AA-52 - MWSUG 2025

Feature Selection and Classification Using Fuzzy Logic

Ross Bettinger

Abstract

We investigate the use of fuzzy logic as applied to feature selection and classification. Fuzzy logic, a generalization of Aristotelian logic, can be useful in situations where there is imprecision or vagueness in the problem domain. Fuzzy logic is applied to transform input data into fuzzy sets that are then suitable for processing by a feature selection algorithm. A fuzzy entropy measure is used to perform classification using a similarity classifier. SAS/IML® was used to perform all computations.

Keywords

Classification, crisp set, data dimension, feature selection, fuzzy entropy, fuzzy logic, fuzzy membership, fuzzy set, fuzzy variable, Łukasiewicz structure, membership function, similarity classifier

Introduction

Model building typically progresses through several distinct phases.[1] Feature selection is usually performed during the exploratory data analysis phase of model building. The workings of a model with a large number of features may be difficult to explain to clients who need to use a model's results, while a model with fewer features but equal accuracy is likely to be more readily accepted by analysts and decision-makers. Fewer features may reduce collinearity between independent features and may produce more accurate predictions. Models with big data and fewer features will be less costly to build and maintain than models with high numbers of features.

Fuzzy logic is a methodology for extending a collection of disjoint sets into a collection of overlapping sets by defining information as grades of membership in a set. In crisp set theory, an object in a set is either a member of the set or it is not [1]. For a crisp set A and set element x, either $x \in A$ or $x \notin A$. The grade of membership is represented by the binary set {0, 1}. In a fuzzy set A, the grade of membership lies in the unit interval [0, 1]. Also, the concept of information entropy [2], defined for probabilities, has been extended by reinterpreting the grade of membership into a fuzzy number [3]. In this paper, we discuss using fuzzy logic and fuzzy entropy measures as a toolset with which to perform feature selection. We can also classify observations into fuzzy classes based on the similarity of an observation to a set of "ideal" vectors.

Some Concepts of Fuzzy Logic

In 1965, Lotfi Zadeh, a professor of mathematics at the University of California, Berkeley, published a paper in which he introduced the term *fuzzy logic* to characterize his concept of *fuzzy set theory* as a logical calculus with which to express in mathematical terms the vagueness and imprecision of statements that cannot be formulated as binary propositions. His work is derived from earlier results in the 1920's by Łukasiewicz and Tarski. Describing an event as both "partially true" and "partially false" will allow us to *fuzzify* an event and think about it in terms of a continuum of truth values instead of as binary states.

[1] The SAS® Institute has developed the SEMMA methodology (Sample, Explore, Modify, Model, Assess) to describe each phase of activity.

Crisp logic can be used to represent only those mathematical propositions (statements) that can be evaluated to one of the predicate values TRUE or FALSE, i.e., the binary numeric values 1 or 0. The statement "It will rain tomorrow" cannot be posed as a crisp logical proposition since it cannot be meaningfully evaluated to TRUE or FALSE.

For example, a host at a dinner party tells the invitees that guests will be seated at 6 PM. Some guests will arrive at 5:50 PM, others around 6 PM, others at 6:15 PM. We can define the intervals as "Early", "On-Time" and "Late" as "between 5:45 PM and 5:55 PM", "5:56 PM and 6:05 PM", and "6:06 PM and 6:15 PM" and evaluate the crisp proposition that "Guest1 arrived early" as TRUE if Guest1 arrived at 5:55 PM. But realistically, how different is that from Guest1 arriving at 5:56 PM? Is Guest1 100% early or 100% on-time? By the law of the excluded middle[2], there are no in-between values. However, isn't it somewhat arbitrary to classify Guest1 as early when he is *almost* on-time? The fuzzy logic formulation would allow both "Early" and "On-Time" as acceptable membership values in the fuzzy set whose objects x are {"Early", "On-Time", "Late"}. We will discuss how fuzzy grades of membership are computed *infra*.

Crisp Sets

Let X represent the universe of discourse such that X contains all possible elements (objects) of interest in a particular context or investigation. Given that X is a set of elements with a generic element in X denoted by x, we have $X = \{x_i\}_{i=1}^{N}$. A *crisp set* in Aristotelian set theory is a set X whose elements have membership grades (values) of either 1 or 0 to signify that an element either belongs or does not belong in X. The *characteristic function* of a set m: $X \to \{0, 1\}$ computes for each $x \in X$ the value $m(x)$ which is called the *grade of membership* of $x \in X$.

In the dinner party example *supra*, the proposition $p =$ "Guest1 is on-time" is FALSE when Guest1 arrives between 5:45 PM and 5:55 PM, and p is TRUE if Guest1 arrives at 5:56 PM. Any arrival time outside of the "On-Time" interval of 5:56 PM to 6:05 PM is FALSE by definition.

Fuzzy Sets

A *fuzzy set* $A \subseteq X$ is defined to be a set of objects which have grades of membership in the real interval [0, 1]. The membership function (MF) $\mu_A(x) \in [0, 1]$ assigns a degree of membership for each $x \in A$. Then fuzzy set A consists of the objects in A with grades of membership $\mu_A(x)$ such that for *fuzzy variables x,*

$$A = \{(x, \mu_A(x)) \mid x \in X\}. \quad (1)$$

If the values of $\mu_A(x)$ are restricted to {0, 1} then A is reduced to a crisp set. We note that $\mu_A(x)$ may be discrete or continuous. Discrete objects may be nominal in measurement scale (enumerated) or ordinal (ordered) [4].

We may visualize the MF $\mu_A(x)$ for each value of fuzzy variable $x \in A$. Continuing with the dinner party example described *supra*, let the fuzzy set $A = \{\text{"Early"}, \text{"On-Time"}, \text{"Late"}\}$, and let the fuzzy variable x have MF $\mu_A(x)$. Figure 1 shows the grades of membership $\mu_A(x)$ as a function of minutes of arrival before or after 6:00 PM. So, if Guest1 arrives in the interval 5:45 PM to 5:59:59 PM, Guest1 is a member of the set "Early" with grades of membership increasing from 0 to 1 and back to 0 as he approaches the start of the party. As Guest1 arrives closer to 6 PM, his membership in the set "Early" decreases and his membership in the set "On-Time" increases. Hence, he may be jointly a member of both sets at the same time

[2] The law of the excluded middle states that, for a proposition $p, p \vee \sim p$, is true. Either proposition p is true or its negation, $\sim p$, is true. Both p and $\sim p$ cannot simultaneously be true. There is no middle value or partial truth permitted.

since the law of the excluded middle does not hold. Similarly, if Guest1 is late, his grade of membership in the set “On-Time” decreases and his grade of membership in the set “Late” increases.

The MFs in Figure 1 are called “triangular” MFs due to their shape. They represent proportionally increasing and decreasing degrees of membership in their respective sets. The set of membership grades specified by a MF must be convex. Only two adjacent MFs may overlap.

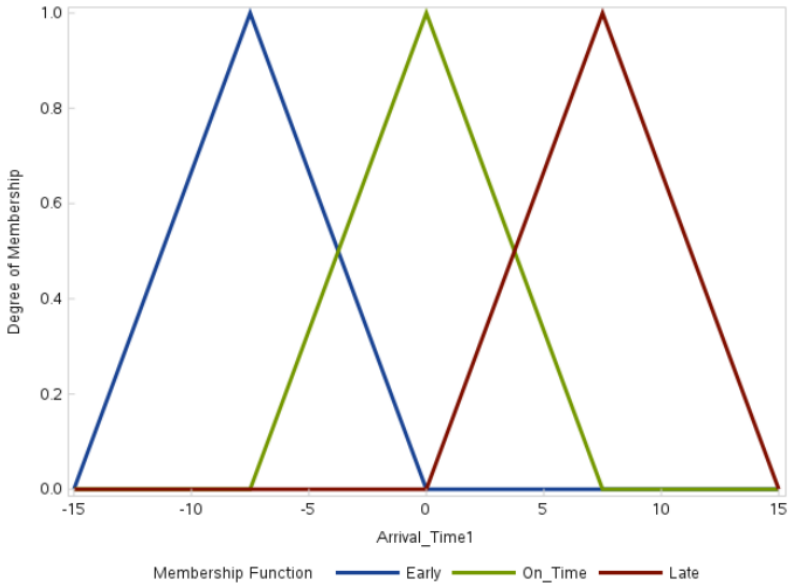

Figure 1: Triangular Membership Function

If the MFs are specified as trapezoids as in Figure 2, we see that there is an interval for each x which represents full membership in each respective fuzzy set. A guest is 100% early if he is between 15 and 10 minutes early, with membership grade “Early” declining rapidly until a guest arriving early, e.g., at 4 minutes to 6 PM, is both early and on-time. The guest who arrives at 6 PM ± 5 minutes is 100% a member of the set “On-Time”. Similarly, the membership grade of a guest in the set “Late” arriving after 6:05 PM increases sharply until 10 minutes after 6 PM, at which point he is 100% a member of the set “Late”.

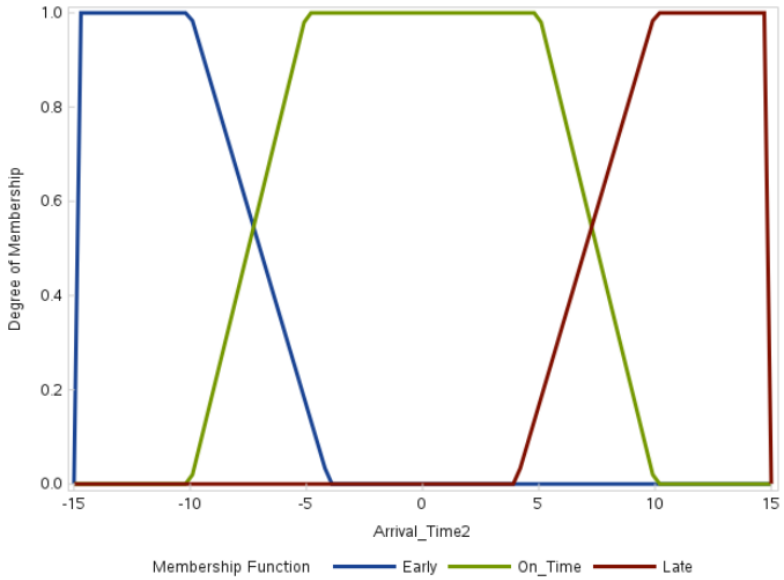

Figure 2: Trapezoidal Membership Function

The host's response may also be expressed in terms of overlapping membership functions. In Figure 3, we see that the host may be surprised by an early guest, pleased that a guest has come on-time, and annoyed when a guest comes late. In this case, we used the PI membership function, which may be understood as a trapezoidal membership function with rounded feet and shoulders. Entry into the fuzzy set is gradual, accelerating upward linearly, until the rate of entry decreases.

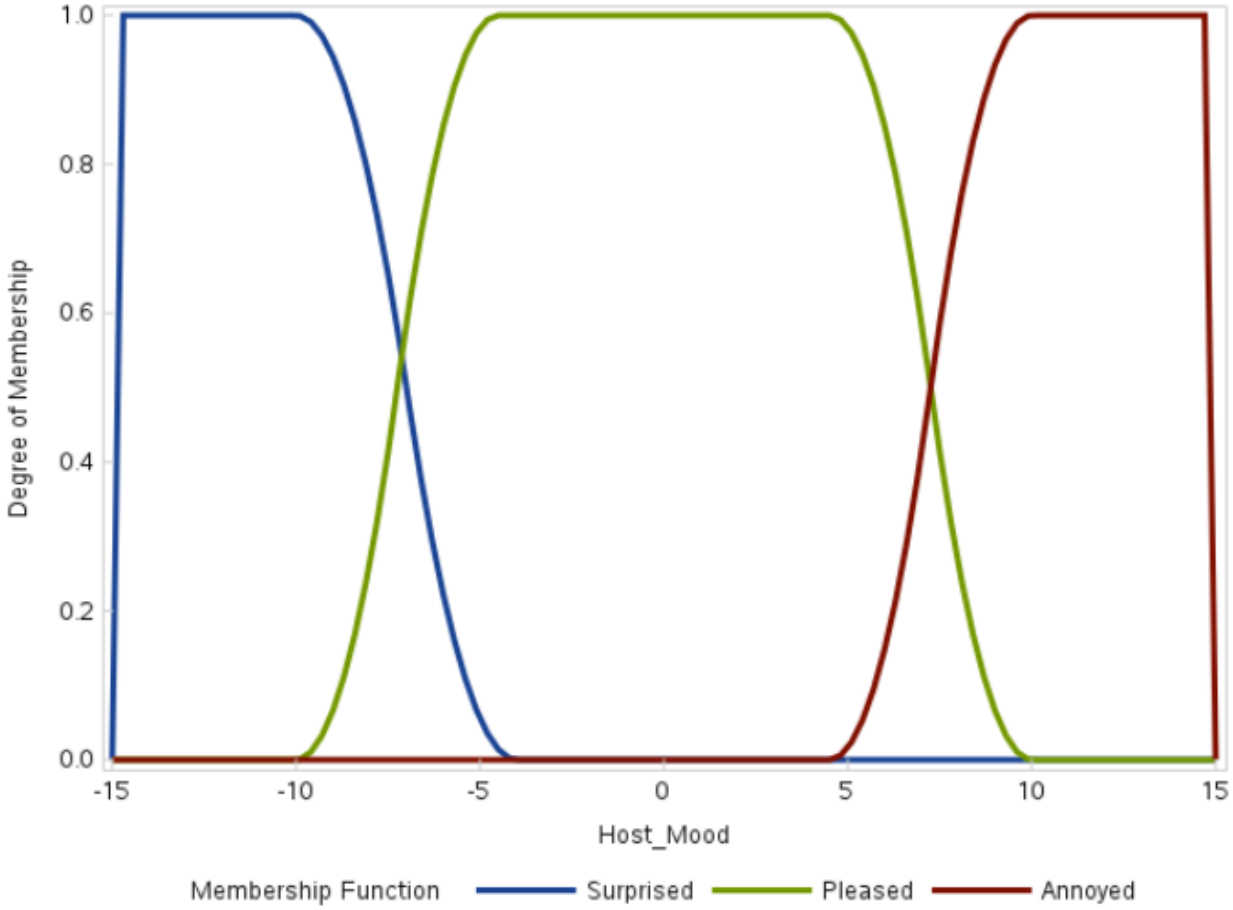

Figure 3: PI Membership Function

The mapping of input MFs into output MFs is a topic that deserves its own place in the discussion and is outside the scope of this paper. See, e.g., Jang et al. [4] or any introduction to fuzzy logic.

A wide variety of MF shapes has been developed to represent set memberships in many situations. The MFs in Figures 1 and 2 are among the simplest and most immediate to understand because they show the fuzzy set membership process as linear and thus proportional, or constant in an interval in the fuzzy variable x. We note that choosing the shape and boundaries of MFs is often a subjective decision informed by the specifics of the domain of discourse.

Entropy

The term *entropy* refers to the amount of disorder in a system, a collection of objects that interact with one another. Entropy as a concept became popular in the mid-1800s and was applied by scientists and engineers to measure, on a macroscopic scale, observable physical quantities such as mass, volume, pressure, and temperature. On a microscopic scale, entropy describes, using statistical methods and probability theory, the motions of microscopic objects such as the molecules of a gas [5].

Shannon Entropy[3]

In 1948, Claude Shannon introduced the concept of *information entropy* as a measure of the information-carrying capacity of a communications channel [2]. Appendix A contains a brief description of this topic. Further discussion of information entropy is beyond the scope of this paper, but the Wikipedia article on entropy and information theory is accessible and informative [5].

We state the formula for information entropy verbatim from [5] to motivate the discussion of *fuzzy entropy*. The function $p(z)$ represents the probability of event z occurring in set $\mathcal{Z}$.

> Given a discrete random variable Z, which takes values in the set $\mathcal{Z}$ and is distributed according to p: $\mathcal{Z} \to [0, 1]$, the entropy is
>
> $$H(Z) = -\sum_{z \in Z} p(z) \log p(z) \tag{2}$$
>
> where Σ denotes the sum over the variable's possible values.

Thus, information entropy $H(\mathrm{Z})$ is the average information[4] contained in each occurrence of a random variable. Shannon's information entropy is directly analogous to the entropy in statistical thermodynamics [5]. In the case of a binary-valued variable with two probabilities $p(z)$ and $q(z) = 1 - p(z)$, equation (2) becomes

$$\begin{aligned} H(Z) &= -(p(z) \log p(z) + q(z) \log q(z)) \\ &= -(p(z) \log p(z) + (1 - p(z)) \log(1 - p(z))) \end{aligned} \tag{3}$$

Fuzzy Entropy

Information entropy is a measure of the indefiniteness of a process that is based upon probability. De Luca and Termini interpret the indefiniteness of a process in terms of *intrinsic ambiguity* instead of *statistical variation* [3]. They define *fuzzy entropy* as *average intrinsic information* which can be used to inform a choice such as classifying a set of objects into fuzzy classes based on memberships in fuzzy sets.

De Luca and Termini extended Shannon's definition of probabilistic entropy to fuzzy sets as

$$H_1(A) = -\sum_{i=1}^{N} (\mu_A(x_i) \log \mu_A(x_i) + \big(1 - \mu_A(x_i)\big) \log(1 - \mu_A(x_i)) \tag{4}$$

where $\mu_A(x_i)$ are the fuzzy membership values of objects x_i in fuzzy set A [6]. They consider two kinds of information involved in the computation of equation (4) for $x_i \in \{0, 1\}$: the total average uncertainty in predicting the particular element x_i in X and the uncertainty of assigning the value of 1 or 0 to that element [3, Section 3].

---

[3] "When Shannon first derived his famous formula for information, he asked von Neumann what he should call it and von Neumann replied "You should call it entropy for two reasons: first because that is what the formula is in statistical mechanizes [sic] but second and more important, as nobody knows what entropy is, whenever you use the term, you will always be at an advantage!" Source: http://www.spatialcomplexity.info/what-von-neumann-said-to-shannon/

[4] From probability theory, we have the concept of expected value (expectation, mean, first moment). The formula for the expected value of a finite discrete random variable X is $E(x) = \sum_{i=1}^{N} p_i x_i$.

Applying Fuzzy Entropy Measures to Feature Selection

The quantity $H_1(A)$ may be considered to be a *fuzzy entropy measure* because it can be used to quantify the deviation of a fuzzy set from a well-defined crisp reference set, A_0.[5] By definition, a crisp set has no imprecision or vagueness, so $H(A_0) = 0$.

Consider sample data set X composed of $|X|$ rows of observations[6] and t columns of features $f_1, \dots, f_t$ that comprise measurements of the objects under consideration. There is also a noncomputational column of class labels associated with X that represents the class in which each observation belongs. We can normalize the magnitude of each feature so that X consists of t feature vectors each mapped into [0, 1].

We may characterize a crisp set of interest by first computing vectors $\mathbf{v}_i = (v_{i1}, \dots, v_{it})$ which may be specified by definition, domain knowledge, or computed from the sample data set X_i as the ideal vectors $\mathbf{v}_i$ of features $1, \dots, t$ in classes $C_i, i = 1{:}\,N$. We can use the $|X_i|$ rows $\mathbf{x_j} = (x_{j1}, \dots, x_{jt})$ from each class C_i as specified by the class label vector, and compute N ideal vectors using the generalized mean

$$v_{ik} = \left(\frac{1}{|X_i|} \sum_{j=1}^{|X_i|} x_{jk}^p \right)^{\frac{1}{p}}, k = 1{:}\,t \tag{5}$$

for each feature k in class C_i. See Appendix B for more details.

For a crisp set A_0, we know that there is no imprecision or vagueness so that an element x of a crisp set either belongs or does not belong to A_0 with membership grade 1 or 0. Let us define a similarity function $S(\mathbf{x}, \mathbf{v})$ which returns a similarity value indicative of sample vector $\mathbf{x}$ being in the same class as ideal vector $\mathbf{v}$. In the ideal case, if $\mathbf{x}$ is in the same class as $\mathbf{v}$, then $S(\mathbf{x}, \mathbf{v}) = 1$. Otherwise, $S(\mathbf{x}, \mathbf{v}) = 0$. In practice, the similarity value will be in $[0, 1]$, with high similarities indicating high fuzzy membership of $\mathbf{x}$ in the class represented by ideal vector $\mathbf{v}$.

If we use the fuzzy entropy value $\mu_A(x_i)$ as the sample similarity value in equation (4), we will observe the following pattern:

Uncertainty/ Imprecision	Similarity Value	Fuzzy Entropy Measure
High	Low	High ($\mu_A(x_j) \approx 0.5$)
Low	$S(\mathbf{x}, \mathbf{v}) \approx 0$ or $S(\mathbf{x}, \mathbf{v}) \approx 1$	Low

Table 1: Relationship Between Similarity Value and Fuzzy Entropy Measure

Table 1 summarizes the relationship between uncertainty, similarity value, and fuzzy entropy measure. If the normalized vectors $\mathbf{x_j}$ and $\mathbf{v_i}$ are relatively dissimilar in the feature space $[0, 1]^t$, there is high uncertainty and hence high entropy. Conversely, if the two vectors are similar, we expect to see low entropy since there is small mismatch when comparing the sample data vector $\mathbf{x_j}$ with ideal vector $\mathbf{v_i}$. Once we have computed the similarity values for all t features using equation (4), we can decide which feature to remove on the basis of its high fuzzy entropy measure with respect to any other feature. A

[5] We note that a crisp set has membership values {0,1} and that log (0) is undefined. But $lim_{x \to 0^+} x \log(x) = 0$. In the case of information entropy, for $p \to 0^+$, the probability of an event approaches 0 and therefore $p \log(p)$ is never actually attained. So, 0 log (0) is never calculated in practice.

[6] The notation $|X_i|$ represents the number of rows of observations in class i in matrix X.

feature with high fuzzy entropy measure does not contribute much information to the distinctions between the classes C_i, so it can be removed without losing essential information in distinguishing between classes.

Once a feature has been removed from the sample data set X, the procedure may be repeated until a point of minimal improvement in $H_1(A)$ has been observed. For example, Figure 4 below is a plot of fuzzy entropy for all features for the Credit Card Default dataset. The units of entropy are "nats" because the natural logarithm was used in the computations. We see that for all of the features in the dataset, the entropy is 26,391.2 nats (Appendix D). Marriage contributes significantly more randomness to the dataset than any other feature, with Education following. If Marriage is removed from the dataset, the entropy is 24,205.6 nats, a decrease of 8.28%. Successive deletions of features from Education to Bill_amt5 remove relatively small amounts of randomness. There is a precipitous increase in entropy with the inclusion of Bill_amt5 in the feature set due to the lack of strong relationship between a cardmember's default status and amount of monthly payments. Inversely, significant explanatory power of monthly payments (Pay_amt1-Pay_amt6) and gender of the cardmember (Sex) appear to be strongly related to credit card default. We would expect that the first steps in building a model to predict credit default would be to start with the fewest features (Sex and monthly payments) and add more features incrementally.[7]

[7] We refrain from commenting on the behavioral significance of the Marriage and Sex features since all measures of relationship are inherently fuzzy by definition, and to attempt to fuzzify further would exceed our brief.

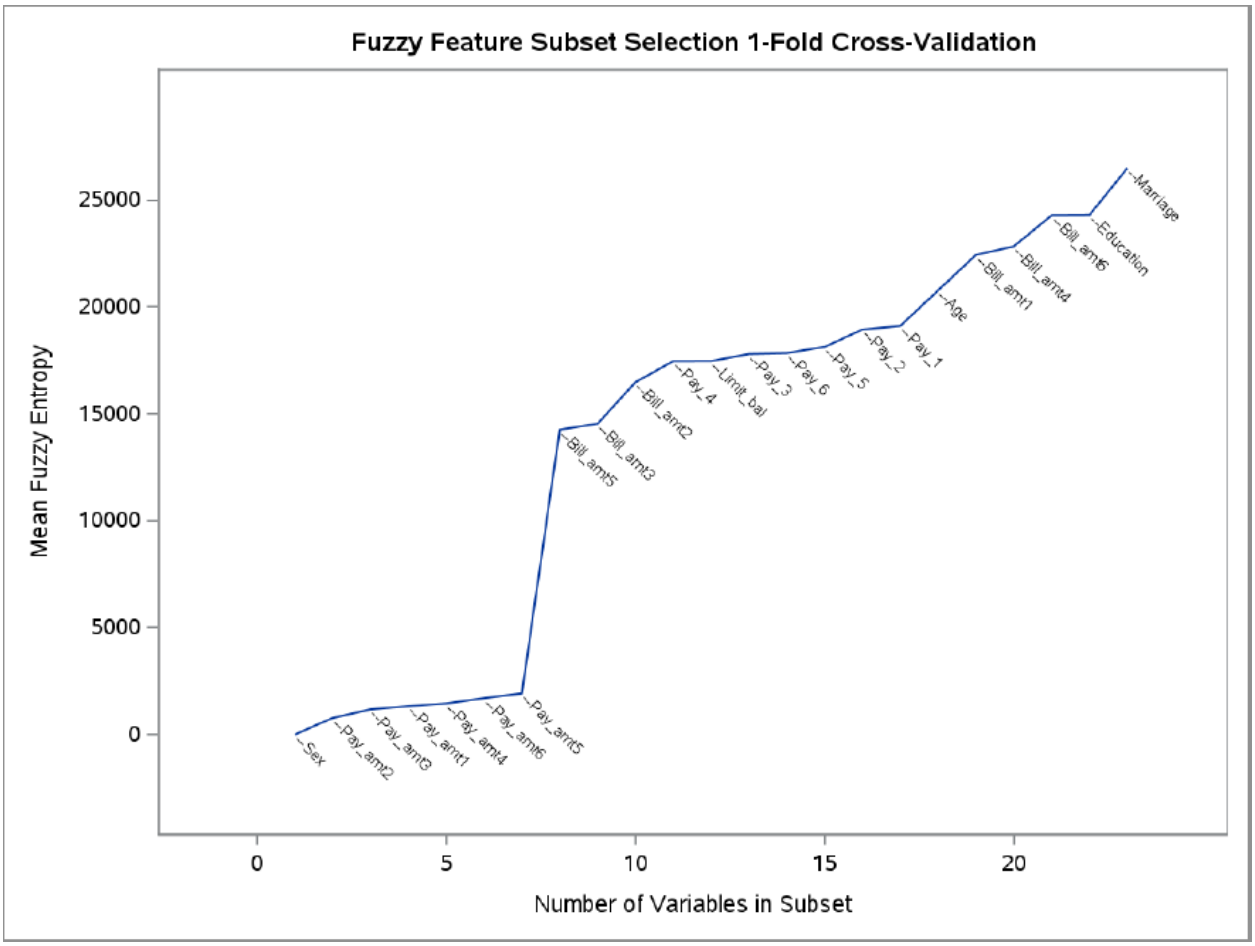

Figure 4: Entropy Plot of Credit Card Default Data

Applying A Similarity Classifier to Classification

Classifying a set of data requires that the observations be assigned into separate classes based on categories of membership. Ideally, each class is described by a nonoverlapping set of characteristics (class labels) so that there is a one-to-one correspondence between observations and features. Ideally, each observation is assigned to a unique class label based on the similarity of a feature vector $\mathbf{x}_j$ to an ideal vector $\mathbf{v}_i$.

Borrowing the terminology from the feature selection topic *supra*, we can state that the goal of the classification task is to assign a data set X of objects $\mathbf{x}_j$ into N classes $C_i, i = 1{:}N$. Each object is described by t features, or measurements $f_1, \dots, f_t$. We may assume that the magnitude of each feature is normalized into the interval $[0, 1]$ so that the feature matrix is composed of $|\mathrm{X}|$ rows and t columns plus a column of class labels.

We may use equation (5) to compute the ideal vectors, and assume that they are "ideal", even though they are based on sample data. Once we have computed the N ideals, we can compute similarities between observations and the ideal vectors. The similarity between $\mathbf{x}_j$ and the ideals $\mathbf{v}_i$ may be computed using the formula for the generalized Łukasiewicz structure, which is called a *similarity measure.* It is discussed in detail in [6]:

$$S(\mathbf{x_j}, \mathbf{v_i}) = \left(\frac{1}{n} \sum_{k=1}^{n} w_k \left(1 - \left| x_{jk}^p - v_{ik}^p \right| \right)^{\frac{m}{p}} \right)^{\frac{1}{m}} \quad (6)$$

for $\mathrm{x}_{jk},\ \mathrm{v}_{ik} \in [0, 1]$. The parameter p is the same as for the generalized mean in equation (5). The parameter m is a positive real number. The parameters w_k are weights that can be assigned to indicate the relative importance of different features. If $\mathbf{x_j}$ is very similar to $\mathbf{v_i}$ then their difference in entropy per feature is small and the weighted sum of all n features is close to 1. If the per-feature differences are large, then the similarity value is close to 0.

So, for a given observation $\mathbf{x_j}$, we compare it to all N ideals and we assign it to class C_i according to

$$S(\mathbf{x_j}, \mathbf{v_i}) = \max_{i=1,\dots,N} S(\mathbf{x_j}, \mathbf{v_i}) \quad (7)$$

The observation $\mathbf{x_j}$ is assigned to class i based on the highest similarity value between the sample data and the ideal vector of attributes for class i.

The parameters m and p and the weights w_k must be determined by optimizing

$$arg \max_{m,p,w_k} \sum_{i,j} S(\mathbf{x_j}, \mathbf{v_i}) \text{ for } m > 0, p > 0, w_k > 0 \quad (8)$$

We used differential evolution as the optimization technique [8]. It has the advantages of simple structure, fast convergence, ease of use, speed and robustness. See Appendix C for a description of the differential evolution algorithm.

Methodology

Feature Selection

Feature selection using a fuzzy entropy measure combined with a similarity classifier is performed in four steps [6].

In **Step 1**, the data are separated into training and validation samples by random assignment of observations into one of the two datasets. We used proportions of 70% training/30% validation in selecting observations. Ideal vectors are called "ideal" in Luukka's terminology [6] because they are, ideally, the representative vectors for each group of data in each class. They may be based on domain knowledge and need not be computed from data.

In **Step 2**, the similarity S of training set feature element x_{jk} with the training set ideal element v_{ik} is

$$S(\mathrm{x}_{jk}, \mathrm{v}_{ik}) = w_k \left(1 - \left| x_{jk}^p - v_{ik}^p \right| \right)^{\frac{1}{p}} \quad (9)$$

where the parameter $p = 1$. Setting p to 1 results in what is called a "normal Łukasiewicz structure" [9]. This similarity measure is used in step 3.

In **Step 3**, the fuzzy entropy measure $H_k = \sum_{i=1}^{N} \sum_{j=1}^{n} H_1\big(S(\mathrm{x}_{jk}, \mathrm{v}_{ik})\big)$ is applied to the training data using the training data ideal vectors to compute the entropy value for each feature k over all observations $(j = 1, \dots, n)$ and ideals $(i = 1, \dots, N)$.

In **Step 4**, the feature k with the highest entropy is removed from the training dataset. Effectively, the column of the dataset and its corresponding ideals are omitted from the dataset either by deleting them or by skipping over them in further processing. Using equation (7) to summarize classification performance for the initial computation of H_k, and for subsequent iterations, a stopping criterion may be the lack of improvement in classification accuracy or in similarity. The subset of the remaining features is

then chosen to represent the original set of data in further processing. The choice of subset may depend on domain knowledge about the data or on observing a relatively large decrease in fuzzy entropy between features.

Feature Classification

In the feature classification step, scenarios are created consisting of sets of parameter values for the differential evolution (DE) algorithm (see Appendix C). The DE algorithm is applied to the training data to compute ideal vectors $\mathbf{v_i}$ as per [8] and values for the m and p parameters required by the generalized Łukasiewicz structure described by equation (6)**.** Similarity values $S(\mathrm{x}_{jk}, \mathrm{v}_{ik})$ for individual feature vectors are calculated, and the vectors are assigned to a class using equation (7). After all of the observations in the training data have been assigned to classes, performance metrics based on the training data are computed. In the validation phase, the m and p parameters derived from the training data are applied to the validation data, feature vectors are classified using equation (7), and performance metrics are similarly computed.

Since the fitness (objective) function which the DE algorithm uses is based on data and is not a mathematical function *per se*, the classification of a feature vector into a class is the outcome of optimizing the m and p parameters in equation (6). We used the softmax function to convert $S(\mathbf{x_j}, \mathbf{v_i})$ into a probability and computed the maximum cross-entropy of equation (7) to assign feature vector $\mathbf{x_j}$ into class *i*. The generalized Łukasiewicz structure as expressed in the fitness function is described in Appendix D.

Data Sets

Eight data sets for feature selection and classification were obtained from the UC Irvine Machine Learning Repository (https://archive.ics.uc.i.edu/dataset) and Kaggle.com (https://www.kaggle.com/datasets). The datasets are briefly described below. Descriptive information is taken from each dataset's webpage.

The Breast Cancer Wisconsin (Diagnostic), Credit Card Default, Dermatology, Iris, Obesity, and Wine Quality datasets were downloaded from the UCI Machine Learning Repository. The Diabetes and Parkinson's Disease datasets were downloaded from the Kaggle website.

Breast Cancer Wisconsin (Diagnostic)

Source	https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic
Description	"Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image."
Creators	William Wolberg, Olvi Mangasarian, Nick Street, W. Street
Dataset Characteristics	Multivariate
Subject Area	Health and Medicine
Associated Tasks	Classification
Feature Type	Real
# Instances	569
# Features	30
Has Missing Values?	No
Class Label Name, Values	"Diagnosis": M=Malignant, B=Benign

Table 2: Wisconsin Breast Cancer

Credit Card Default

Source	https://archive.ics.uci.edu/dataset/350/default+of+credit+card+clients
Description	"This research aimed at the case of customers' default payments in Taiwan and compares the predictive accuracy of probability of default among six data mining methods. From the perspective of risk management, the result of predictive accuracy of the estimated probability of default will be more valuable than the binary result of classification - credible or not credible clients." The data were collected from April to September, 2005.
Creators	I-Cheng Yeh
Dataset Characteristics	Multivariate
Subject Area	Business
Associated Tasks	Classification
Feature Type	Integer, Real
# Instances	30,000
# Features	23
Has Missing Values?	No
Class Label Name, Values	"Default_Payment_Next_Month": 1=Customer has defaulted on payment in month n and will repay in month n+1. 0=No default

Table 3:Credit Card Default

Dermatology

Source	https://archive.ics.uci.edu/dataset/33/dermatology
Description	"The aim for this dataset is to determine the type of Erythemato-Squamous Disease." "The differential diagnosis of erythemato-squamous diseases is a real problem in dermatology. They all share the clinical features of erythema and scaling, with very little differences. The diseases in this group are psoriasis, seborrheic dermatitis, lichen planus, pityriasis rosea, chronic dermatitis, and pityriasis rubra pilaris. Usually, a biopsy is necessary for the diagnosis but unfortunately these diseases share many histopathological features as well. Another difficulty for the differential diagnosis is that a disease may show the features of another disease at the beginning stage and may have the characteristic features at the following stages. Patients were first evaluated clinically with 12 features. Afterwards, skin samples were taken for the evaluation of 22 histopathological features. The values of the histopathological features are determined by an analysis of the samples under a microscope."
Creators	Nilsel Ilter, H. Güvenir
Dataset Characteristics	Multivariate
Subject Area	Health and Medicine
Associated Tasks	Classification
Feature Type	Categorical, Integer
# Instances	366
# Features	34

Has Missing Values?	Yes. 8 missing (in Age attribute). Distinguished with '?'. Mean imputation was performed to assign values computed from the sample data to place-holder values.
Class Label Name, Values	"Class": 1=Psoriasis, 2=Seborrheic Dermatitis, 3=Lichen Planus, 4=Ptyriasis Rosea, 5=Chronic Dermatitis, 6=Ptyriasis Rubra Pilaris

Table 4: Dermatology

Diabetes

Source	https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database https://www.kaggle.com/datasets/mathchi/diabetes-data-set/data
Description	"This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective of the dataset is to diagnostically predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage."
Creators	Donor of database: Vincent Sigillito
Dataset Characteristics	Multivariate
Subject Area	Health and Medicine
Associated Tasks	Classification
Feature Type	Real
# Instances	768
# Features	8
Has Missing Values?	No
Class Label Name, Values	"Outcome": 1="Tested positive for diabetes", 0="Did not test positive"

Table 5: Diabetes

Iris Data

Source	https://archive.ics.uci.edu/dataset/53/iris
Description	"This is one of the earliest datasets used in the literature on classification methods and widely used in statistics and machine learning. The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are not linearly separable from each other."
Creators	R.A. Fisher
Dataset Characteristics	Multivariate
Subject Area	Biology
Associated Tasks	Classification
Feature Type	Real
# Instances	150
# Features	4
Has Missing Values?	No
Class Label Name, Values	"Species": Iris Setosa, Iris Versicolor, Iris Virginica

Table 6: Iris Data

Obesity

Source	https://archive.ics.uci.edu/dataset/544/estimation+of+obesity+levels+based+on+eating+habits+and+physical+condition
Description	"This dataset include [sic] data for the estimation of obesity levels in individuals from the countries of Mexico, Peru and Colombia, based on their eating habits and physical condition. The data contains 17 attributes and 2111 records, the records are labeled with the class variable NObesity (Obesity Level), that allows classification of the data using the values of Insufficient Weight, Normal Weight, Overweight Level I, Overweight Level II, Obesity Type I, Obesity Type II and Obesity Type III. 77% of the data was generated synthetically using the Weka tool and the SMOTE filter, 23% of the data was collected directly from users through a web platform.."
Creators	Fabio Mendoza Palechor, Alexis De la Hoz Manotas
Dataset Characteristics	Multivariate
Subject Area	Health and Medicine
Associated Tasks	Classification, Regression, Clustering
Feature Type	Integer
# Instances	2111
# Features	16
Has Missing Values?	No
Class Label Name, Values	"NObeyesdad": Insufficient Weight, Normal Weight, Overweight Level I, Overweight Level II, Obesity Type I, Obesity Type II, Obesity Type III

Table 7: Obesity

Parkinson's Disease

Source	https://www.kaggle.com/datasets/vikasukani/parkinsons-disease-data-set
Description	"This dataset is composed of a range of biomedical voice measurements from 31 people, 23 with Parkinson's disease (PD). Each column in the table is a particular voice measure, and each row corresponds to one of 195 voice recordings from these individuals ('name' column). The main aim of the data is to discriminate healthy people from those with PD, according to the 'status' column which is set to 0 for healthy and 1 for PD."
Creators	Max A. Little, Patrick E. McSharry, Eric J. Hunter, Lorraine O. Ramig
Dataset Characteristics	Multivariate
Subject Area	Health and Medicine
Associated Tasks	Classification
Feature Type	Real
# Instances	196
# Features	22
Has Missing Values?	No
Class Label Name, Values	"Status": 1=Parkinson's Disease, 0=Healthy

Table 8: Parkinson's Disease

Wine Quality

Source	https://archive.ics.uci.edu/dataset/186/wine+quality
Description	"Two datasets are included, related to red and white vinho verde wine samples, from the north of Portugal. The goal is to model wine quality based on physicochemical tests (see [Cortez et al., 2009], http://www3.dsi.uminho.pt/pcortez/wine/)."
Creators	Paulo Cortez, A. Cerdeira, F. Almeida, T.Matos, J. Reis
Dataset Characteristics	Multivariate
Subject Area	Business
Associated Tasks	Classification, Regression
Feature Type	Real
# Instances	4898
# Features	11
Has Missing Values?	No
Class Label Name, Values	"Quality": Score between 0 and 10 (integer)

Table 9: Wine Quality

Results

We performed feature selection and classification using fuzzy entropy on each dataset and produced the results shown below. A 10-fold cross-validation was used to verify that the distribution of entropy was relatively unchanged over randomly-selected samples.

The fuzzy entropies are tabulated in Appendix E and grouped into subsets by subjectively deciding when a significant gap opened between two groups of features. These gaps are the boundaries between subsets of features of a particular dataset. Subsets are labelled proceeding from low entropy to high entropy. For example, in the Wisconsin Breast Cancer Entropy Plot (Figure 5), subset 2 contains features Std_Area to Worst_Smoothness, and subset 1 contains Mean_Smoothness. Then models were built on grouped subsets 2, and 12, where each subset contains all of the features in subset 2 and all of the features in subsets 1 and 2, respectively.

The following performance metrics are computed for binary-valued classification problems[8] from the 2x2 classification matrix:

- **Accuracy** is the percent of samples correctly classified out of all of the samples present in the dataset. Accuracy may be misleading when a dataset is unbalanced, e.g., when one class has much higher frequency than another.
- **Precision** (for the positive class, e.g., class = 1) is the percent of samples *actually* belonging to the positive class out of all of the samples that were *predicted* to be of the positive class by the classification algorithm.
- **Recall** (for the positive class) is the percent of samples *predicted* to be belonging to the positive class out of all of the samples that *actually* belong to the positive class.
- **F1 Score** (for the positive class) is the harmonic mean of the precision and recall scores computed for the positive class.
- **Specificity** is the percent of samples *predicted* correctly to be in the negative class out of all of the samples in the dataset that *actually* belong to the negative class.

[8] https://www.v7labs.com/blog/confusion-matrix-guide

- **AUC** is the area under the Receiver Operating Characteristic curve and is a measure of a classifier's ability to distinguish between true positive classes and false positive classes. In this paper, it is computed from Somers' D statistic according to the relationship $AUC = \frac{1}{2}(D + 1)$ where D is computed from the classification matrix.
- **Matthews Correlation Coefficient** is the correlation between the predicted and actual classifications, taking all of the classification matrix cells (TP, TN, FP, FN) into account. It is a measure of the quality of binary classifications, where +1 indicates perfect prediction and -1 indicates inverse prediction. It is a balanced measure that is useful when considering unbalanced datasets, e.g., even if the classes are of widely varying sizes. It can be calculated directly from the classification matrix.

For multinary-valued classification problems, the N-class confusion matrix was converted into a 2x2 classification matrix by considering the class value c_{ii} to be the positive class, e.g., $c_{ii} = 1$, and $c_{ij} = 0$ for $i \neq j$ to be the negative class. This method is called "one-versus-rest" and produces a 2x2 classification matrix for which binary performance metrics can be computed.

We used logistic regression[9] to compute the AUC for training and validation datasets as a comparison to the fuzzy classifier.

Dataset Name: Breast Cancer Wisconsin (Diagnostic)

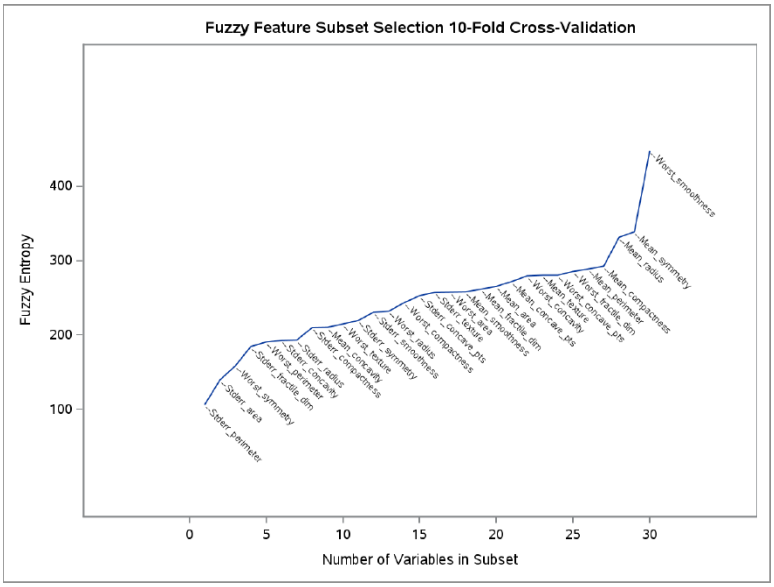

Figure 5: Breast Cancer Entropy Plot

There are two subsets of features observed. Subsets are numbered from right-to-left, starting with the features at the right side of the plot. The features in each subset are indicated in Appendix E. Subset 1 contains the rightmost feature, Worst Smoothness. Subset 2 consists of numerous features that individually add marginal amounts of uncertainty to the classification procedure, and Subset 1 provides an increase in imprecision Class value 1 (positive value) indicates a malignancy in the skin sample. The logistic regression algorithm has detected complete separation of classes in subsets 2 and 12 in each dataset.

[9] Logistic regression is a statistical model that represents the log of the odds ratio of an event as a linear combination of one or more independent variables. The coefficients of the independent variables are parameters which must be computed by optimizing a likelihood ratio. We used SAS 9.4 PROC LOGISTIC to perform the computations.

Training Data

		Fuzzy Classifier							Logistic Regression
Sub-set	Class	Accu-racy	Preci-sion	Recall	Speci-ficity	F1 Score	AUC	MCC	AUC
2	0	0.9273	0.9108	0.9800	0.8389	0.9441	0.9095	0.8505	1.0000
2	1	0.9273	0.9615	0.8389	0.9800	0.8961	0.9095	0.8452	
12	0	0.9298	0.9142	0.9800	0.8456	0.9459	0.9128	0.8505	1.0000
12	1	0.9298	0.9618	0.8456	0.9800	0.9000	0.9128	0.8505	

Table 10: Wisconsin Breast Cancer Training Data Performance Metrics

Validation Data

		Fuzzy Classifier							Logistic Regression
Sub-set	Class	Accu-racy	Preci-sion	Recall	Speci-ficity	F1 Score	AUC	MCC	AUC
2	0	0.9235	0.9896	0.8879	0.9841	0.9360	0.9360	0.8494	1.0000
2	1	0.9235	0.8378	0.9841	0.8879	0.9051	0.9360	0.8494	
12	0	0.9353	0.9364	0.9626	0.8889	0.9493	0.9258	0.8606	1.0000
12	1	0.9353	0.9333	0.8889	0.9626	0.9106	0.9258	0.8606	

Table 11: Wisconsin Breast Cancer Validation Data Performance Metrics

Dataset Name: Credit Card Default

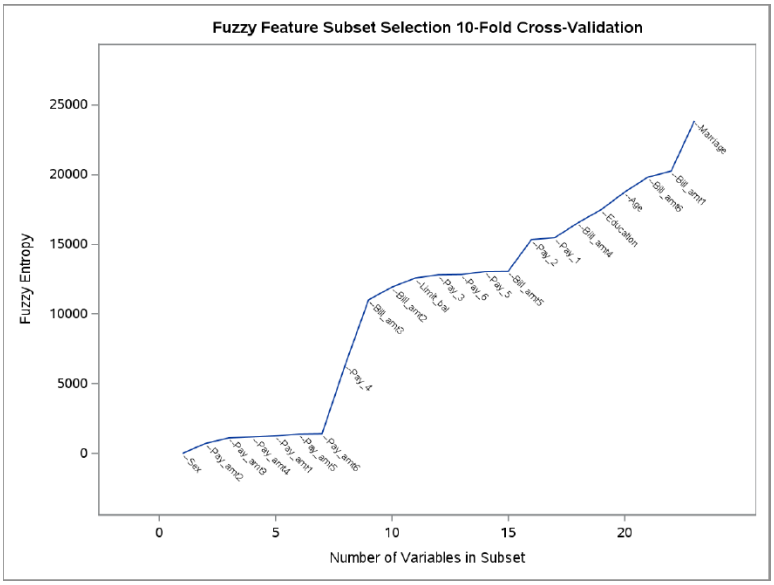

Figure 6: Credit Card Default Entropy Plot

There are three subsets of features observed, and similar remarks apply as for Wisconsin Breast Cancer. Class value 1 (positive value) indicates that a default on payment in month n was followed by a payment in month n+1. We note the decrease in classification performance as more features are added to the

modeling process. The decline in accuracy from Subset 3 to Subset 23, and the further decline from Subset 23 to subset 123 demonstrates the importance of selecting only those features that produce ideals that represent each class. The logistic regression algorithm returns better predictive accuracy than the fuzzy classifier.

Training Data

		Fuzzy Classifier							**Logistic Regression**
Subset	Class	Accuracy	Precision	Recall	Specificity	F1 Score	AUC	MCC	AUC
3	0	0.7863	0.8198	0.9297	0.2832	0.8713	0.6065	0.2747	0.6670
3	1	0.7863	0.5347	0.2832	0.9297	0.3703	0.6065	0.2747	
23	0	0.7305	0.8080	0.8574	0.2855	0.8320	0.5714	0.1565	0.7250
23	1	0.7305	0.3633	0.2855	0.8574	0.3198	0.5714	0.1565	
123	0	0.6569	0.7921	0.7579	0.3025	0.7746	0.5302	0.0575	0.7260
123	1	0.6569	0.2627	0.3025	0.7579	0.2812	0.5302	0.0575	

Table 12: Credit Card Default Training Data Performance Metrics

Validation Data

		Fuzzy Classifier							**Logistic Regression**
Subset	Class	Accuracy	Precision	Recall	Specificity	F1 Score	AUC	MCC	AUC
3	0	0.7889	0.8219	0.9313	0.2831	0.8732	0.6072	0.2773	0.6580
3	1	0.7889	0.5368	0.2831	0.9313	0.3707	0.6072	0.2773	
23	0	0.7276	0.8084	0.8532	0.2816	0.8302	0.5674	0.1463	0.7250
23	1	0.7276	0.3505	0.2816	0.8532	0.3123	0.5674	0.1463	
123	0	0.6624	0.8161	0.7325	0.4134	0.7720	0.5729	0.1318	0.7250
123	1	0.6624	0.3031	0.4134	0.7325	0.3498	0.5729	0.1318	

Table 13: Credit Card Validation Data Performance Metrics

Dataset: Dermatology

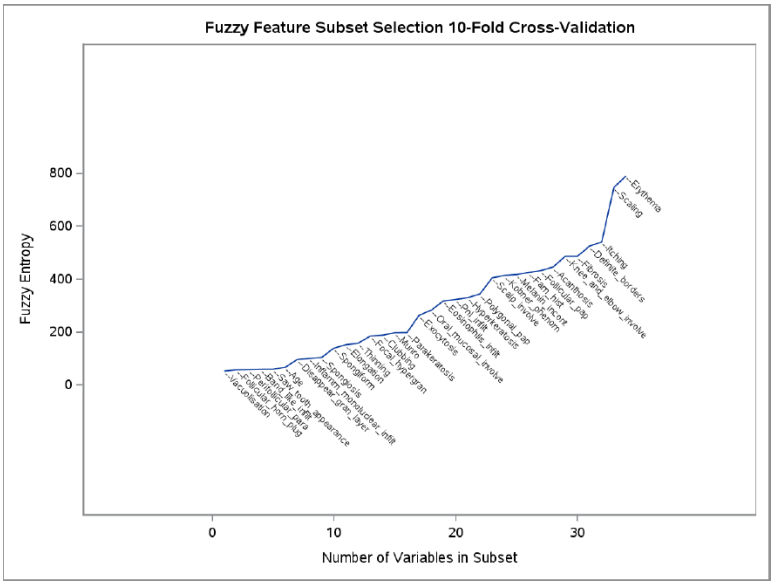

Figure 7: Dermatology Entropy Plot

Features that are similar to an ideal vector produce less entropy and are hence closely grouped. The 10-fold cross-validation entropy chart appears to show three subsets of features, but the actual difference between groups is relatively few nats in each case, so we created only one set of features comprising all features for analysis.

The logistic regression program reports that "[t]here is a complete separation of data points detected" in each class. This message indicates that each class is disjoint from every other class and that the logistic regression model may be invalid. However, we can use this disjointedness to test the ability of the fuzzy classifier to discriminate between classes. We would expect to see 100% accuracy and AUC = 1 for each class, but instead, we observe variation in the performance metrics.

Training Data

		Fuzzy Classifier							Logistic Regression
Sub-set	Class	Accu-racy	Preci-sion	Recall	Speci-ficity	F1 Score	AUC	MCC	AUC
1	1	0.9344	0.9844	0.7975	0.9944	0.8811	0.896	0.8453	1.0000
1	2	0.8456	0.5484	0.3953	0.9352	0.4595	0.6653	0.3789	1.0000
1	3	0.8571	0.5814	0.9804	0.8269	0.7299	0.9037	0.6817	1.0000
1	4	0.8958	0.6429	0.5143	0.9554	0.5714	0.7348	0.5171	1.0000
1	5	0.9498	0.8750	0.7568	0.9820	0.8116	0.8694	0.7856	1.0000
1	6	0.9768	0.7222	0.9286	0.9796	0.8125	0.9541	0.8075	1.0000

Table 14: Dermatology Training Data Performance Metrics

Validation Data

		Fuzzy Classifier							Logistic Regression
Sub-set	Class	Accu-racy	Preci-sion	Recall	Speci-ficity	F1 Score	AUC	MCC	AUC
1	1	0.8972	0.8667	0.7879	0.9459	0.8254	0.8669	0.7545	1.0000
1	2	0.8411	0.5385	0.3889	0.9326	0.4516	0.6607	0.3681	1.0000
1	3	0.8598	0.5882	0.9524	0.8372	0.7273	0.6735	0.9131	1.0000
1	4	0.8598	0.4667	0.5000	0.9140	0.4828	0.7070	0.4021	1.0000
1	5	0.9439	1.0000	0.6000	1.0000	0.7500	0.8000	0.7505	1.0000
1	6	0.9626	0.6667	0.6667	0.9802	0.6667	0.8234	0.6469	1.0000

Table 15: Dermatology Validation Data Performance Metrics

Dataset Name: Diabetes

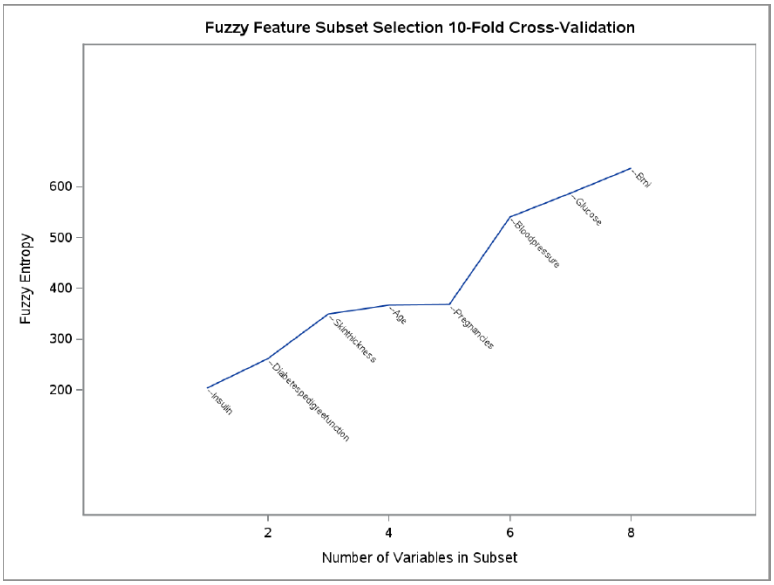

Figure 8: Diabetes Entropy Plot

The entropy plots show roughly equivalent distributions of disorder so we created only one subset using all of the features.

Training Data

		Fuzzy Classifier							**Logistic Regression**
Sub-set	Class	Accu-racy	Preci-sion	Recall	Speci-ficity	F1 Score	AUC	MCC	AUC
1	0	0.7156	0.8294	0.7086	0.7287	0.7643	0.7186	0.4196	0.8430
1	1	0.7156	0.5732	0.7287	0.7086	0.6417	0.7186	0.4196	

Table 16: Diabetes Training Data Performance Metrics

Validation Data

		Fuzzy Classifier							**Logistic Regression**
Sub-set	Class	Accu-racy	Preci-sion	Recall	Speci-ficity	F1 Score	AUC	MCC	AUC
1	0	0.713	0.8559	0.6733	0.7875	0.7537	0.7304	0.4391	0.8370
1	1	0.713	0.5625	0.7875	0.6733	0.6562	0.7304	0.4391	

Table 17: Diabetes Validation Data Performance Metrics

Dataset Name: Iris

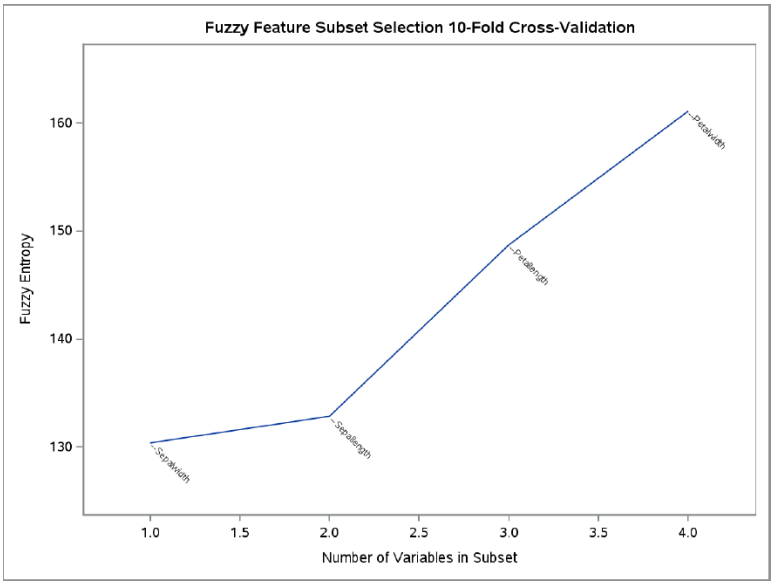

Figure 9: Iris Entropy Plot

The Iris dataset contains the species class label, petal and sepal length and petal and sepal width features. We note that the features are roughly evenly separated in terms of entropy per feature, so we created only one set of features because there is no obvious separation into subsets. The data for species 1 and 3 are completely separated according to the logistic regression AUC, so the validities of the metrics for these logistic regression models are questionable.

Training Data

		Fuzzy Classifier							Logistic Regression
Subset	Class	Accuracy	Precision	Recall	Specificity	F1 Score	AUC	MCC	AUC
1	1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
1	2	0.9143	0.8824	0.8571	0.9429	0.8696	0.9000	0.8059	0.8040
1	3	0.9143	0.8611	0.8857	0.9286	0.8732	0.9071	0.8087	1.0000

Table 18: Iris Training Data Performance Metrics

Validation Data

		Fuzzy Classifier							Logistic Regression
Subset	Class	Accuracy	Precision	Recall	Specificity	F1 Score	AUC	MCC	AUC
1	1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
1	2	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.8960
1	3	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

Table 19: Iris Validation Data Performance Metrics

Dataset Name: Obesity

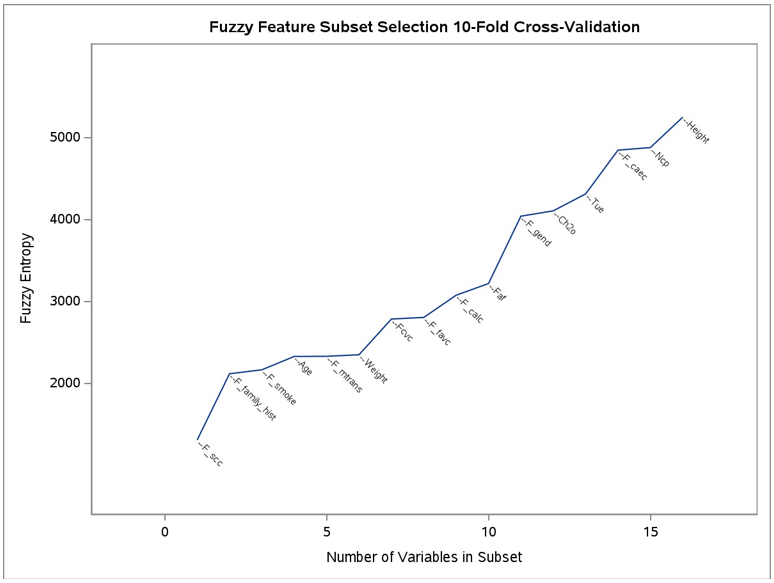

Figure 10: Obesity Entropy Plot

The obesity entropy plots reveal distinct “breaks” in the distribution of entropy, but there are relatively few features comprising a subset so all of the features were collected into one set.

Training Data

		Fuzzy Classifier							Logistic Regression
Sub-set	Class	Accu-racy	Preci-sion	Recall	Speci-ficity	F1 Score	AUC	MCC	AUC
1	0	0.8580	0.4662	0.6859	0.8835	0.5551	0.7847	0.4868	1.0000
1	1	0.8614	0.3333	0.0199	0.9937	0.0376	0.5068	0.0521	0.8810
1	2	0.8513	0.2258	0.0345	0.9812	0.0598	0.5078	0.0377	0.8180
1	3	0.8600	0.0000	0.0000	0.9969	.	0.4984	-0.0208	0.8210
1	4	0.8310	0.4787	0.1829	0.9603	0.2647	0.5716	0.21860	0.8180
1	5	0.7600	0.3419	0.7644	0.7592	0.4725	0.7618	0.3921	0.9950
1	6	0.7519	0.3818	0.9956	0.7077	0.5519	0.8516	0.5174	1.0000

Table 20: Obesity Training Data Performance Metrics

Validation Data

		Fuzzy Classifier							Logistic Regression
Sub-set	Class	Accu-racy	Preci-sion	Recall	Speci-ficity	F1 Score	AUC	MCC	AUC
1	0	0.8687	0.4922	0.7778	0.8820	0.6029	0.8299	0.5488	1.0000
1	1	0.8623	0.4000	0.0233	0.9945	0.0440	0.5089	0.0687	0.8740
1	2	0.8465	0.4107	0.2644	0.9394	0.3217	0.6019	0.2471	0.8270
1	3	0.8592	0.0000	0.0000	0.9963	.	0.4982	-0.0225	0.8580
1	4	0.8386	0.5349	0.2190	0.9620	0.3108	0.5905	0.2677	0.8370
1	5	0.8386	0.4581	0.7978	0.8453	0.5820	0.8215	0.5199	1.0000
1	6	0.7658	0.3951	0.9897	0.7252	0.5647	0.8575	0.5297	1.0000

Table 21: Obesity Validation Data Performance Metrics

Dataset Name: Parkinson's Disease

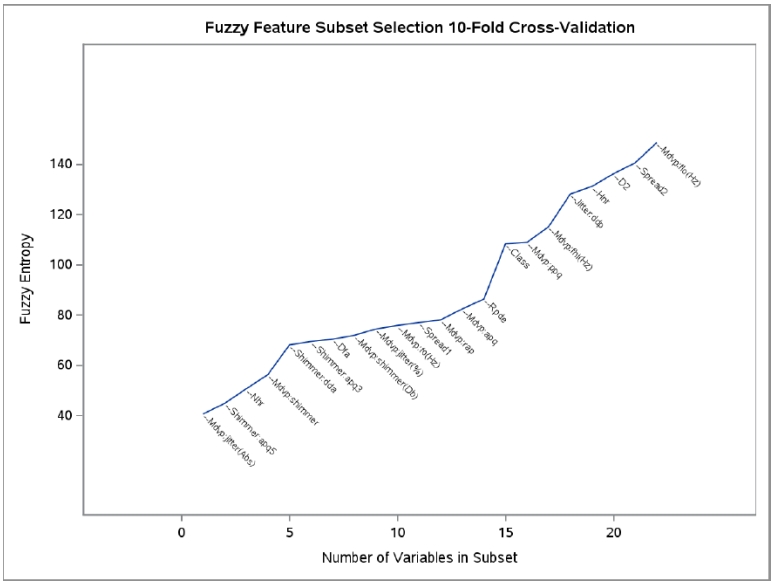

Figure 11: Parkinson's Disease Entropy Plot

The range of entropy is so limited that there would be minimal marginal improvement to separating the features into subsets, so all of the features were used as a single set.

Compared to the high AUC score in the training data-built logistic regression model, the fuzzy classifier performance is mediocre. The perfect AUC score in the validation data-built logistic regression model indicates that there is complete separation of classes and hence the validity of the model is questionable.

Training Data

		Fuzzy Classifier							**Logistic Regression**
Subset	Class	Accuracy	Precision	Recall	Specificity	F1 Score	AUC	MCC	AUC
1	0	0.6423	0.4302	1.0000	0.5100	0.6016	0.7550	0.4684	0.9450
1	1	0.6423	1.0000	0.5100	1.0000	0.6755	0.7550	0.4684	

Table 22: Parkinson's Training Data Performance Metrics

Validation Data

		Fuzzy Classifier							**Logistic Regression**
Subset	Class	Accuracy	Precision	Recall	Specificity	F1 Score	AUC	MCC	AUC
1	0	0.7586	0.4348	0.9091	0.7234	0.5882	0.8162	0.5069	1.0000
1	1	0.7586	0.9714	0.7234	0.9091	0.8293	0.8162	0.5069	

Table 23: Parkinson's Validation Data Performance Metrics

Dataset Name: Wine Quality

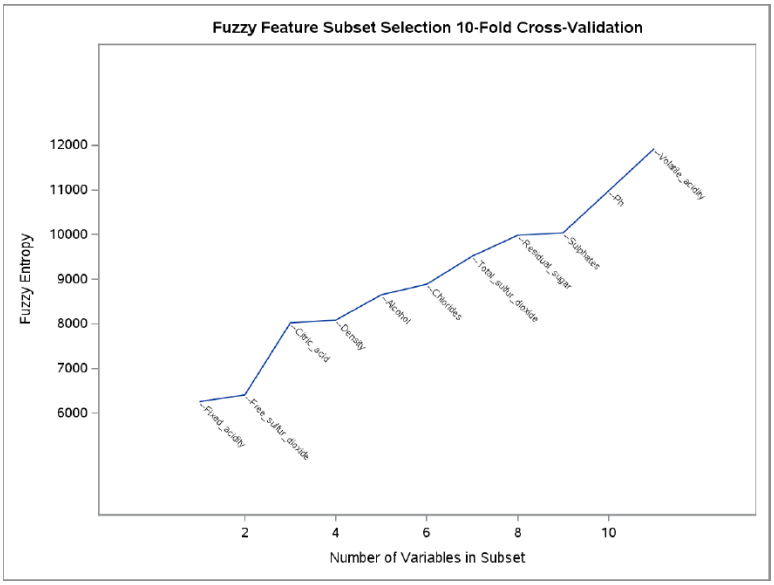

Figure 12: Wine Quality Entropy Plot

We grouped all of the wine quality features into one set because, despite the fact that there are two and three apparent groups of features in Figure 12, the number of features in each group is too small to merit separate processing.

Training Data

		Fuzzy Classifier							Logistic Regression
Sub-set	Class	Accu-racy	Preci-sion	Recall	Speci-ficity	F1 Score	AUC	MCC	AUC
1	3	0.9051	0.0165	0.3333	0.9077	0.0314	0.6205	0.0562	0.7750
1	4	0.7102	0.0675	0.5987	0.7141	0.1213	0.6564	0.1231	0.7810
1	5	0.6430	0.4443	0.3413	0.7908	0.3861	0.5661	0.1429	0.7870
1	6	0.5685	0.5281	0.1042	0.9279	0.1741	0.5161	0.0568	0.6000
1	7	0.7913	0.2623	0.1415	0.9207	0.1838	0.5311	0.0811	0.7850
1	8	0.8486	0.0700	0.3309	0.8646	0.1155	0.5977	0.0955	0.8080
1	9	0.9602	0.0216	1.0000	0.9602	0.0423	0.9801	0.1441	0.9950

Table 24: Wine Quality Training Data Performance Metrics

Validation Data

		Fuzzy Classifier							Logistic Regression
Sub-set	Class	Accu-racy	Preci-sion	Recall	Speci-ficity	F1 Score	AUC	MCC	AUC
1	3	0.9131	0.0062	0.1111	0.9168	0.0117	0.5140	0.0069	0.8200
1	4	0.8375	0.0685	0.3125	0.8554	0.1124	0.5839	0.0838	0.7930
1	5	0.6566	0.4705	0.3354	0.8144	0.3916	0.5749	0.1661	0.7830
1	6	0.5614	0.4800	0.0424	0.9644	0.0778	0.5034	0.0174	0.6140
1	7	0.7943	0.1765	0.0650	0.9396	0.0950	0.5023	0.0071	0.8080
1	8	0.6596	0.0382	0.4386	0.6663	0.0702	0.5525	0.0374	0.8180
1	9	0.9044	0.0000	0.0000	0.9048	.	0.4524	-0.0070	1.0000

Table 25: Wine Quality Validation Data Performance Metrics

Summary

We introduced the concept of a fuzzy set in comparison to a crisp set, and the differences in conceptualizing characteristic of each approach to describing real-world events. We explained how to map an actual event into its fuzzy set formulation through fuzzy membership functions. Following Luukka [6], we converted fuzzy membership into entropy by using DeLuca and Termini's concept of fuzzy entropy. We created entropy plots to motivate feature selection. We implemented a fuzzy classifier in SAS/IML® based on the Łukasiewicz normal structure and tested its efficacy as a classifier against several datasets and subsets of features selected using fuzzy entropy for feature selection. We compared the fuzzy classifier performance to logistic regression models and observed that fuzzy classifier performance lagged logistic regression performance based on AUC but was able to produce results when logistic regression failed due to complete or quasi-complete separation of data.

Appendix A

Information Entropy

From reference [4], we know that an event E occurs with probability p. When p is high, E occurs relatively frequently so that its occurrence is no surprise to us. But if E occurs relatively infrequently, we are surprised. Hence, the "surprisal" or "self-information" of E may be expressed as

$$\frac{1}{p(E)} \tag{A.1}$$

so that the self-information is low when $p(E)$ is near 1 and high when $p(E)$ is near 0. Then we can state the information content of an event E as

$$I(E) = log\left(\frac{1}{p(E)}\right) = -log\big(p(E)\big) \tag{A.2}$$

To motivate the choice of the logarithm function to express the self-information of an event, consider a message that is encoded as binary digits, or bits. If we choose the base 2 for the logarithm function, we have

$$I(E) = -log_2(p(E)) \tag{A.3}$$

which is the number of bits required to represent the event E.

Let $E = \{e_1, \dots, e_N\}$. Then the self-information, i.e., the average number of bits or information content of all of the events e_i in the set E is

$$H(E) = -\sum_{e_i \in E} p(e_i)\, log_2\, p(e_i) \tag{A.4}$$

Appendix B

Generalized Mean and Łukasiewicz Structure

The formula for the generalized mean is

$$M_p(x_1, \dots, x_n) = \left(\frac{1}{n}\sum_{i=1}^{n} x_i^p\right)^{\frac{1}{p}} \tag{B.1}$$

where the power, p, is fixed for all i. If $p = 1$ then M_p is the arithmetic mean. If $p = 2$ then M_p is the root mean square. In general, p determines the influence of disproportionately larger values of x_i^p so that larger values of p place greater emphasis on more extreme values.

The formula for the generalized Łukasiewicz structure [6] is

$$S(\mathbf{x_j}, \mathbf{v_i}) = \left(\frac{1}{n}\sum_{k=1}^{n} w_k \left(1 - \left|x_{jk}^p - v_{ik}^p\right|\right)^{\frac{m}{p}}\right)^{\frac{1}{m}} \tag{B.2}$$

where $\boldsymbol{x_j}$ are data vectors in X and $\boldsymbol{v_i}$ are ideals C_i.

Appendix C

Differential Evolution Optimization Algorithm

Differential evolution, an algorithm for computing the solutions to global optimization problems by iteratively improving a candidate solution, was introduced by Storn and Price in 1995 [10][10]. The authors describe this algorithm as "A new heuristic approach for minimizing possibly nonlinear and non-differentiable continuous space functions."

The algorithm consists of four steps: initialization, mutation, crossover, and selection. It is stated thusly:

$$\text{Find } X = (x_1, x_2, \dots, x_D), X \in R^D \qquad \text{(D1)}$$

to minimize the objective function $f(X)$ over the population

$$P_G = (X_{1,G}, X_{2,G}, \dots, X_{NP,G}), G = 0, \dots, G_{max}$$

where $X_{i,G} = \left(x_{1,i,G}, x_{2,i,G}, \dots, x_{D,i,G}\right), i = 1, 2, \dots, NP \text{ and } G = 0, \dots, G_{max}$

where:

D is the number of features represented in $f(X)$, e.g., the length of the parameter vector
G is the generation index that identifies the successive generations of iteration
i is the population index from 1 to NP, the number of population vectors
P_G is the $NP\ x\ D$ population matrix of NP row vectors, each of length D, that contains the "genes" that are transmitted by the adaptive mechanisms of mutation and crossover for successive generations
$X_{i,G}$ is the i'th row vector of genes in the G'th generation
$x_{j,i,G}$ is the gene for the j'th feature in the i'th row vector of genes in the G'th generation

[10] The program used in this paper was downloaded from GitHub and translated into SAS/IML®. The source is https://github.com/jvgomez/sgps/blob/master/Matlab/WSN/gavec3.m .

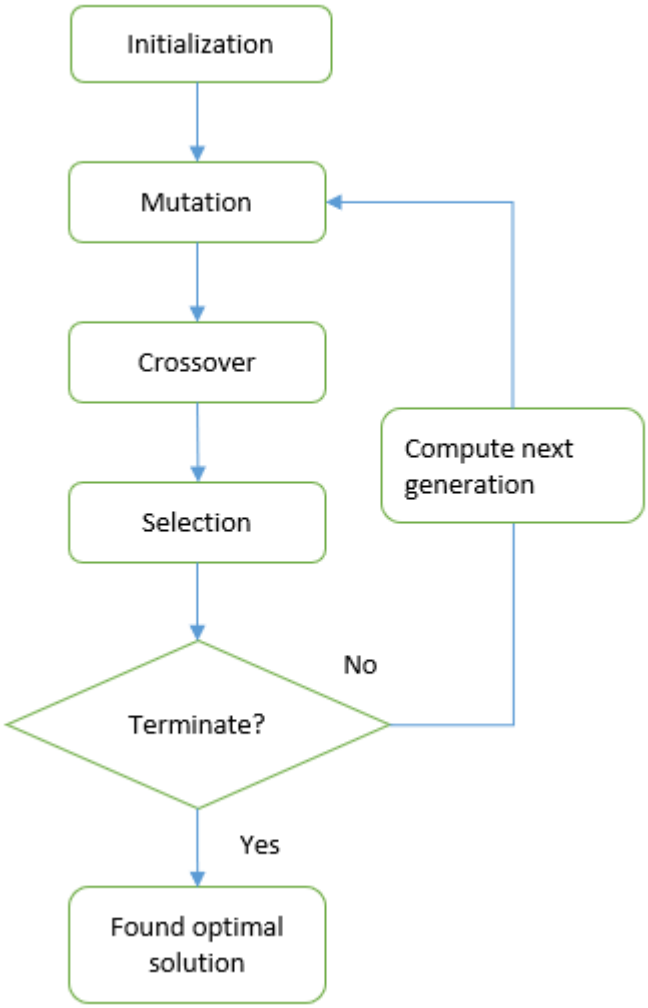

Figure D1: Flowchart of Differential Evolution Algorithm

Step 1: Initialization

The initial $P_{G=0}$ is chosen for the entire parameter space, $i = 1, \dots, NP, j = 1, \dots, D$ so that

$$x_{j,i,0} = u_j \cdot \left(x_j^U - x_j^L\right) + x_j^L, i = 1, \dots, NP \; and \; j = 1, \dots, D \quad \text{(D2)}$$

where u_j is a random variate from a uniform [0,1] distribution that is a constant for each value of feature index j, x_j^L is $min\left(x_j\right)$ and x_j^U is $max\left(x_j\right)$.

Step 2: Mutation

For each feature vector j, three other index vectors $\{r1, r2, r3\}$ are selected randomly so that

$r1 \neq r2 \neq r3 \neq i$, i=0, 1, …, NP-1 . (D3)

The mutation vector is computed by

$$v_{j,i,G+1} = x_{j,r1,G} + F \cdot \left(x_{j,r2,G} - x_{j,r3,G}\right), i = 1, \dots, NP \; and \; j = 1, .., D \quad \text{(D4)}$$

The real-valued factor F is a constant in (0, 1]. It scales the difference $\left(x_{j,r2,G} - x_{j,r3,G}\right)$ and thus controls the amplification of the mutation, the differential variation.

Step 3: Crossover

The purpose of the crossover operation is to increase the diversity of the mutated parameter vectors $v_{j,i,G}$ .A trial vector $u_{j,i,G+1}$ is computed as

$$u_{j,i,G+1} = \begin{cases} v_{j,i,G+1} \; if \; U \leq CR \vee j = k \\ x_{j,i,G} \; otherwise \end{cases} \quad \text{(D5)}$$

where U is a uniform random variate drawn from the interval [0, 1].

The crossover factor CR is a real constant in [0, 1] and controls the probability that a trial vector will come from the mutation generation instead of the current generation.

Step 4: Selection

The trial vector $u_{j,i,G+1}$ is compared to the feature vector $x_{j,i,G}$ using a "greedy approach", i.e., the feature vector that minimizes the objective function $f(X_{i,G})$ is chosen over the current generation vector, $X_{i,G}$. There is no look-ahead feature. If the mutated crossover vector produces a better optimum, it is selected.

$$X_{i,G+1} = \begin{cases} if \; U_{i,G+1} \leq f\left(X_{i,G}\right) \\ X_{i,G} \; otherwise \end{cases} \quad \text{(D6)}$$

Iteration continues until the maximum number of iterations is reached or the value of the objective function is less than a specified constant, e.g., 1E-6 .

Price *et al.*[11] have provided guidelines for choosing the parameters NP, CR, and F (p. 166).

Appendix D

Łukasiewicz Fitness Function

```
start lukasiewicz_obj_fcn( x, y ) ;
   /* Lukasiewicz objective function ("normal" (unweighted) Lukasiewicz structure)
    *
    * purpose: compute obj fcn value = cross-entropy for classes i using softmax() fcn
    *          to map similarities into probabilities
    *          range of obj fcn is [ 0, 1 ]
    *
    * parameters:
    *    x ::= 1 x D matrix of parameter vectors:
    *             1 row     of population parameters
    *             D columns of parameter values per row in x
    *
    *    y ::= <parameters> // <ideal vectors> // <feature vectors>
    *          <parameters>       ::= <n_classes> <n_feat>
    *          <ideal vector i>   ::= <ideal_i_feat_1> ...
    *                                 <ideal_i_feat_n_feat> *** n_classes rows
    *          <feature vector i> ::= <feat_i_value_1> ...
    *                                 <feat_i_value_n_feat>      *** n_cv_sample_scaled rows
    *
    * example of y matrix:
    *    n_classes n_feat <# of unique features>
    *    ideal_1_feature_1 ... ideal_1_feature_n_feat
    *    ideal_2_feature_1 ... ideal_2_feature_n_feat
    *    ...
    *    ideal_n_classes_feature_value_1  ... ideal_n_classes_feature_value_n_feat
    *    feature_vector_1_feature_value_1 ... feature_vector_1_feature_value_n_feat class_label_1
    *    feature_vector_2_feature_value_1 ... feature_vector_2_feature_value_n_feat class_label_2
    *    ...
    *    feature_vector_n_feature_value_1 ... feature_vector_n_feature_value_n_feat class_label_n
    */
```

```
   /* note: process only one pair (m, p) in x feat vector per iteration over classes n_classes */

   m = x[ 1 ] ; /* power parameter for generalized mean */
   p = x[ 2 ] ; /* power parameter for distance         */

   /* if either parameter <= 1e-4 then add penalty value */
   penalty = 0 ;

   if m < 1e-4 then do ; penalty =           1e6 ; m = 1 ; end ;
   if p < 1e-4 then do ; penalty = penalty + 1e6 ; p = 1 ; end ;

   n_feat      = y[ 1, 1 ] ; /* # of class means in class mean matrix        */
   n_classes   = y[ 1, 2 ] ; /* # of features in x matrix                    */
   uniq_class  = y[ 1, 2 + ( 1:n_classes ) ] ; /* # of unique class values in cv_sample_scaled */
   ideals      = y[ 1 + ( 1:n_classes ), 1 : n_feat ] ; /* n_classes rows, n_classes cols */
   feat_mat    = y[( 1 + n_classes + 1 ):nrow( y ), 1:n_feat ] ; /* feat vec rows, n_feat cols */

   feat_sim     = j( 1, n_feat,    0 ) ; /* for element-wise        similarity per class mean */
   similarity   = j( 1, n_classes, 0 ) ; /* for feature vector-wise similarity per class mean */
   softmax_prob = j( 1, n_classes, 0 ) ;

   /*****************************************************************************/

   /* randomly select one feature vector to which to apply parameters m, p */
   call randgen( ndx_row, 'Integer', nrow( feat_mat )) ;

   class_label = y[ 1 + nrow( ideals ) + ndx_row, ncol( y ) ] ; /* at end of input vec y */

   do j = 1 to n_classes ;
      /* compute similarities between individual feature values and respective class means */
      feat_sim = (( 1 - abs( feat_mat[ ndx_row,] ## p - ideals[ j,] ## p )) ## ( m / p ) ) ;

      /* compute generalized mean similarity btwn feat vector feat_sim and class mean vector j */
      similarity[ j ] = ( feat_sim[ : ] ) ## ( 1 / m ) ; /* sum feature similarities down cols */
   end ; /* j */

   softmax_prob = exp(similarity) / exp( similarity )[ + ] ; /* convert similarities to probs */

   /* compute entropy for similarity corresponding to class to which feature vector belongs */
   cross_entropy = -log( softmax_prob[ loc( class_label = uniq_class ) ] ) ;

   return cross_entropy + penalty ;
finish lukasiewicz_obj_fcn ;
```

Appendix E

Fuzzy Feature Selection

Fuzzy entropy was computed initially for all of the features in each dataset and the one with the highest entropy was removed. Fuzzy entropy was computed successively and the feature that contributed the most disorder was similarly removed. This process was repeated until only one feature remained.

The fuzzy entropic contribution of each feature was tabulated and is reported below along with the sub-sets derived from the groupings of features. 10-fold cross-validation was performed on the training data for each dataset.

Dataset: Wisconsin Breast Cancer (Diagnostic)

Var #	Var Name	Fuzzy Entropy	% Chg Fuzzy Entropy	Subset
30	Mean_smoothness	447.62	.	1
29	Worst_smoothness	338.84	-24.30	2
28	Mean_symmetry	331.33	-2.22	2
27	Worst_texture	292.57	-11.70	2
26	Mean_texture	288.74	-1.31	2
25	Mean_radius	285.51	-1.12	2
24	Mean_perimeter	280.41	-1.79	2
23	Worst_concave_pts	280.38	-0.01	2
22	Worst_symmetry	279.39	-0.35	2
21	Worst_radius	271.91	-2.68	2
20	Worst_perimeter	265.32	-2.42	2
19	Mean_fractile_dim	261.70	-1.36	2
18	Mean_compactness	258.10	-1.37	2
17	Stderr_concave_pts	257.71	-0.15	2
16	Worst_compactness	257.34	-0.15	2
15	Mean_area	252.87	-1.74	2
14	Mean_concave_pts	243.55	-3.69	2
13	Worst_fractile_dim	232.12	-4.70	2
12	Stderr_smoothness	230.54	-0.68	2
11	Worst_concavity	219.58	-4.75	2
10	Stderr_texture	214.71	-2.21	2
9	Stderr_symmetry	210.37	-2.02	2
8	Mean_concavity	209.90	-0.23	2
7	Stderr_compactness	193.31	-7.90	2
6	Worst_area	192.69	-0.32	2
5	Stderr_radius	190.54	-1.11	2
4	Stderr_fractile_dim	184.52	-3.16	2
3	Stderr_perimeter	158.62	-14.04	2
2	Stderr_concavity	139.81	-11.86	2
1	Stderr_area	106.29	-23.97	2

Table E1: Wisconsin Breast Cancer

Dataset: Credit Card Default

Var #	Var Name	Fuzzy Entropy	% Chg Fuzzy Entropy	Subset
23	Marriage	23844.80	.	1
22	Bill_amt1	20251.90	-15.07	2
21	Bill_amt6	19817.80	-2.14	2
20	Age	18743.40	-5.42	2
19	Education	17494.00	-6.67	2
18	Bill_amt4	16546.80	-5.41	2
17	Pay_1	15477.30	-6.46	2
16	Pay_2	15341.10	-0.88	2
15	Bill_amt5	13067.10	-14.82	2
14	Pay_5	13047.50	-0.15	2
13	Pay_6	12836.50	-1.62	2
12	Pay_3	12809.90	-0.21	2
11	Limit_bal	12579.10	-1.80	2
10	Bill_amt2	11926.30	-5.19	2
9	Bill_amt3	11019.80	-7.60	2
8	Pay_4	6371.23	-42.18	2
7	Pay_amt6	1411.61	-77.84	3
6	Pay_amt5	1384.76	-1.90	3
5	Pay_amt1	1253.72	-9.46	3
4	Pay_amt4	1179.70	-5.90	3
3	Pay_amt3	1112.78	-5.67	3
2	Pay_amt2	719.84	-35.31	3
1	Sex	0.00	-100.00	3

Table E2: Credit Card Default

Dataset: Dermatology

Var #	Var Name	Fuzzy Entropy	% Chg Fuzzy Entropy	Subset
34	Erythema	790.17	.	1
33	Scaling	746.63	-5.51	1
32	Itching	540.64	-27.59	1
31	Definite_borders	524.93	-2.91	1
30	Fibrosis	486.27	-7.37	1
29	Knee_and_elbow_involve	486.20	-0.01	1
28	Acanthosis	445.60	-8.35	1
27	Follicular_pap	432.46	-2.95	1
26	Fam_hist	425.22	-1.68	1
25	Melanin_incont	417.24	-1.88	1
24	Kobner_phenom	413.88	-0.81	1
23	Scalp_involve	405.00	-2.15	1
22	Polygonal_pap	343.91	-15.08	1
21	Hyperkeratosis	330.07	-4.02	1
20	Pnl_infilt	323.40	-2.02	1
19	Eosinophils_infilt	317.83	-1.72	1
18	Oral_mucosal_involve	282.80	-11.02	1
17	Exocytosis	263.16	-6.95	1
16	Parakeratosis	197.84	-24.82	1
15	Munro	197.34	-0.25	1
14	Clubbing	188.19	-4.64	1
13	Focal_hypergran	185.17	-1.60	1
12	Thinning	157.78	-14.79	1
11	Elongation	152.33	-3.46	1
10	Spongiform	139.36	-8.52	1
9	Spongiosis	104.26	-25.18	1
8	Inflamm_monoluclear_infilt	100.44	-3.66	1
7	Disappear_gran_layer	96.24	-4.18	1
6	Age	67.06	-30.32	1
5	Saw_tooth_appearance	59.58	-11.15	1
4	Band_like_infilt	59.30	-0.47	1
3	Perifollicular_para	58.07	-2.08	1
2	Follicular_horn_plug	57.29	-1.35	1
1	Vacuolisation	52.44	-8.45	1

Table E3: Dermatology

Dataset: Diabetes

Var #	Var Name	Fuzzy Entropy	% Chg Fuzzy Entropy	Subset
8	Bmi	634.99	.	1
7	Glucose	587.42	-7.49	1
6	Bloodpressure	539.10	-8.23	1
5	Pregnancies	368.48	-31.65	1
4	Age	367.07	-0.38	1
3	Skinthickness	350.29	-4.57	1
2	Diabetespedigreefunction	224.36	-35.95	1
1	Insulin	203.38	-9.35	1

Table E4: Diabetes

Dataset: Iris

Var #	Var Name	Fuzzy Entropy	% Chg Fuzzy Entropy	Subset
4	Petalwidth	161.08	.	1
3	Petallength	148.71	-7.68	1
2	Sepallength	132.84	-10.67	1
1	Sepalwidth	130.38	-1.86	1

Table E5: Iris

Dataset: Obesity

Var #	Var Name	Fuzzy Entropy	% Chg Fuzzy Entropy	Subset
16	Height	5247.46	.	1
15	Ncp	4876.60	-7.07	1
14	F_caec	4845.40	-0.64	1
13	Tue	4311.20	-11.02	1
12	Ch2o	4104.85	-4.79	1
11	F_gend	4039.62	-1.59	1
10	Faf	3218.38	-20.33	1
9	F_calc	3075.38	-4.44	1
8	F_favc	2804.94	-8.79	1
7	Fcvc	2784.97	-0.71	1
6	Weight	2349.86	-15.62	1
5	F_mtrans	2330.44	-0.83	1
4	Age	2329.31	-0.05	1
3	F_smoke	2166.01	-7.01	1
2	F_family_hist	2117.88	-2.22	1
1	F_scc	1309.90	-38.15	1

Table E6: Obesity

Dataset: Parkinson's

Var #	Var Name	Fuzzy Entropy	% Chg Fuzzy Entropy	Subset
22	Mdvp:flo(Hz)	148.64	.	1
21	Spread2	140.59	-5.41	1
20	D2	136.28	-3.07	1
19	Hnr	131.22	-3.71	1
18	Jitter:ddp	128.21	-2.29	1
17	Mdvp:fhi(Hz)	115.22	-10.13	1
16	Mdvp:ppq	109.00	-5.40	1
15	Class	108.40	-0.55	1
14	Rpde	86.42	-20.28	1
13	Mdvp:apq	82.48	-4.55	1
12	Mdvp:rap	78.15	-5.25	1
11	Spread1	77.04	-1.42	1
10	Mdvp:fo(Hz)	75.91	-1.46	1
9	Mdvp:jitter(%)	74.49	-1.87	1
8	Mdvp:shimmer(Db)	71.98	-3.37	1
7	Dfa	70.47	-2.11	1
6	Shimmer:apq3	69.52	-1.35	1
5	Shimmer:dda	68.19	-1.91	1
4	Mdvp:shimmer	56.35	-17.37	1
3	Nhr	50.71	-10.00	1
2	Shimmer:apq5	44.91	-11.44	1
1	Mdvp:jitter(Abs)	40.66	-9.46	1

Table E7: Parkinson's

Dataset: Wine Quality

Var #	Var Name	Fuzzy Entropy	% Chg Fuzzy Entropy	Subset
11	Volatile_acidity	11922.10	.	1
10	Ph	10979.10	-7.91	1
9	Sulphates	10039.50	-8.56	1
8	Residual_sugar	9982.52	-0.57	1
7	Total_sulfur_dioxide	9515.34	-4.68	1
6	Chlorides	8887.02	-6.60	1
5	Alcohol	8649.39	-2.67	1
4	Density	8083.91	-6.54	1
3	Citric_acid	8020.81	-0.78	1
2	Free_sulfur_dioxide	6408.63	-20.10	1
1	Fixed_acidity	6255.31	-2.39	1

Table E8: Wine Quality

References

[1] Zadeh, Lotfi, “Fuzzy Sets”, *Information and Control*, vol. 8 (3) San Diego. pp 338-353, 1965.

[2] Shannon, Claude Elwood, “A Mathematical Theory of Communication”, The Bell System Technical Journal 1948-07-01: Vol 27 Issue 3, 379-423, 1948.

[3] De Luca, A., and S. Termini, “A Definition of a Nonprobabilistic Entropy in the Setting of Fuzzy Sets Theory”, *Information and Control, 20, 301-312, 1972.*

[4] Jang, J.-S. R., Sun, C.-T., Mizutani, E., *Neuro-Fuzzy and Soft Computing*, Prentice-Hall, Inc., Upper Saddle River, NJ, 1997.

[5] https://en.wikipedia.org/wiki/Entropy_(information_theory)

[6] Luukka, Pasi, “Feature selection using fuzzy entropy measures with similarity classifier”, Expert Systems with Applications 38 (2011) 4600-4607.

[7] Luukka, P., Saastamoinen, K., Könönen, V. “A classifier based on the maximal fuzzy similarity in the generalized Łukasiewicz structure”. In *Proceedings of the FUZZ-IEEE 2001 Conference, Melbourne, Australia, 2001.*

[8] Sampo, Jouni, Luukka, P. “Similarity Classifier with Generalized Mean; Ideal Vector Approach”. In *Lecture Notes in Computer Science*, September, 2006.

[9] Luukka, P., Saastamoinen, K., Könönen, V., Turunen, E. “A Classifier Based on Maximal Fuzzy Similarity in Łukasiewicz Structure”, Aalto University, 2001.

[10] R. Storn and K. Price. “Differential Evolution - A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces”. ICSI Technical Report TR-95-012, March 1995.
Source: https://www.icsi.berkeley.edu/icsi/publication_details?n=952

[11] K. Price, Storn, R., and Lampinen, J., “Differential Evolution”, Springer -Verlag, Berlin, 2005.

Acknowledgements

We thank A. Getman-Pickering, J. King, J.T. Lehman, P. Luukka, J. Naraguma, and A. Scheeline for their generosity in taking the time to review this document and provide useful comments.

Contact Information

Your comments and questions are valued and encouraged. Contact the author at:

Name:	Ross Bettinger
Enterprise:	Consultant
E-mail:	rsbettinger@gmail.com