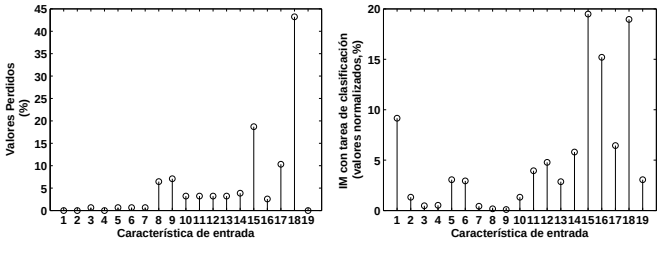
Valores Perdidos	KNNclassify		MI-KNNclassify	
	KNNimpute	MI-KNNimpute	KNNimpute	MI-KNNimpute
5 %	85.29	86.09	85.53	86.21
10 %	83.78	84.25	84.38	84.60
20 %	78.85	79.02	78.76	79.23
30 %	75.18	75.51	75.64	75.73
40 %	69.91	69.60	69.99	69.69

TABLA II

TASA DE ACIERTO (%) EN EL CONJUNTO DE TEST PARA EL PROBLEMA *Telugu* USANDO KNNCLASSIFY Y MI-KNNCLASSIFY TRAS IMPUTAR LOS VALORES PERDIDOS EN x_1 Y x_2.

B. Problema Hepatitis

El objetivo de este problema es determinar la supervivencia de un paciente (i.e., dos posibles clases: vida o muerte) tras la diagnosis de hepatitis. El conjunto de datos está compuesto por 155 pacientes, donde cada uno de ellos está asociado a un vector de 19 características (edad, sexo, niveles de fostatasa y albumina, etc). Existen 32 casos de pacientes que fallecieron de hepatitis, y los restantes consiguieron superar la enfermedad. Este problema presenta 80 casos con valores perdidos, siendo incompletos un 5.7 % de los datos de entrada. Las Figuras 1(a) y 1(b) muestran respectivamente los porcentajes de valores perdidos y valor normalizado de IM con respecto a la tarea de clasificación para cada característica de entrada. Analizando ambas gráficas, podemos ver como las dos



(a) (b)

Fig. 1. Primero, en (a) se muestran los porcentajes de valores perdidos en cada característica. Mientras, (b) muestra los valores de IM normalizados (en %) entre cada característica de entrada y la tarea de clasificación.

características más relevantes (x_{15} y x_{18}) son las que presentan un mayor porcentaje de valores perdidos.

La Tabla III muestra las tasas de acierto (%) en el conjunto de test, sin realizar una imputación previa, obtenidas mediante KNNclassify y MI-KNNclassify para diferentes valores de K_C. En este problema, se han probado los siguientes valores para K_I y K_C: 1, 5, 10, 15, y 20. Los valores seleccionados por validación cruzada se muestran en negrita. Como puede observarse en la Tabla III, MI-KNNclassify mejora las prestaciones obtenidas mediante KNNclassify.

	Número de vecinos más cercanos (K_C)				
	1	5	10	15	20
KNNclassify	80.09	81.39	**80.02**	81.28	81.28
MI-KNNclassify	76.06	79.24	83.12	**83.20**	81.99

TABLA III

TASA DE ACIERTO (%) EN EL CONJUNTO DE TEST PARA EL PROBLEMA *Hepatitis* SIN IMPUTACIÓN VS. NÚMERO DE VECINOS MÁS CERCANOS (K_C) EN KNNCLASSIFY Y MI-KNNCLASSIFY.

A continuación, los valores perdidos son estimados mediante KNNimpute y MI-KNNimpute. Las tasas de acierto en el conjunto de test tras la imputación de datos incompletos son los siguientes: **84.56** (KNNclassify + KNNimpute), **85.08** (KNNclassify + MI-KNNimpute), **85.70** (MI-KNNclassify + KNNimpute) y **86.25** (MI-KNNclassify + MI-KNNimpute). A tenor de estos resultados, es evidente comprobar que la imputación usando KNN basado en la IM ayuda a mejorar las prestaciones en términos de clasificación.

VII. CONCLUSIONES Y TRABAJOS FUTUROS

Este artículo presenta un nuevo algoritmo KNN (K Nearest Neighbours) para clasificación e imputación de patrones incompletos usando el concepto de la Información Mutua (IM). El método propuesto selecciona los K vecinos más cercanos considerando la relevancia de las características de entrada para resolver la tarea de clasificación. Para ello se ha propuesto una métrica de distancia basada en la IM. Durante la etapa de imputación de valores perdidos, este método proporciona una imputación orientada resolver la tarea de clasificación. Además, se presenta un eficiente clasificador KNN basado en la métrica de distancia ponderada por la IM. Los resultados experimentales obtenidos en dos problemas de clasificación muestran las ventajas del procedimiento propuesto.

Los trabajos futuros son implementar un estimador robusto de la IM para valores perdidos y realizar un estudio comparativo con otros métodos computacionales de imputación.

REFERENCIAS

[1] R. J. A. Little and D. B. Rubin, *Statistical Analysis with Missing Data*, 2nd ed., New Jersey, USA: John Wiley & Sons, 2002.

[2] O. Troyanskaya, M. Cantor, O. Alter, G. Sherlock, P. Brown, D. Botstein, R. Tibshirani, T. Hastie, and R. Altman, "Missing value estimation methods for DNA microarrays", *Bioinformatics*, vol. 17, no.6, pp. 520-525, 2001.

[3] G. E. Batista and M. C. Monard, "An analysis of four missing data treatment methods for supervised learning", *Applied Artificial Intelligence*, vol. 17 no. 5, pp.519-533, 2003.

[4] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection". *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pp. 1137-1143, 1995.

[5] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed., New Jersey, USA: Wiley-Interscience, 2006.

[6] N. Kwak and C.-H. Choi, "Input feature selection by mutual information based on Parzen window", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 12, pp. 1667-1671, 2002.

[7] A. L. Barker, *Selection of distance metrics and feature subsets for k-nearest neighbor classifiers*, Ph. D. Thesis, Univ. of Virginia, USA, 1997.