Information Theory and Machine Learning

Xiaokai Wei
University of Illinois at Chicago
E-mail: xwei2@uic.edu

Abstract

Machine learning techniques are becoming increasingly useful primarily with the rapid development of Internet. A variety of machine learning methods have drawn inspirations or borrowed ideas from information theory. In this paper, we present a survey of such interactions between machine learning and information theory. Four important areas of machine learning are examined from the perspective of information theory: clustering, semi-supervised learning, feature selection and metric learning. Finally, we conclude this paper by summarizing the way how information theory interacts with machine learning and pointing out some open questions.

I. INTRODUCTION

Machine Learning receives tremendous attention in recent years from both statistics and computer science communities. It derives information automatically from data and such information can be used to predict on future data. Machine learning gives birth to a wide variety of exciting technologies, such as search engines, various recommendation systems and the IBM Watson, and hence has a profound impact on our daily life. As we all know, information theory has many interesting interactions with statistical theory. So, it is not surprising that information theory also plays an important role in many machine learning algorithms. For example, entropy is employed in popular classification models including decision tree and max-entropy regression model. This paper will present a survey of applications of information theory in machine learning and discuss some open questions.

The rest of paper is organized as follows. We first introduce some basic background knowledge about machine learning. Second, we examine the clustering problem, a popular area for applying information theory. Third, we show how information theory can be employed in semi-supervised learning. Fourth, we will see how mutual information can be utilized to select relevant features in machine learning tasks. Fifth, we turn to the metric learning problem and see how information theory can help. Finally, we conclude this paper by summarizing how information theory interacts with machine learning and pointing out some open questions.

II. MACHINE LEARNING PRELIMINARIES

Research on machine learning can be broadly divided into three categories according to how much supervision information we have: supervised learning (e.g., regression and classification), unsupervised learning (e.g., clustering), semi-supervised learning. In the classical setting of supervised learning, let \( X = (x_1, \cdots, x_n) \) denote a collection of feature vectors for \( n \) training instances in the training data. Each data instance \( x_i \in \mathbb{R}^m \) is a vector of \( m \) features, and therefore \( X \) is a matrix. Examples of features include

1) In document classification task, each document is usually represented as "bag of words", i.e., a vector each dimension of which is a word in vocabulary.

2) Banks may represent a person as feature vector (Age, HasJob, OwnHouse, CreditRating, \cdots) and use a classification model to determine whether to approve a loan application from him/her.

Each data instance in the training set has a class label \( y_i \in \{1, 2, \cdots, c\} \). The data instances in the testing set do not have labels. The goal is to "train" a model based on the training data and use this model to predict the class labels for those unlabeled data instances. In most situations, a machine learning problem has a objective to optimize. A common example is the empirical loss/error on the training data.

\[
\sum_{i=1}^{n} \mathcal{L}(\hat{y}_i, y_i)
\]

where \( \hat{y}_i \) is the label predicted by the learned model and \( \mathcal{L} \) can actually be any loss functions such as the squared loss.
III. INFORMATION THEORETIC CLUSTERING

Clustering is a kind of unsupervised learning problem and a very active area of research in both machine learning and data mining communities. The goal of (hard) clustering is to assign data instances into one of the clusters and the instances in the same cluster should exhibit similar properties. Clustering is essential to many real-world tasks. For instance, given a collection of text documents, one may want to organize them according to their content similarities (i.e., to assign them into different clusters). An example is shown in Figure 1.

Fig. 1. Example of Clustering. It is obvious that these data points can be divided into three clusters.

A. Mutual Information Criterion

Most of information theoretic clustering methods are based on mutual information. The mutual information is computed between cluster membership $Y$ and feature representations of data points in terms of distributions $X$. So the objective we are trying to maximize is $I(X : Y)$. We know

$$I(X; Y) = H(X) - H(X|Y)$$

$H(X)$ does not depend on the specific clustering, so we can use the conditional entropy $H(X|Y)$ as a measure of the clustering quality. The difficulty of such an information theoretic clustering usually lies in estimating the mutual information $I(X; Y)$. This requires the estimation of the joint probability density function of $(X, Y)$. This estimation need to be carried out on the given dataset, often by some non-parametric method [6], such as histogram- and kernel-based pdf estimations. In [16], they propose a new optimization procedure for Information Theoretic Clustering (ITC). The problem is formulated as an instance of max-k-cut on weights graphs. They then apply semi-definite programming (SDP) relaxation to find approximate solutions. The relaxed problem is convex and thus can be solved efficiently.

B. Information Bottleneck Method

The information bottleneck (IB) framework, which was introduced in [14], has various applications. In this framework, they first define relevant information in a signal $x \in X$ as being the information that this signal provides about another signal $y \in Y$. For instance, the information that face images provide about the names of the people portrayed is considered relevant. IB tries to find a short code $\hat{X}$ for $X$ that preserves the maximum information about $Y$. In other words, IB attempts to minimize the quantity $I(X, \hat{X})$ to gain compression while maximizing the mutual information $I(\hat{X}, Y)$ as in Figure [?]. $\hat{X}$ is just the so-called information bottleneck. The overall quantity considered in IB is

$$\min_{p(\hat{X}|X)} I(X; \hat{X}) - \beta I(\hat{X}; Y)$$
Fig. 2. A source coding view of information bottleneck

where $\beta$ reflects the tradeoff between compression and preservation of mutual information. This measure is similar to rate distortion theory in spirit.

Among its various applications, we introduce its application in clustering. The algorithm can be used to generate a soft (i.e., each data point has mixed membership over different clusters) clustering of the data using a deterministic annealing procedure. When used for clustering documents, it first generates a partitioning of the words $p(\hat{w}|w)$, which is supposed to preserve information about the documents. Second, the original document representation is replaced by a representation based on the word-clusters. Then, a partitioning of the documents $p(\hat{d}|d)$ is found that preserves information about the words.

C. Information Theoretic Co-clustering

More often than not, we want to cluster features as well as cluster data instances. For example, documents may be clustered based upon their word distributions and words may be clustered based upon their distribution over documents. Intuitively, clustering documents and clustering words can reinforce each other. It has been shown that clustering data instance and its features simultaneously can improve the clustering performance. In addition, co-clustering can also be used to do dimension reduction for the documents.

Let $X$ and $Y$ be discrete random variables that take values in the sets $\{x_1, \cdots, x_m\}$ and $\{y_1, \cdots, y_n\}$ respectively. Let $p(X,Y)$ denote the joint probability distribution between $X$ and $Y$. We will think of $p(X,Y)$ as a $m \times n$ matrix. We are interested in simultaneously clustering or quantizing $X$ into (at most) $k$ disjoint or hard clusters, and $Y$ into (at most) $l$ disjoint or hard clusters. Let the $k$ clusters of $X$ be written as: $\hat{X} = \{\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_k\}$, and let the clusters of $Y$ be written as: $\hat{Y} = \{\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_l\}$. In other words, we are trying to find maps $C_X$ and $C_Y$,

$$C_X: X \rightarrow \hat{X}$$
$$C_Y: Y \rightarrow \hat{Y}$$

The criterion we use to judge the quality of a co-clustering by the resulting loss in mutual information. An example is shown in Figure 3.

$$I(X;Y) - I(\hat{X};\hat{Y})$$

$$p(X,Y) = \begin{bmatrix}
.05 & .05 & .05 & 0 & 0 & 0 \\
.05 & .05 & .05 & 0 & 0 & 0 \\
0 & 0 & 0 & .05 & .05 & .05 \\
0 & 0 & 0 & .05 & .05 & .05 \\
.04 & .04 & 0 & .04 & .04 & .04 \\
.04 & .04 & 0 & .04 & .04 & .04
\end{bmatrix} \quad p(\hat{X},\hat{Y}) = \begin{bmatrix}
.3 & 0 \\
0 & .3 \\
.2 & .2
\end{bmatrix}$$

Fig. 3. An example of calculation of probability of $X$ and $Y$ in the information theoretic co-clustering. Apparently, we can group the rows into three clusters: $\hat{x}_1 = \{x_1, x_2\}$, $\hat{x}_2 = \{x_3, x_4\}$, $\hat{x}_3 = \{x_5, x_6\}$. Similarly, the natural column clustering is: $\hat{y}_1 = \{y_1, y_2, y_3\}$, $\hat{y}_2 = \{y_4, y_5, y_6\}$. It can be verified that the mutual information lost due to this co-clustering is only 0.0957, and that any other co-clustering results in a larger loss of mutual information.

The lower the loss, the better the clustering result. It can be proved that a local minima of this criterion can be achieved in an iterative manner.
IV. INFORMATION THEORETIC SEMI-SUPERVISED LEARNING

In the classical setting of supervised learning, all the training data have class labels (the so-called supervision information) associated with them. But labeled data can be hard to get in practice since

1) Labels may require human experts
2) Labels may require special devices (e.g., to determine the category/class of chemical compounds)

In comparison, unlabeled data is often cheap in large quantity. Semi-supervised learning studies how to make use of unlabeled data. We can see necessity of utilizing unlabeled data by looking at the following in Figure 4. In the setting of semi-supervised learning, we assume there are \(l\) labeled data instances and \(u\) unlabeled data instances. Most semi-supervised learning algorithms are based on the “cluster assumption”: data points in the coherent cluster are likely to have the same class label.

![Figure 4. Illustration of semi-supervised learning. ‘+’ is labeled as positive and ’o’ is unlabeled. If we only consider labeled data, it is difficult to determine the decision boundary. If we take into consideration the unlabeled data, we can see Line 3 is likely to be a good decision boundary.](image)

A. Entropy-based Approaches

Approaches in the semi-supervised learning field usually add entropy over the unlabeled data in the objective of optimization. In the entropy-based method in [20], the author assumes that if the two classes are well-separated, then \(p(y|x)\) on any unlabeled instance should be close to 0 or 1, that is, entropy \(H(p)\) over unlabeled data instances should be small. So entropy regularizer

\[
\Omega = \sum_{j=l+1}^{l+u} H(p(y = 1|x_j, w))
\]

where \(w\) is the parameter we want to learn. This regularization term is combined with the empirical loss as the objective.

In [12], the idea is similar in that it also assume data points in a high-density region is likely to belong to the same class. A mutual information based regularization term is introduced to discourage conditionals \(p(y|x)\) in regions with high \(p(x)\).

B. Information Rate-based Approach

[3] extends [12] and formulates semi-supervised learning as a communication problem. Regularization is interpreted as the rate of information, which again discourages complex conditionals \(p(y|x)\) in regions with high \(p(x)\). The problem becomes finding the unique \(p(y|x)\) that minimizes a regularized loss on labeled data. To solve this minimization problem, they give a local propagation algorithm.
V. INFORMATION THEORETIC FEATURE SELECTION

It is obvious that not all features have the same discriminative power: when we classify articles into different categories, words like *the*, *is* usually do not help but words like *football*, *soccer* often occur in the *Sports* class and hence can be useful when determining the class label. Moreover, there is a widely observed phenomenon in machine learning called "curse of dimensionality": high dimension can not only increase the computational cost but introduce some other serious problems like sparsity, which makes learning a reasonable model more difficult. So we want to select a subset of relevant features while still achieving comparable or even better performance.

A common approach based on information theory is to use the Mutual Information between the feature and class label. Intuitively, if a feature and the class label has high mutual information, the feature is very likely to be a good indicator of the class. If the mutual information is low, then the feature is probably irrelevant and therefore can be discarded. So we can simply rank the features with respect to the mutual information. Then we retain the top-k features and discard the others. But such an approach ignores the dependency between features. Some features, although having high mutual information with class label, may be redundant, i.e., behave similarly with the other features. So we can just retain one of those features which behave almost identically rather than keep all of them. Many papers take into consideration redundancy besides relevance.

A. Mutual Information-based Feature Selection (MIFS)[2]

The ranking score for the $n$-th feature $X_n$ is the following,

$$ J_{mifs} = I(X_n; Y) - \beta \sum_{k=1}^{n-1} I(X_n, X_k) $$

where $\beta$ is parameter that controls the relative importance relevance and redundancy and $X_1, \ldots, X_{n-1}$ are the features already selected. On the right-hand side of the above formula, the first term represents the relevance of feature $X_n$, while the second term captures how similar $X_n$ is with the previously selected features. Clearly, by defining the ranking criterion this way, we make a tradeoff between relevance and redundancy.

B. Maximum-Relevance Minimum-Redundancy (MRMR)[10]

The ranking score for the $n$-th feature $X_n$ is defined as,

$$ J_{mifs} = I(X_n; Y) - \frac{1}{n-1} \sum_{k=1}^{n-1} I(X_n, X_k) $$

This is essentially the same with MIFS by noting that $\beta = \frac{1}{n-1}$ here. This is just to take the mean of redundancy term.

C. Joint Mutual Information (JMI)[18]

This paper proposes a criterion based on joint mutual entropy,

$$ J_{jmi} = \sum_{k=1}^{n-1} I(X_n X_k; Y) $$

$$ = I(X_n; Y) - \frac{1}{n-1} \sum_{k=1}^{n-1} [I(X_n; X_k) - I(X_n; X_k|Y)] $$

This is the information between the targets and a joint random variable, defined by pairing the candidate $X_n$ with each of selected features. As we can see, $J_{jmi}$ is just $J_{mrmr}$ plus $\frac{1}{n-1} \sum_{k=1}^{n-1} I(X_n; X_k|Y)$.

D. Information Fragments [15]

It is a criterion used in Computer Vision, which is defined as the following.

$$ J_{if} = \min_k I(X_n X_k; Y) - I(X_k; Y) $$

This criterion measures the gain of combining a new feature $X_n$ with each existing feature $X_k$, over simply using $X_k$ by itself.
E. Conditional Mutual Information Maximization (CMIM) [7]

This is a very popular criterion, which is defined as

\[ J_{cmim} = \min_k I(X_n; Y|X_k) \]

\[ = I(X_n; Y) - \max_k [I(X_n; X_k) - I(X_n; X_k|Y)] \]  

CMIM uses the information between a feature and the class label, conditioned on each of previously selected features. Taking the maximum means that \(X_n\) would be considered redundant as long as \(X_n\) share too much information with one of selected features. This measure is shown to empirically better than the previously proposed methods.

F. Other Information Theoretic Measure

[19] develops a feature selection measure based on Renyi’s quadratic and cross entropy. Renyi’s quadratic entropy is

\[ H_2(X) = -\log(\int p^2(x) \, dx) \]

Also, they use a non-parametric estimator \(\hat{V}(X)\) of Renyi’s quadratic entropy. Rather than selecting features individually, they utilize a projection matrix \(W\) and transform the original feature matrix \(X\) into \(Y = WX\). The criterion for determining \(W\) is

\[ E(W) = (1 - \lambda)\hat{V}(WX) + \lambda\hat{V}(WC; C) \]

where \(\hat{V}(WC; C)\) is the cross entropy between \(WC\) and class label \(C\). An advantage of this measure is that it is robust to outliers. A half quadratic optimization method is used to solve the problem in an iterative manner.

VI. INFORMATION THEORETIC METRIC LEARNING

There are many machine learning techniques which involve distance/similarity between data instances. For example, K-means needs to calculate the distance to centroids to re-assign each point to clusters. k-nearest neighbors (kNN), which is a simple yet effective classification technique, determines the class label of a data instance by a majority voting of its \(k\) nearest neighbors. In such circumstances, measures like Cosine Similarity or Gaussian Kernel Function can be used to compute the necessary similarity. But such an arbitrarily selected similarity measure may not be optimal and do not fit the given data very well. The idea behind metric learning is to learn a distance metric guided by the data. For an illustrative example, one may see the following Figure 5.

![Fig. 5. Motivation of metric learning. By learning a distance metric, we project the data points to another space where data points in the same class are close to each. Such a learned metric can obviously facilitate classification based on kNN](image-url)
A. Minimizing an Information theoretic distance measure

Many papers concerning metric learning use Mahalanobis distance, which is parameterized by a matrix $A$. The Mahalanobis distance between $x_i$ and $x_j$ is as follows.

$$d_A(x_i, x_j) = (x_i - x_j)^T A (x_i - x_j)$$

Note that when $A$ is an identity matrix, this is just an Euclidean distance. Based on Mahalanobis distance, one just needs to learn matrix $A$ which fits the data well. [4] expresses the learning a Mahalanobis distance function as a Bregman optimization problem, by minimizing the differential relative entropy (the LogDet divergence) between two multivariate Gaussians (characterized by covariance matrices) subject to linear constraints on the distance function. One Gaussian is constructed from some prior knowledge and they are trying to make learned Gaussian close to the former Gaussian. The constraints insures that data instances known to be similar should have small distance between them and data instances known to be dissimilar should be far away from each other.

$$\min_A \text{KL}(p(x; A_0)||p(x; A))$$

s.t. 

$$d_A(x_i, x_j) \leq u, (i, j) \in S \text{ (for similar data instances)}$$

$$d_A(x_i, x_j) \geq l, (i, j) \in D \text{ (for dissimilar data instances)}$$

In paper [17], an ideal kernel(similarity) matrix $P$ is first constructed based on the assigned class labels. Then another matrix $Q$ based on distance metric is constructed to approximate first matrix. So, a good distance metric is one that can makes the latter matrix $Q$ and former matrix $P$ as similar as possible. So how to quantify the difference between two matrices. Inspired by the idea of information geometry, They relate these two kernel matrices to two Gaussian distributions $Pr(x|P)$ and $Pr(x|Q)$, and the difference between the two kernel matrices is then computed by the Kullback-Leibler (KL) divergence between the two Gaussian distributions.

$$d(P)||Q) = KL(Pr(x|P)||Pr(x|Q))$$

And it can be shown that $d(P)||Q)$ can be further represented in the form of Bregman Divergence. Hence, the optimal distance metric is then found by minimizing the divergence between the two distributions. Unlike many existing metric learning methods where an optimization problem need to be solved (can be computationally expensive), this proposed approach has a closed-form solution and are computationally more efficient. [17] is similar to [4] in that they both use Bregman Divergence for matrix.

B. Coding Length-based Approach

In [1], the similarity is defined from the perspective of coding theory. The basic idea is that two objects should be considered more similar the more we can ”compress” one given the information in the other. Intuitively, two items are similar if they share common aspects. Learning a similarity metric is learning what aspects tend to be shared more than others. Hence, they define the similarity as the gain in coding length obtained by encoding $x$ when $x'$ is known. Put another way, this is the gain in coding length when shifting from independent encoding of the pair to joint encoding.

$$\text{CodeSim}(x, x') = cl(x) - cl(x|x', H_1)$$

$$= \log p(x|x', H_1) - \log p(x)$$

where $H_1$ is the hypothesis stating that the two data instances share the same label. This measure can also be seen as a log-likelihood ratio statistic,

$$\log p(x|x', H_1) - \log p(x) = \frac{p(x, x'|H_1)}{p(x, x'|H_0)}$$

where $H_0$ is the hypothesis stating independence between the points. Hence, the coding similarity is the optimal statistic for determining whether two data instances are drawn from the same class or independently. This similarity measure can be learned efficiently under some simplifying Gaussian assumptions and outperforms previously proposed approaches.

VII. MISCELLANEOUS

A. Application in information Retrieval

One classical task in information retrieval is to retrieve relevant document given a query issued by the user. A simple yet popular technique is to first represent the query and documents as distribution over words and then calculate the KL-divergence between these two distributions. Then we can simply return the top documents with low $KL(Pr(q)||Pr(d))$. 

B. Factor Graph

Probabilistic inference is an important problem in statistical machine learning. [9] introduces factor graph and the sum-product algorithm, a generic message-passing algorithm. It has a variety of applications in the artificial intelligence, signal processing, and digital communications communities, the techniques in which may be derived as specific instances of the sum-product algorithm. Genealogically, factor graphs are a straightforward generalization of the Tanner graphs. Tanner [13] introduced bipartite graphs to describe families of codes which are generalizations of the low-density parity-check (LDPC) codes.

Factor graphs can be used for decoding. For instance, Turbo code and LDPC code can be represented in the form of factor graph. Iterative decoding of Turbo codes and LDPC codes can be accomplished via a message-passing approach introduced in [9].

When it comes to probabilistic inference, many real-world problems (e.g., image denoising) can be represented as graphical probabilistic models such as Bayesian Network (BN) or Markov Random Field (MRF). Both BN and MRF can be transformed into factor graphs (Figure 6 and therefore we can inference their parameters using sum-product algorithm.

![Figure 6](image)

Fig. 6. Graphical probabilistic models. (a) A Markov random field (b) A Bayesian network (c) A factor graph [9]

VIII. Conclusion

We have introduced how information theory is applied in several machine learning techniques. As we can observe, there are basically two ways of applying information theory in the machine learning community.

1) Use basic information theoretic concepts (e.g., entropy, mutual information, KL-divergence) as objectives or regularization terms in the optimization problem. Most of works in this survey paper fall into this category.

2) Use more sophisticated information theory, such as rate-distortion theory and coding theory to motivate new techniques or provide additional insights for existing machine learning techniques.

There are certain other areas of machine learning which have information theory-based approaches. Due to time and space constraint, we do not include these areas (e.g., active learning, reinforcement learning, compressed sensing) in this paper.

As open questions, some other components of information theory (e.g., channel capacity, network information theory) might be explored in the context of machine learning. Besides, some relatively young machine learning algorithms, such as structural learning [8] and collective classification [11], may also be related to more sophisticated information theory and can be seen from an information theoretic perspective. Discovering such unveiled relationships can also be a promising direction.

References