Chapter 6
Information-Theoretic Approaches

Max Garzon, Sambriddhi Mainali, and Kalidas Jana

Abstract An entirely different but extremely relevant approach to dimensionality reduction can be taken using a different criterion, namely quantifying the information content of the features involved, within themselves or in relation to others. It turns out that Shannon's definition of information yields surprisingly interesting reductions. This chapter discusses five major variations of this idea, including comparisons using the concept of mutual information previously used in statistics and machine learning.

The problem of reliable telecommunication across a noisy channel (such as a phone line or extraterrestrial space between planets) led Shannon to fundamental research, an objective definition of *information* and the well-known theory of error-detecting and error-correcting codes in his foundational paper [1]. The theory blossomed into the field of information theory. The key concept in this field, Shannon entropy, quantifies the degree of *uncertainty* of (complementary to information in) a random process. This metric provides the mathematical foundation for information-theoretic analyses of channel capacity that characterize the maximum amount of information that can be transmitted through a noisy channel, while allowing noise removal without loss of information [1]. It has been also interpreted as a measure of the degree of *randomness* and/or *diversity* in a stochastic process. (The concept of entropy itself can be traced back to [2], later used in physics for heat theory and thermodynamics by [3], but here it will only refer to Shannon entropy and be just referred to as entropy and denoted by the customary H.)

Independence between features can be quantified using Shannon's conditional entropy $H(X_1 \mid X_2)$ between two features X_1 and X_2. When this entropy is low,

M. Garzon (✉) · S. Mainali
Computer Science, The University of Memphis, Memphis, TN, USA
e-mail: mgarzon@memphis.edu; smainali@memphis.edu

K. Jana
Fogelman College of Business, Memphis, TN, USA
e-mail: kjana@memphis.edu

M. Garzon et al. (eds.), *Dimensionality Reduction in Data Science*,
https://doi.org/10.1007/978-3-031-05371-9_6

X_2 essentially determines X_1, and thus X_1 is not an informative feature and could be discarded in favor of X_1. Five major variations of this kind will be reviewed.

The goal of this chapter is to show that the concept of entropy can be very effective for dimensionality reduction in unexpected ways.

6.1 Shannon Entropy (H)

This section provides definitions of Shannon entropy, conditional entropy, and their interpretations, along with a description of software libraries that can be used when manual computation becomes prohibitively costly, for example, when working with large datasets. Shannon entropy affords a nonlinear and nongeometric approach to dimensionality reduction based on information theory. A feature selection strategy can be based on the concept of conditional Shannon entropy of a random variable. (Sect. 11.1 defines background concepts in statistics and probability.)

According to Shannon, the *information content* $I(a)$ provided by an observation $[X = a]$ (or just $X = a$) of a random variable (RV) X on a probability space with a sample space Ω is the real number

$$I(a) = -\log_2 (P(X = a)),$$

where $P(X = a)$ is the probability of the event $[X = a] = \{e \in \Omega : X(e) = a\} = X^{-1}(a)$ associated with an observation of a value a for X.

The Shannon *entropy* $H(X)$ of a discrete RV X is the expected value (mean) I of the information content of the observations of all possible values of X, i.e., if X takes on only a finite number of values $a_1, \ldots, a_n$ with corresponding probabilities $P_1, \ldots, P_n$, then the entropy of X is given by

$$H(X) = -\sum_{i=1}^{n} P(X_i = a_i)\ \log_2\ P(X_i = a_i)\ = -\sum_{i=1}^{n} P_i\ \log_2 P_i\,. \tag{6.1}$$

Since it takes just about $\log_2(n)$ bits to express an integer n in, say, binary, $I(a)$ amounts to the average number of bits necessary to remove the *uncertainty* in answering a question like *what is the value of the observation* of X? when performing the random experiment in the background probability space. This is the key idea in Shannon's definition of *information* (content).

Entropy can thus be regarded as a measure of the average uncertainty in determining any given outcome of an observation of X or its quantification as the average number of bits necessary to identify all possible (unique) values of X (such as the 2 outcomes of a Bernoulli trial, somewhere between 0 and 1 bit, or the 6 outcomes of the roll of a die, somewhere between 2 and 3 bits). (If natural logarithms are used, the unit is called "nats" not bits. Throughout this book, entropies are reported in bits.)

Example 6.1 (Entropy of a Bernoulli Trial) In a Bernoulli trial (defined in Sect. 11.1), i.e., a sample space Ω with just two outcomes (*success* and *failure*, or heads and tails, or simply 1 and 0) with probabilities, p and $1 - p$, respectively, the RV X with value $X = 1$ if and only if the outcome is a success ($X = 0$ if it is a failure), the probability distribution of X is $P(X = 1) = P$ and $P(X = 0) = 1 - P$. The entropy is

$$H(X) = -P \log_2(P) - (1 - P) \log_2(1 - P).$$

□

Figure 6.1 shows the graph of the Shannon entropy $H(X)$ of the Bernoulli RV X. It is a concave function of P that attains a minimum value of zero for $P = 0$ and $P = 1$ and reaches a maximum value of 1 bit at $P = 0.5 = 1 - p$. Thus, the entropy is 0 when the outcome of the trial is a sure event (implying that there is no uncertainty in the outcomes of the random experiment), but the entropy is maximum when the outcomes are equally likely.

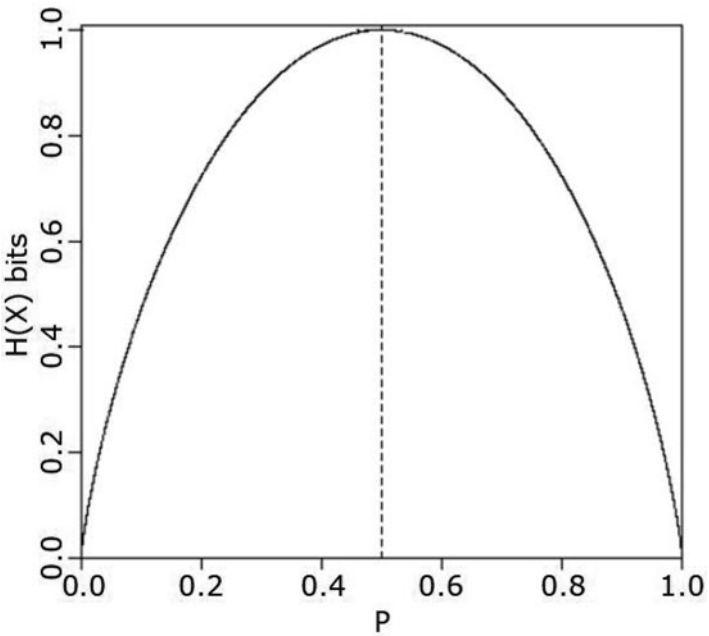

Fig. 6.1 For a Bernoulli random variable X with just two outcomes, the uncertainty is maximum $H(X) = 1$ when the two outcomes are equally likely (with probability $P = \frac{1}{2}$) and it is minimum when one of them is certain ($P = 1$, hence the other impossible $1 - P = 0$, or vice versa)

Example 6.2 (Entropy of a Roll of a Fair Die) If a die is fair, then a throw of the die has 6 equally likely outcomes, each with probability 1/6. Therefore, the Shannon entropy in this case is

$$H(X) = -\sum_{i=1}^{6} P_i \ \log_2 P_i = -\sum_{i=1}^{6} \frac{1}{6} \ \log_2(\frac{1}{6}) = \log_2 6 = 2.56 \text{ bits}.$$

□

There is more uncertainty in this stochastic process than in a Bernoulli trial because $2.58 > 1$, which makes sense intuitively because there are now $6 > 2$ possibilities. In other words, knowing the outcome of a dice roll is more informative (removes more uncertainty) than knowing the outcome of a coin toss indeed!

If two RVs X and Z on the same sample space are correlated (e.g., the value up on one of the dice in a roll of two and the sum of the two values for both dice), it is to be expected that knowing the value of the outcome of the corresponding experiment (rolling the dice) for the observations for one of them will reduce uncertainty in the value of the other. This reduction can be quantified precisely as follows:

> The *conditional entropy* $H(Z \mid X)$ of a RV Z on (or *relative to*) another RV X is the average entropy of Z conditioned on the observation of a value of X, i.e., the expected value $E(H(Z_a))$ of the entropies of the RVs given by $Z_a : [Z \mid X = a]$, across all possible values of a in the range of X.

Example 6.3 (Bivariate RVs) For conditional entropy for two scalar discrete RVs X and Z, let X take on two values 1 and 2 and Z take on three values 1, 2, and 3, with a joint probability distribution of X and Z given in Table 6.1. The Shannon entropy of Z is $H(Z) = 1.56$ bits. □

Thus, given the joint probability distribution (defined in Sect. 11.1) of a sample of observations of X and Z, where X takes values $x_1, \ldots, x_m$ and Z takes values $z_1, \ldots, z_n$, the conditional entropy of Z given X is

$$\begin{aligned} H(Z \mid X) &= \sum_{i=1}^{m} P(X = x_i) H(Z \mid X = x_i) \\ &= -\sum_{i=1}^{m} P(X = x_i) \sum_{j=1}^{n} P(Z = z_j | X = x_i) \ \log_2(P(Z = z_j | X = x_i)). \end{aligned}$$

Table 6.1 Shannon entropy of Z and conditional entropy of Z given X for the joint probability distribution in the entries

$Z \setminus X$	1	2
1	0.15	0.10
2	0.20	0.20
3	0.30	0.05

$H(Z) = 1.56$ bits
$H(Z \mid X) = 1.48$ bits

This conditional entropy can be interpreted as the *uncertainty left in the values of Z given the observations of covariate feature X* in a given data point Z_{x_i}, averaged across all data points x_i .

Example 6.4 (Conditional Entropy) If the joint probability distribution is as given in Table 6.1, the conditional entropy of Z given X is $H(Z \mid X) = 1.48$ bits. □

The conditional entropy can be generalized to any number of conditions, i.e., $H(Z \mid X_1, X_2, \cdots, X_m)$ can be interpreted as the average uncertainty left in feature Z given the values of joint observations $X_1 = x_1, X_2 = x_2, \cdots, X_m = x_m$ in the same data points.

This definition of conditional entropy will be applied in Sect. 6.4 to achieve dimensionality reduction to 12 or 6 features in a dataset of 345 malware features taken from a sample of Microsoft's Malware Classification dataset with 1805 features (described in Chap. 1 and Sect. 11.4). Because the sample dataset is big, manual computation of conditional entropy is prohibitively costly. Software will come in handy to make computation manageable for such big datasets. One such software is the R package `infotheo`. It computes Shannon entropy using the Miller–Madow asymptotic bias corrected empirical estimator (which requires discretized features, and the option `equalfreq` can be used to discretize the data, where necessary, as illustrated in Sect. 6.4).

The Shannon entropy is also defined for continuous random variables as well. However, currently available computing software such as `infotheo` uses only discretized data even if it is continuous at the source. This is not really an issue since most data nowadays are collected with digital sensors and are thus discrete. Otherwise, truly continuous data has to be processed on conventional computers and so it has to be digitized in the process anyway.

6.2 Reduction by Conditional Entropy

This section and the next introduce alternative methods to use Shannon's entropy to reduce dimensionality in datasets with too many features, along with an assessment of their effectiveness in preserving significant information from the original dataset. The key idea here is to select more informative features or remove features whose information content is determined by the selected features, as determined by Shannon conditional entropy.

Example 6.5 The problem [**MalC**] of Malware Classification calls for an assignment of types to malwares from a predetermined set of types. It is a major problem in cyber security, where new malware is emerging at an alarming rate (for instance, more than 4.62 million new instances of malicious code were detected in June–July, 2019 [4].) A rigorous analysis to this problem is critical to study the evolution of malware and in developing appropriate countermeasures to contain cybercrime. Such an analysis can be either static or dynamic. A dynamic analysis depends on the

execution of malware in a controlled environment [5, 6] and is costly effortwise [7]. On the other hand, a static analysis relying on decompilation tools (like IDA Pro) is more effective and efficient [8, 9]. However, it suffers from a major information retrieval issue since important information (like code layout, meta annotations, and even source language) in the source code is usually lost in the compilation process and cannot be retrieved in decompilation. □

This section addresses the problem only indirectly by identifying important features to classify a piece of malware into categories for which known countermeasures are available. There are just too many features that can be associated with a piece of malware (e.g., 1809 in the Microsoft's dataset, described in Sect. 11.5.) Variants of entropy methods can be used to reduce the dimensionality of pre-identified features of these malwares to solve the classification problem. A first reduction by Ahmadi [10] (*arxiv.org/abs/1802.10135*) in 2016 used only 344 features extracted from the original Microsoft Malware Classification Challenge containing more than a thousand features [11].

The versatility and effectiveness of these methods can be illustrated with a second type of problem, the noisy classification problem (described in detail in Sect. 11.3.)

Example 6.6 One question of interest in any approach to DR is how good the approach is at identifying dependencies (statistical or other) in the features in the dataset. In a controlled experiment, a synthetic set can be designed with perfect knowledge of these dependencies from independent (e.g., randomly generated) raw (primitive) features. The effectiveness of DR methods can then be assessed by how well they discover these hidden, yet most relevant and independent features from a full set that includes other confounding features derived from the few primitive ones.

The primitive features were generated using the method described in [12] and publicly available as an API in Python at *sklearn.datasets.make_classification*. The primitive features are mixtures of several Gaussian clusters located near corners of the 12D hypercube and correspond to the labels in a classification problem. For each class label, the informative features are drawn independently from the standard normal distribution $N(0, 1)$. Four more dependent features were added as various linear combinations (with random coefficients) of the primitive features, as provided in the API. In the second phase, 6 more predictors were generated as repeats of two randomly selected (but uniform for all data points) primitive features/columns. One more feature was added as the sum of squares of two features selected randomly, one more feature consisting of the values of the predictions of a linear regression model fitted using two other randomly selected features as predictors, and the next feature was the deviation in the prediction from the true value. The last predictor was obtained in a similar manner but using the squares of the randomly chosen feature to predict one from the other. One last feature was generated as the outcome of the natural *logarithm* of a randomly selected but uniform predictor (a transformation that does not change entropy), for a total of 22 predictors. (A detailed description of how the synthetic datasets were generated is given with the datasets SYN12 and SYN23 in Sect. 11.5.)

A second dataset was generated likewise but halving the number of all parameters involved in generating features in the first set. □

These datasets are used in Sect. 6.3 as well for assessment of the variant of the method being discussed there.

For this variant of entropy methods, it is important to quantify the informational independence between two predictors, unlike a dependent variable and a predictor used in conventional methods. Such quantification is done using Shannon conditional entropy between two predictors X_1 and X_2, i.e., $H(X_1 \mid X_2)$. When this entropy is high, knowledge of the values of X_2 removes little of the uncertainty in the values of X_1 and selection of X_1 may be necessary even if X_2 has been included [13]. Conversely, when this entropy is zero (or very low), X_2 (essentially, respectively) determines X_1, and thus X_2 is an informative feature and would suffice because models in a solution could derive any information in X_1 from it. This concept is analogous to the concept of multicolinearity in regression models described in Sect. 2.1. Multicolinearity is the condition that occurs if a few or all predictors are linearly correlated. As with multicolinearity, the information dependence between predictors is undesirable.

This section illustrates the application of two variants of this conditional entropy first introduced in [13]. A few significant predictors from the full list of predictors are obtained that possibly help in ascertaining target feature values (e.g., malware type.) To use as few Xs as possible, only predictors informative of the target are of interest. In particular, they should be as independent from other predictors as possible. Information-theoretically, that means that the conditional entropies relative to the other predictors (i.e., the uncertainty left in the predictor given another predictor's value) should be high overall. Thus, to decide whether to include X_i, the average is computed as

$$avg_i = avg \ \{H(X_i \mid X_j) : 1 \leq j \neq i \leq m\}$$

of the $H(X_i \mid X_j)$ over all other X_j to quantify how informative predictor X_i is as compared to others, for each i in the range of p predictors (excluding the target feature). For a given dataset, as many features as desired can thus be selected from the top features *maximizing* this average, after sorting the list in decreasing order.

A second paired variant of the same idea is to use the double conditional entropy to compute $H(X_i \mid X_j, X_k)$, for all j and k to determine how informative X_i is given the pair X_j, X_k when compared to all other pairs. If the average of these entropies is high, X_i should be selected, as above. The same process can be repeated to select more features as needed.

To compute conditional entropies between two predictors, `infotheo` package available in R was used, as mentioned above. (The package can be installed using the "install.packages(`infotheo`)" command in the R-console, as described in Sect. 11.6.) The Microsoft Malware Classification dataset was used to select two datasets using each variant. The first dataset contained only six predictors and the second contained 12 predictors. Then, these datasets were fed to several statistical

and machine learning algorithms to assess the performance of these variants based on the performance of the resulting solutions for the classification problems on the datasets SYN12 and SYN23.

To assess the quality of this method of DR, one can proceed in two ways. First, the criterion being used (comparison of the information-theoretic content of the features) provides a good rationale why the choices may be effective and interpretable. However, there remains the issue of whether Shannon entropy really defines information as humans conceive of it, a much harder question that has remained unanswered. Alternatively, one can compare the effect of the choices on how good solution models are (as described in Sect. 2.4) when obtained on various sets of predictors. Thus, machine learning models trained on these reduced sets of predictors were compared against the scores yielded by machine learning models trained on the full set of predictor features, for both the Malware and synthetic datasets.

Following the standard procedure described above in Sect. 2.4, these datasets were split into training and testing subsets in a proportion of 80%–20%. The machine learning models included Linear Discriminant Analysis (LDA), Support Vector Machine (SVM), Multinomial Logistic Regression (MLR), Gaussian Naive-Bayes (GNB), Random Forest (RF), k-Nearest Neighbors (**kNNs**), and Neural Networks (NNs). The statistical models (the first four) were implemented using R scripts and the machine learning models were implemented using Python code (as in Sect. 11.6.) As usual, the F1-scores were selected as the metric to assess the quality of the solution models trained and these scores and are shown in Fig. 6.2 and Table 6.2 in the next section.

The methods have been used with similar success on other datasets, including [**BioTC**] and the synthetic datasets, selecting sets of various sizes (6 and 12 features) as well, as described in [13]. In terms of solutions, some models (RFs, kNNs) perform significantly better regardless of the problem (**[MalC]**, **[BioTC]**, or **[NoisC]**), while others (SVMs, LDAs, and MLRs) perform very inconsistently across these three representative problems. On average, solution models using only 6 features mostly performed poorly as compared to those using 12 features. Six (6) features might be too few for solving a complex problem, so reducing features too much may hurt for abiotic data.

Moreover, these DR methods offer the additional advantage of being computationally efficient because they are parallelizable. To compute conditional entropies between a pair of features (either two predictors or one predictor plus a target), the information about other features is not required, and hence several disjoint subsets of data can always be extracted and assigned to different computing nodes. This makes the process of feature extraction feasible by parallelization, even for big datasets.

In summary, conditional entropy performs competitively, if not very well, across the board compared to random selection of features or other methods.

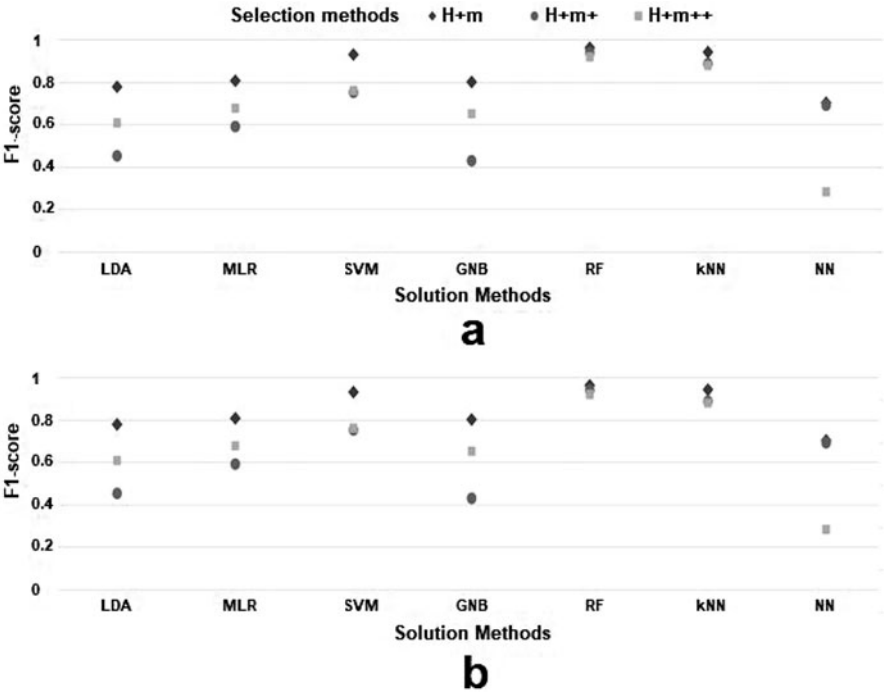

Fig. 6.2 Overall performance of DR to (**a**) 6 features (*top*) and (**b**) 12 features (*bottom*) based on conditional entropy on the problem of Malware Classification [**MalCP**]. Conditional entropy performs competitively, if not very well, with most machine learning solutions, The DR methods are selection by conditional entropy on predictors only (single H+m, paired H+m+, iterated H+m++) excluding targets, except LDA, GNB, and NN for paired iterative entropies (H++)

6.3 Reduction by Iterated Conditional Entropy

The results of Shannon conditional entropy for DR discussed in the previous section suggest the possibility that the interaction between various predictors at various levels might (perhaps jointly) contain better information about other predictors (which might have been deemed as most informative by themselves.) Therefore, another interesting variation is to select features based on a recursive procedure in which earlier choices affect together later selections.

In order to demonstrate the effectiveness of this alternative method for DR, the same procedure problems and datasets presented in the examples in Sect. 6.2 were used to assess the quality of machine learning models being trained using features selected by this alternative (indicated by H+m++). The scores are presented in Figs. 6.2 and 6.3, together with previously discussed variants of conditional entropy. On average, machine learning models seem to give better scores when the dimensionality reduction is done using the iterated conditional entropy method on the malware dataset. There seems to be no significant difference in the performance

Table 6.2 Performance comparison (F1-scores) between machine learning models trained using all features (*last column*) and those trained using reduced features (*second* and *third columns*) from the large synthetic dataset SYN23 for the problem of Noisy Classification **[NoisC]**

DR variant	Solution	Reduced	All features
All features	LDA		0.6937
	MLR		0.8013
	SVM		0.8690
	GNB		0.5085
	RF		0.9291
	kNN		0.9680
	NN		0.6860
H+m (Conditional)	LDA	0.9402	
	MLR	0.5728	
	SVM	0.8104	
	GNB	0.9096	
	RF	0.9717	
	kNN	0.9460	
	NN	0.8726	
H+m+ (Paired Conditional)	LDA	0.9484	
	MLR	0.9345	
	SVM	0.9902	
	GNB	0.9425	
	RF	0.9800	
	kNN	0.9578	
	NN	0.9082	
H+m++ (Iterated)	LDA	0.9481	
	MLR	0.9345	
	SVM	0.9906	
	GNB	0.9547	
	RF	0.9794	
	kNN	0.9910	
	NN	0.9417	

of machine learning models trained on the features selected using single and paired conditional entropies. Moreover, these models seem to give better performance when 12 reduced features are used as predictors instead of 6. On the other hand, the running times on an HPC of the relative entropy calculations to select features are in the order of minutes (single entropies), under 2 h (paired entropies) and under 6 h (iterated entropies.) So, there is a trade-off between the performance of the machine learning models and the computational resources required to select features using these information-theoretic DR methods.

In order to demonstrate that feature selection using entropy methods without target is most likely an optimal choice than using the whole dataset, the methods was tried also on the synthetic datasets mentioned above (described in Sect. 11.5).

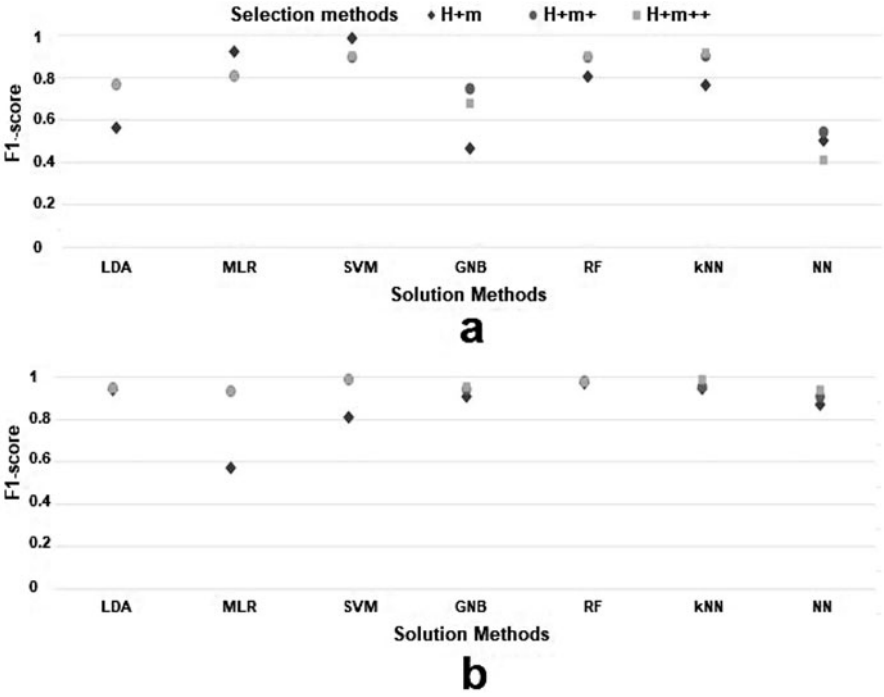

Fig. 6.3 Overall performance of DR to (**a**) 6 features (*top*) and (**b**) 12 features (*bottom*) based on conditional entropy on the problem of Synthetic Data Classification [**NoisC**] (with controlled dependencies) on synthetic datasets SYN12 and SYN23. Conditional entropy performs competitively, if not very well, with most machine learning solutions. The DR methods are selection by conditional entropy on predictors only (H+m, paired H+m+, iterated H+m++) excluding targets, except LDA, GNB, and NN for paired iterative entropies (H++)

Standard metrics like accuracy, precision, recall, and F1-score were used to evaluate the performance of different ML models trained and validated on these datasets. A dimensionality reduction method was deemed to be of a good quality if an average performance score of several machine learning models trained to solve these problems is above 81% (average of the scores reported in [13]) for both problems.

A comparison of the scores is shown in Table 6.2. As described above, the synthetic dataset was designed so as to contain some predictors that could be derived using some kind of combination of other predictors. Therefore, these features are not informationally independent. Although the impact of this dependency is not so evident in some machine learning models on the given dataset, the usage of all features led to solution models with the performance being dominated by those trained using reduced features on average. Therefore, as a general rule of thumb, one can easily conclude that in the presence of too many predictors, it is always a wise choice to look out for some dimensionality reduction methods to obtain only informationally rich features. However, there is still a question that

remains unanswered, i.e., what is the threshold determining too many predictors? The answer to this question really ultimately depends on the type of data science problem at hand and the choice of features in the dataset. One can only hope to consider all constraints impacting the search for a solution, for example, answering the following questions (perhaps among others):

- Is it computationally feasible to include all features available?
- Is there enough time to train a model with all features and validate it?
- Are there enough datapoints to include all features (for example, a dataset containing only 10 points might be enough to train a statistical model if there is only one predictor, but not when more than six predictors are to be considered)?

As with conditional entropy in Sect. 6.2, the methods have been used with similar success on other datasets, including one for BioTC and the synthetic datasets SYN12 and SYN23, selecting sets of various sizes (6 and 12 features) as well, as first described in [13]. In terms of solutions, some models (RFs, kNNs) perform better regardless of the problem (**[MalC]**, **[BioTC]**, or **[NoisC]**), while others (SVMs, LDAs, and MLRs) perform very inconsistently across these three representative problems. On average, solution models using only 6 features performed mostly poorly as compared to those using 12 features. Six (6) features might be too few for solving a complex problem, so reducing features too much may hurt for abiotic data.

In terms of the computational efficiency, the advantage of being parallelizable is not as impressive. Because of their nature, to compute iterated conditional entropies between a pair of features (either both predictors or one predictor plus another target), data from other features is now required and the number of possible combinations is explosive. These facts make the process of feature extraction feasible less attractive, particularly for big datasets. Perhaps, a combination of the two methods, first selecting a smaller subset using conditional entropy and then using iterated conditional entropy of the smaller subset, may be a more productive approach.

In summary, conditional entropy and iterated conditional entropy perform competitively, if not very well, across the board compared to random selection of features or other methods.

6.4 Reduction by Conditional Entropy on Targets

This section shows how dimensionality reduction of predictor features can be achieved using conditional entropy of *the target feature* relative to the predictors. Two examples are used to illustrate reductions to a set of 6 or 12 features out of 344 singles features and 6 or 12 paired features in a sample dataset of 345 features from the Microsoft's Malware Classification dataset of 1805 features for predicting the target feature "Class" (type of malware). As a result, a principled argument can be made in Sect. 6.5 to show how selecting features by minimizing conditional entropy

is equivalent to selecting features by maximizing mutual information with respect to the target feature, the commonly used approach in the literature on information-theoretic methods for feature selection.

In statistics, when selecting features for predictors, the target feature is naturally taken into account. In this variant, the degree with which a target Y is predictable given a feature X_i is decided using conditional entropy $H(Y \mid X_i)$ between Y on a feature X_i. When this entropy is low (high), the uncertainty is low (high) for the values of Y given the values of X_i, so X_i is informative (or is not, respectively.) Hence, X_i should be selected if this entropy is low. This is the opposite of the criterion in Sects. 6.2 and 6.3 when predictors are being compared for selection, but the approach is implemented just the same way, with the following changes:

- Calculate the $H(Y \mid X_i)$ over all X_i to quantify how informative predictor feature X_i is compared to others.
- Select the top features minimizing this entropy after sorting the list in increasing order.

As mentioned in Sect. 6.1, the conditional entropy $H(Z \mid X)$ can be generalized to any number of conditions $X_1, X_2, \cdots, X_m$ instead of a single condition X (Table 6.3).

Example 6.7 Table 6.4 shows 12 predictor features selected based on

$$H(Class \mid X_1, X_2, \cdots, X_m)$$

Table 6.3 Top 12 *single* features X_i $(i = 1, \cdots, 12)$ (*second column*) selected by sorting 344 values of conditional entropies $H(Class \mid X_i)$ in a sample dataset of 345 features taken from the Microsoft Malware Classification dataset of 1805 features for predicting the target feature "Class." Table 6.4 shows the values of the entropies

ID	Selected feature
1	db0_por
2	dbN0_por
3	db3_all
4	db3_rdata
5	dc_por
6	ent_p_1
7	ent_p_2
8	ent_p_4
9	ent_q_diff_diffs_10
10	ent_q_diff_diffs_11
11	ent_q_diff_diffs_1_mean
12	ent_q_diff_diffs_2_mean
13	ent_q_diff_diffs_2_min
14	ent_q_diffs_0
15	ent_q_diffs_mean
16	ent_q_diffs_var
17	known_Sections_por

Table 6.4 Top single conditional entropies $H(Class \mid X_i)$ (entries in a row i in the second column) and top conditional entropies $H(Class \mid X_i, X_j)$ (if more than one entry in a row in columns $j \geq 1$) of the top 12 *pairs* (X_i, X_j) obtained by sorting $\binom{344}{2} = 58{,}996$ values of $H(Class \mid X_i, X_j)$ corresponding to 58,996 pairs of the features in Table 6.3 used to solve the Malware classification problem [**MalC**]. A blank entry indicates that the corresponding feature was not selected, either as a single i or as part of a pair i, j

$i : j$	$H(Class \mid X_i)$	1	2	3	4	5	···	17
1								
2								
3								
4								
5	1.756							
6	1.765							
7	1.756							
8	1.756							
9	1.521							0.898
10	1.548	0.913	0.916	0.882	0.880	0.878		0.889
11	1.747							
12	1.745							
13	1.681							
14	1.676							
15	1.566				0.908	0.874		0.875
16	1.586		0.915	0.909				
17								

with $m = 1$ and 2, respectively, for predicting the target "Class", i.e., the 345th feature in the sample dataset of 345 features taken from the population of Microsoft's Malware Classification dataset of 1805 features, by implementing the above two-step procedure using `infotheo`. It is worth mentioning that the number 12 should not be interpreted as any sort of "optimal" number of features in the sense of being determined by some optimality criterion. Rather, it is just a low number chosen for the purpose of comparison with choosing fewer features. □

To assess the quality of this reduction, various machines learning solutions were computed for the Microsoft's malware dataset for the [**MalC**] problem, similar to the procedure for the previous variants of the entropy method. Table 6.5 shows the results. Furthermore, various machines learning solutions were also computed on synthetic datasets SYN12 and SYN23 (described in Sect. 11.5) with primitive features hidden from the conditional entropy reduction method. Tables 6.6 and 6.7 show the F1-scores for comparison.

In summary, a careful comparison with the other conditional entropy methods shows that, perhaps surprisingly, *taking into account the targets does not really seem to make such a significant difference* in the quality of the solutions to a data science problem. However, choosing 12 features rather than 6 again does help significantly

Table 6.5 Performance (F1-scores) comparison between machine learning models trained using 6 and 12 singles predictor features and 6 and 12 paired predictor features, selected by conditional entropy of the target feature "Class" in the Malware Classification dataset. (Conditional entropy of singles and paired features are denoted by H and $H+$, respectively)

Entropy variant	ML solution model	6 features	12 features	All features
All features	LDA			
	MLR			0.6934
	SVM			0.3772
	GNB			0.7013
	RF			0.8338
	kNN			
	NN			
	Adaboost			0.7802
H	LDA	0.3579	0.4624	
	MLR	0.6433	0.8398	
	SVM	0.7618	0.8152	
	GNB	0.3189	0.5871	
	RF	0.9746	0.9869	
	kNN	0.9593	0.9690	
	NN	0.6949	0.8397	
H+	LDA	0.7087	0.7946	
	MLR	0.8230	0.9217	
	SVM	0.8398	0.9223	
	GNB	0.1072	0.6151	
	RF	0.9883	0.9895	
	kNN	0.9125	0.9323	
	NN	0.7447	0.9584	

improve the quality of the solutions. Pairing now, however, does seem to make a difference as well, in general.

6.5 Other Variations

A common information-theoretic approach used in statistics and data science, in particular for feature selection, makes use of the concept of *mutual information* $I(Y : X)$.

The *mutual information* (also known as *information gain*) is given by

$$I(Y : X) = H(Y) - H(Y \mid X).$$

This concept rather quantifies information independence between two variables, analogously to the concept of statistical independence.

Table 6.6 Performance (F1-scores) comparison between machine learning models trained using 6 and 12 singles predictor features, and 6 and 12 paired predictor features, selected by conditional entropy of the target feature "Class" in the Noisy Classification dataset SYN13. (Conditional entropy of singles and paired features are denoted by H and $H+$, respectively)

Entropy variant	ML solution model	6 features	12 features	All features
All features	[14]			0.7970
	[15]			0.9600
H	LDA	1	1	
	MLR	0.9805	0.9828	
	SVM	0.9912	0.9946	
	GNB	0.8598	0.9358	
	RF	0.9806	0.9865	
	kNN	0.9840	0.9709	
	NN	0.9153	0.9826	
H+	LDA	1	1	
	MLR	0.9811	0.9830	
	SVM	0.9922	0.9946	
	GNB	0.8598	0.9358	
	RF	0.9809	0.9867	
	kNN	0.9840	0.9709	
	NN	0.9232	0.9807	

Table 6.7 Performance (F1-scores) comparison between machine learning models trained using 6 and 12 singles predictor features, and 6 and 12 pairs predictor features, selected by conditional entropy of the target feature "Class" in Noisy Classification dataset (SYN22). (Conditional entropy of singles or paired features is denoted by H or $H+$, respectively)

Entropy variant	ML solution model	6 features	12 features	All features
All features	[14]			0.7970
	[15]			0.9600
H	LDA	0.5635	0.9402	
	MLR	0.9246	0.5728	
	SVM	0.9865	0.8104	
	GNB	0.4654	0.9096	
	RF	0.8063	0.9717	
	kNN	0.7657	0.946	
	NN	0.5029	0.8726	
H+	LDA	0.9975	1	
	MLR	0.9292	0.9849	
	SVM	0.9875	0.9783	
	GNB	0.9940	1	
	RF	0.9910	9895	
	kNN	0.7646	0.9511	
	NN	0.7033	0.9779	

Example 6.8 The mutual information in the variables Z and X in Example 6.3 in Sect. 6.1 above is

$$I(Z : X) = H(Z) - H(Y \mid X) = 1.56 - 1.48 = 0.08 \text{ bits} .$$

□

Example 6.9 In the **[MalC]** problem, there are quite a few predictors in the original dataset (1809 to be exact). Even after using the reduced dataset by [10] (344 predictors), there are still too many, and they may hide a number of dependencies (e.g., some of these features might be collinear). One could alternatively follow the approach discussed in previous sections, but using the mutual information as a selection criterion instead of conditional entropy. □

A comparison of the definitions of conditional entropy and mutual information makes it is clear that they are complementary quantities in the entropy $H(Y)$ of the target Y, so that the lowest conditional entropy corresponds to the highest mutual information with a predictor X, and vice versa. Therefore, selecting features by maximizing mutual information, for any problem and dataset, is equivalent to selecting features by minimizing conditional entropy, as has been done in Sects. 6.1–6.4.

References

1. Shannon, C. E. (1948). A note on the concept of entropy. *Bell System Technical Journal*, *27*(3), 379–423.
2. Clausius, R. (1879). *The mechanical theory of heat, nine memoirs on the development of concept of "entropy"*. McMillan and Co.
3. Boltzmann, L. (1878). On some problems of the mechanical theory of heat. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, *6*(36), 236–237.
4. Yang, P., Zhou, H., Zhu, Y., Liu, L., & Zhang, L. (2020). Malware classification based on shallow neural network. *Future Internet*, *12*(12), 219.
5. Fattor, A., Lanzi, A., Balzarotti, D., & Kirda, E. (2015). Hypervisor-based malware protection with access miner. *Computers & Security*, *52*, 33–50.
6. Hashemi, H., Azmoodeh, A., Hamzeh, A., & Hashemi, S. (2017). Graph embedding as a new approach for unknown malware detection. *Journal of Computer Virology and Hacking Techniques*, *13*(3), 153–166.
7. Pektaş, A., & Acarman, T. (2017). Classification of malware families based on runtime behaviors. *Journal of Information Security and Applications*, *37*, 91–100.
8. Fan, C. I., Hsiao, H. W., Chou, C. H., & Tseng, Y. F. (2015). Malware detection systems based on API log data mining. In *2015 IEEE 39th Annual Computer Software and Applications Conference* (Vol. 3, pp. 255–260). IEEE.
9. Santos, I., Brezo, F., Ugarte-Pedrero, X., & Bringas P. G. (2013). Opcode sequences as representation of executables for data-mining-based unknown malware detection. *Information Sciences*, *231*, 64–82.
10. Ahmadi, M., Ulyanov, D., Semenov, S., Trofimov, M., & Giacinto, G. (2016). Novel feature extraction, selection and fusion for effective malware family classification. In *Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy* (pp. 183–194).

11. Ronen, R., Radu, M., Feuerstein, C., Yom-Tov, E., & Ahmadi, M. (2018). Microsoft malware classification challenge. *CoRR*, abs/1802.10135.
12. Guyon, I. (2003). Design of experiments of the nips 2003 variable selection benchmark. In *NIPS 2003 Workshop on Feature Extraction and Feature Selection* (Vol. 253).
13. Mainali, S., Garzon, M., Venugopal, D., Jana, K., Yang, C. C., Kumar, N., Bowman, D., & Deng, L. Y. (2021). An information-theoretic approach to dimensionality reduction in data science. *International Journal of Data Science and Analytics*, *12*, 185–203. https://doi.org/10.1007/s41060-021-00272-2
14. Donoho D. L. (2000). High-dimensional data analysis: The curses and blessings of dimensionality. In *AMS Conference on Math Challenges of the 21st Century*.
15. Achlioptas, D. (2003). Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of Computer and System Sciences*, *66*(4), 671–687.