On Estimating Mutual Information for Feature Selection

Erik Schaffernicht[1], Robert Kaltenhaeuser[1],
Saurabh Shekhar Verma[2], and Horst-Michael Gross[1]

[1] Neuroinformatics and Cognitive Robotics Lab,
Ilmenau University of Technology, Germany
Erik.Schaffernicht@tu-ilmenau.de
[2] College of Technology, GBPUAT, Pantnagar, India

Abstract. Mutual Information (MI) is a powerful concept from information theory used in many application fields. For practical tasks it is often necessary to estimate the Mutual Information from available data. We compare state of the art methods for estimating MI from continuous data, focusing on the usefulness for the feature selection task. Our results suggest that many methods are practically relevant for feature selection tasks regardless of their theoretic limitations or benefits.

Keywords: Mutual Information, Probabilitty Density Estimation, Feature Selection.

1 Introduction

Mutual Information (MI) is a well known concept from information theory and has been utilized to capture the dependence structure between pairs of random variables X and Y. In contrast to approaches like correlation coefficients MI is not limited to the linear dependencies but includes any nonlinear ones. In an information theoretic sense, MI quantifies the information variable X contains about Y and vice versa.

Identifying relevant features for a given learning problem in order to eliminate irrelevant and redundant inputs that complicate the learning process is defined as the feature selection task. Applying Mutual Information to calculate the relevance of a given input channel is a very intuitive and common approach. In its most simple form it allows a feature ranking, but there are more sophisticated filter approaches based on MI.

The practical challenge of using Mutual Information for feature selection is the estimation of this measure from the available data. Similar to [1], we compare different approaches of estimating MI, but in contrast we include new approaches for estimation and focus on the feature selection task.

A brief recap of all the considered methods will be given in the next section. The results of our tests will be shown in section 3, where we draw conclusion about the usefulness of different methods for feature selection.

K. Diamantaras, W. Duch, L.S. Iliadis (Eds.): ICANN 2010, Part I, LNCS 6352, pp. 362–367, 2010.
© Springer-Verlag Berlin Heidelberg 2010

2 Methods for Estimating Mutual Information

The goal is to estimate the mutual information, which is given by

$$I(X;Y) = \int \int p(x,y) log_2 \frac{p(x,y)}{p(x)p(y)} dxdy \tag{1}$$

In this paper, we consider the histogram estimation with Scott's rule [2], Cellucci's adaptive partitioning of the XY plane [3], an approach using an ensemble of histograms, kernel density estimation (KDE) [4], least-squares mutual information (LSMI) [5] and k-nearest neighbor estimation (K-NN) [6].

This selection is far from complete, literature thrives with other methods, but it captures the intuitive methods (Histogram and KDE) as well as the current standard method (K-NN) and a very recent method that claims superiority to this (LSMI).

2.1 Histogram Approach

The standard histogram partitions the axes into distinct bins of width w_i and then counts the number n_i of observation falling into the bin i. In order to turn this count into a normalized probability density, we simply divide by the total number N of observations and by the width w of the bins to obtain probability values

$$p_i = \frac{n_i}{N w_i} \tag{2}$$

for which $\int p(x)dx = 1$. This gives a model for the density $p(x)$ that is constant over the width of each bin.

The mutual information between X and Y given by eqn.1 changes to eqn.3

$$I(X;Y) = \sum_i \sum_j P_{ij} \, log_2 \left(\frac{P_{ij}}{P_i P_j} \right) \tag{3}$$

where $P_i = p_i \cdot w$ is the probability of bin i in the marginal space and $P_{ij} = p_{ij} \cdot w^2$ is the probability of bin ij in the joint space. We use Scott's rule [2] to approximate the value of the bin width.

2.2 Ensemble of Histograms

Histograms, especially histograms with a constant bin width, are highly dependent on the choice of the width of the bins. Histograms with different bin widths applied to the same dataset can provide very different results of Mutual Information due to estimation errors.

One possibility to handle this problem is using an ensemble of many histograms, all using different bin widths. We use Scott's rule and the parameter λ to determine the size of the ensemble n and the width of the bins or the number

of bins, respectively. Let k_{Scott} be the number of bins calculated by Scott's rule. All integer values in the interval $[\lceil k_{Scott}/\lambda \rceil , \lfloor k_{Scott} \cdot \lambda \rfloor]$ provide the number of bins for one instance.

After creating the histograms and estimating the values for the mutual information for each different bin width $I_i(X;Y)$ as shown in the previous section (Eqn. 3), the final Mutual Information can be calculated by using the arithmetic average over the estimated mutual information values.

2.3 Adaptive Partitioning of the XY Plane

Instead of using a constant bin width, it is possible to define variable sized bins based on the data. One of these methods is Cellucci's adaptive partitioning of the XY plane [3]. The plane is partitioned by dividing each axis into multiple equiprobable segments. Additionally it should satisfy the Cochran criterion on the expectancies $E(n_{ij})$ of the bins, which requires $E(n_{ij}) \geq 1$ for all elements of the partition and $E(n_{ij}) \geq 5$ for at least 80% of the bins.

To obtain this, each axis is partitioned that $P_x(i) = 1/k$ and $P_y(j) = 1/k$, where $P_x(i)$ is the probability of segment i of the x-axis and k denotes the number of bins in the marginal space and should be equal for each axis.

The bins in the marginal space are chosen such that each one has an occupancy of N/k points. Eqn. 3 is used to compute the values.

2.4 Kernel Density Estimation (KDE)

With kernel density estimation, the probability density function of X can be estimated by the superposition of a set of kernel functions $k(u)$, centered on the data points:

$$p(x) = \frac{1}{Nh^d} \sum k\left(\frac{x - x_n}{h}\right) \tag{4}$$

In general, the kernel function satisfies $k(u) \geq 0$ and $\int k(u)du = 1$. Using Gaussian kernel funtions, the probability density functions are given as

$$p(x,y) = \frac{1}{N} \sum \frac{1}{2\pi h^2} exp\left(-\frac{(x - x_n)^2 + (y - y_n)^2}{2h^2}\right). \tag{5}$$

2.5 Least-Squares Mutual Information (LSMI)

The least-squares Mutual Information [5] uses a concept named density ratio estimation. Instead of approximating the probability density functions $p(x)$, $p(y)$ and $p(x,y)$ separately, the density ratio function

$$\omega(x,y) = \frac{p(x,y)}{p(x)\,p(y)} \tag{6}$$

is estimated here in a single shot. The advantage of doing this is to avoid the division by estimated densities, which tend to magnify the estimation error.

Therefore, the approximated density ratio function $\hat{\omega}_\alpha(x,y)$ is modeled by a linear model $\hat{\omega}_\alpha(x,y) := \alpha^T\varphi(x,y)$ where $\alpha = (\alpha_1,...,\alpha_b)^T$ is a vector of parameters to be learned from samples, $\varphi(x,y) = (\varphi_1(x,y),...,\varphi_b(x,y))^T$ denotes a vector of basis functions, such that $\varphi(x,y) \geq 0$ for all $(x,y) \in D_x \times D_y$. To determine α the squared error J_0 is minimized

$$J_0(\alpha) = \frac{1}{2}\int_x\int_y (\hat{\omega}_\alpha(x,y) - \omega(x,y))^2 p(x)\,p(y)\,dx\,dy. \tag{7}$$

2.6 K-Nearest Neighbor Approach(K-NN)

The K-NN approach uses a fixed number k of nearest neighbors to estimate the MI. For each point in the dataset, the minimum volume V that encompasses K points is determined. By counting the number of points inside this volume in the marginal spaces the Mutual Information can be estimated.

The Mutual Information is estimated as

$$I(X;Y) = \psi(k) - \frac{1}{k} - \frac{1}{N}\sum_{i=1}^{N}[\psi(n_x(i)) + \psi(n_y(i))] + \psi(N) \tag{8}$$

where $\psi(x)$ is the digamma function and n_x denotes the neighbours in one dimension.

It can be expanded easily to m variables approximating the Joint Mutual Information (JMI):

$$I(X_1;...;X_m) = \psi(k) - \frac{m-1}{k} - \frac{1}{N}\sum_{i=1}^{N}[\psi(n_{x_1}(i)) + ... + \psi(n_{x_m}(i))] + (m-1)\,\psi(N) \tag{9}$$

3 Experiments

Our first batch of experiments resembles those presented in [1]. All approaches had to approximate the MI between two variables where the real Mutual Information was known due to the design of experiments. This includes linear, quadratic and trigonometric dependencies with different levels of noise and a changing number of available samples. For details, refer to [1]. The results are in line with those presented by Khan. The most precise and most consistent results were achieved by the K-NN, which proved to be the standard everyone has to compare to, and the KDE approach. The adaptive histogram approach turned out to be very inconsistent in case of sparse data, while LMSI showed a tendency for strong deviations of the MI for different data sets. The ensemble of histograms evinced small benefits compared to the basic histogram in high noise scenarios.

For the second batch of experiments, we focused on the feature selection tasks. For feature extraction the exact value of the Mutual Information is secondary,

Table 1. Results on the UCI data sets. Given is the balanced error rate, bold entries mark the best results.

Method	Ionosphere	German Credit	Breast Cancer	Parkinsons	Hearts
Histogram	0.0994	0.3791	**0.0463**	0.1601	0.3679
Ensemble	0.1193	0.3791	**0.0463**	0.1601	0.3752
Adapt. Hist	0.1009	**0.3596**	0.0639	0.0921	0.4554
KDE	0.1193	0.3693	**0.0463**	0.1576	0.3752
LSMI	**0.0817**	0.3693	0.0548	0.1356	0.3621
KNN	0.1126	0.3956	0.0632	**0.0647**	0.4068
KNN JMI	0.1432	0.3866	0.0775	0.1632	**0.3512**

more important is the correct ranking of the features, where systematic estimation errors will cancel out each other.

For the actual feature selection two different algorithms were used. The first is MIFS - Mutual Information for Feature Selection [7], which is a simple approximation of the JMI. At each step of the algorithm, the feature is selected, which possesses the highest MIFS value:

$$MIFS = I(X_i; Y) - \beta \sum_{s \in S} I(X_i; s) \tag{10}$$

where S denotes the set of already selected features, X_i is the feature for which the MIFS value is calculated and Y are the labels. Furthermore, β is a free parameter stating the influence of the already selected features on the remaining candidate features. β was heuristically determined to keep the MIFS value positive for the first eight features. This method was combined with all approaches to consider multi-dimensional influences presented in Sec. 2.

The second algorithm uses a forward selection strategy based on the Joint Mutual Information (JMI) [8]. In each step the feature that possesses the maximum JMI between the candidate feature, the already selected features and the labels is chosen. The computation of the JMI was done using the K-NN approach.

The feature selection test were performed on five different datasets from the UCI Machine Learning repository [9]. We used the algorithms to extract the eight best features from the data sets and tested them by using a nearest neighbor classifier and the leave one out strategy to compute the balanced error rate. The resulting error rates are shown in Tab. 1. Equal error rates for different methods are the result of selecting the same features (not necessarily in the same order).

On one hand, the table shows that each method achieves for one data set the best results. On the other hand, every method is inferior to others for some data sets. The most consistent results based on the ranking were achieved by LMSI, the KDE and the histogram approach, while the worst outputs are resulting from the K-NN approach directly estimating the JMI. This particular way of handling the JMI is outperformed by the MIFS approximation on a regular basis. In terms of computational costs, the histogram and KDE are cheapest, while the LSMI is the most expensive method due to the inherent cross validation.

4 Conclusions

In this paper, we investigated methods for estimating Mutual Information from data. For application scenarios, in which the exact value is required, our results are very similar to those published in [1]. The conclusion is to use either the Kernel Density Estimation or the Kraskov's Nearest Neighbor method.

Concerning the feature selection task only a correct ranking of the input variables is required. Most consistent performers are the Least Squares Mutual Information, the Kernel Density Estimation and simple Histogram estimation. The most problematic approach is the direct estimation of the Joint Mutual Information using the Kraskov Nearest Neighbor Method, in almost every case it was outperformed by the MIFS approximation.

The basic conclusion to be drawn from these investigations is that there is no best method to estimate Mutual Information in the feature selection context, but all considered method are more or less useful depending on the data. Nevertheless, we suggest using the KDE method, because of its good results in both types of experiments.

References

1. Khan, S., Bandyopadhyay, S., Ganguly, A.R., Saigal, S., Erickson, D.J., Protopopescu, V., Ostrouchov, G.: Relative performance of mutual information estimation methods for quantifying the dependence among short and noisy data. Physical Review E 76, 026209 (2007)
2. Scott, D.W.: Multivariate density estimation: theory, practice and visualization. John Wiley & Sons, New York (1992)
3. Cellucci, C.J., Albano, A.M., Rapp, P.E.: Statistical validation of mutual information calculations: Comparison of alternative numerical algorithms. Physical Review E 71(6), 066208 (2005)
4. Silverman, B.W.: Density Estimation for Statistics and Data Analysis. Chapman and Hall, London (1986)
5. Suzuki, T., Sugiyama, M., Sese, J., Kanamori, T.: A least-squares approach to mutual information estimation with application in variable selection. In: Proceedings of the 3rd Workshop on New Challenges for Feature Selection in Data mining and Knowledge Discovery (FSDM 2008), Antwerp, Belgium (2008)
6. Kraskov, A., Stögbauer, H., Grassberger, P.: Estimating mutual information. Physical Review E 69, 066138 (2004)
7. Battiti, R.: Using mutual information for selecting features in supervised neural net learning. IEEE Transactions on Neural Networks 5(4), 537–550 (1994)
8. Kwak, N., Choi, C.H.: Input feature selection by mutual information based on parzen window. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(12), 1667–1671 (2002)
9. Asuncion, A., Newman, D.: UCI machine learning repository (2007), `http://archive.ics.uci.edu/ml/`