Scalable Mutual Information Estimation using Dependence Graphs

Morteza Noshad, Yu Zeng, Alfred Hero

Electrical Engineering and Computer Science
University of Michigan
Motivation: Measure of Dependence

Gene Selection Problem (Liu et al 2010)

Tree learning based on pairwise dependencies (Chow & Liu 1968)

Observations (mixed signal) → True Sources

ICA

Independent Component Analysis (Isomura & Toyoizumi 2016)
Outline

1. Mutual Information
2. Ensemble Dependence Graph Estimator
3. Application in Deep Learning
4. Conclusions and Future Work
Measure of Dependence

- Mutual information (MI) is a measure of dependence between two random variables.
- MI is widely used in information theory, statistics and machine learning.

**Mutual Information**

The general mutual information function between $X_1$ and $X_2$ is

$$I_g (X_1; X_2) := \int g \left( \frac{f_1(x_1)f_2(x_2)}{f_{12}(x_1, x_2)} \right) f_{12}(x_1, x_2)dx_1dx_2,$$

where $g$ is smooth convex function with $g(1) = 0$.

- For Shannon mutual information, $g(x) = x \log x$. 

Problem Definition: Estimation

- **Goal:** Accurate and computationally fast estimation of divergence

- **Assumption:**
  - Densities are (Hölder) smooth and bounded from below and above

- **Convergence analysis:** find rate of decrease of MSE in #samples

\[
MSE = Bias^2 + Variance = cN^{-\beta/(2\beta+d)}
\]
Problem Definition: Estimation

- **Goal:** Accurate and computationally fast estimation of divergence

- **Assumption:**
  - Densities are (Hölder) smooth and bounded from below and above

- **Convergence analysis:** find rate of decrease of MSE in #samples

\[
MSE = Bias^2 + Variance = cN^{-\beta/(2\beta+d)}
\]

⇒ Optimal *parametric* MSE rate: \( \beta \to \infty \)

\[
RMSE = \sqrt{MSE} = cN^{-1/2}
\]
This work achieves optimal rates using ensemble estimators.
Previous Work on Estimation of Information Measures

- $N$: Number of samples; $k$: Parameter of kNN graph; $d$: Dimension.
- The densities are assumed to be $d$ times differentiable.

| [1] | Poczos and Schneider (2011) |
Proposed Approach

Input Data → Locality Sensitive Hashing → Dependence Graph → Ensemble Estimation → MI Estimate
Locality Sensitive Hashing

- $N$ i.i.d pairs $(X_i, Y_i)$ are drawn from $P_{XY}$.
- $X = \{X_1, ..., X_N\}$ and $Y = \{Y_1, ..., Y_M\}$.
- Hash map of $X$ and $Y$: $H : \mathbb{R}^d \rightarrow \{1, ..., F\}$.
- $F$ is the number of buckets and is a linear function of $N$.
- $H(x)$ specifies a vertex index of a so called Dependence Graph.
Locality Sensitive Hashing

- Hash map of $X$ and $Y$: $H : \mathbb{R}^d \rightarrow \{1, ..., F\}$.

Locality-Sensitive Hashing (LSH) $H$

$$H(u) = [h(u_1), h(u_2), ..., h(u_d)], h(u) = \left\lfloor \frac{u + b}{\epsilon} \right\rfloor$$

- $u = [u_1, \ldots, u_d]$ represents $X$ or $Y$.
- $b$ is a fixed random number in the range $[0, \epsilon]$.
- $\epsilon$ is a bandwidth parameter of the estimator.
- $H$ maps neighboring points to common value.
Dependence Graph

- A bipartite graph with two sets of nodes $V$ and $U$.
- Map the points in $X$ and $Y$ to the nodes in $U$ and $V$ using $H$.

An example of a dependence graph
Assign the weights $\omega_i$ and $\omega'_j$ respectively to the nodes $v_i$ and $u_j$.

$\omega_{ij}$ denotes the weight of the edge $(v_i, u_j)$.

$$
\omega_i = \frac{N_i}{N}, \quad \omega' = \frac{M_j}{N}, \quad \omega_{ij} = \frac{N_{ij}/N}{(N_i/N)(M_j/N)}
$$

$N_i$ and $M_j$: respectively the number of nodes mapped to $v_i$ and $u_j$.

$N_{ij}$ is the number of node pairs $(X_k, Y_k)$ mapped to $(v_i, u_j)$.

$N_{ij} \leq N_i, N_j$. We only consider the edges with $N_{ij} > 0$.

$\omega_i, \omega_j$ and $\omega_{ij}$ respectively are estimates for $f_i, f_j$ and $f_{ij}/f_if_j$.

**Dependence Graph Estimator of MI**

The base dependence graph estimator is defined as follows

$$
\hat{I}(X, Y) := \sum_{e_{ij} \in E_G} \omega_i\omega'_j g(\omega_{ij})
$$
Assume that $f_1$ and $f_2$ are density functions with continuous and bounded derivatives of up to the order $d$.

**Theorem**

*The bias of the estimator can be upper bounded as*

$$
E\left[ \hat{l}_g(X, Y) \right] = \int g \left( \frac{f_1(x)}{f_2(x)} \right) f_2(x) dx + \sum_{i=1}^{d} C_i'' \epsilon_i^i + O\left( \frac{1}{N \epsilon^d} \right).
$$

- Variance is also proved to be upper bounded by $O(1/N)$. 
Let $\mathcal{L} := \{l_1, l_2, \ldots, l_L\}$ be a set of index values.

Consider an ensemble of estimators $\{E_l\}_{l \in \mathcal{L}}$, and the weights $w$ with $\sum_{l \in \mathcal{L}} w(l) = 1$.

$w_0(l)$ are the solutions of a specific offline optimization problem.

The ensemble estimator $E_{w_0} := \sum_{l \in \mathcal{L}} w_0(l)E_l$ achieves the optimal parametric rate $O(1/N)$. 
Comparison of EDGE, Ensemble DKE and KSG Shannon MI estimators. $X \in \{1, 2, 3, 4\}$, and each $X = x$ is associated with multivariate Gaussian random vector $Y$, with $d = 4$. 

Numerical Results
Schwartz-Ziv and Tishby (2017) proposed to use mutual information to analyze deep neural nets.

- $I(Y; T)$: The information of the hidden layer $T$ with respect to $Y$.
- $I(X; T)$: The compression of $X$.

**Figure:** A DNN with dimension 12-10-7-5-4-3-2 with tanh activations
Compression happens in any network.

- Saxe et al (ICLR 2018) refuted this claim by showing that there is no compression with ReLU activation.
- The estimation method used by both of the papers was inaccurate (histogram).

Learning consists of two distinct phases; fitting and compression.

Compression occurs due to the diffusion-like behavior of SGD.

We need a stronger estimator in order to get accurate results for higher dimensions.
Information Plane Using EDGE

- MNIST handwriting dataset classification.
- Compression is observed for both tanh and ReLU activations.
- The estimated intrinsic dimension is 14 (Costa & Hero 2006).
- We choose $L = 20$ as the number of basic estimators for the ensemble estimator.

**Figure:** Information plane estimated using EDGE
Our Results

- Compression happens in any network.
  - We observe compression in DNNs with ReLU and tanh activations as well as CNNs.
- Compression could start from the beginning of the training process.
- We observe compression with other optimization methods such as BGD and Adam.
Propose EDGE, an optimal estimator of mutual information based on locality sensitive hashing (LSH) and dependence graph.

Prove that the MSE convergence rate is $O(1/N)$.

Apply EDGE on estimation of Information Plane (IP) in deep learning.

Future Work:

- Explore the impact of choosing different hash functions in practice.
- Derive non-asymptotic convergence rate.
Questions?

Thank you