MAXIMIZATION OF MUTUAL INFORMATION FOR INFORMATION CLUSTERING

Tilman Hinnerichs
tilman@hinnerichs.com

This summary is a summary of the talk given at the February 21st 2020 at MPI MiS by Tilman Hinnerichs.

1 Clustering

Clustering algorithms are a group of algorithms for the exploration of similarity structures among datasets. It aims on identifying groups within the data, that are somehow similar to each other. As no additional data is used, these are classified as unsupervised learning methods. Crucially, these so called clusters do not have any label and are only based on the underlying information of each data point. In general, whether a clustering is good or bad can only be determined by an expert, or ground truth. Alternatively, quality of clusters can be measured by various dataset independent measures, e.g., average radius of the clusters and distance between different clusters. The various algorithms differ in their interpretation of similarity.

2 k-means-clustering

k-means is one of the simplest and widely known clustering algorithms and shall thus be explained here briefly. k hereby denotes the number of clusters.
The steps are the following:

1. Randomly initialize k points as initial means
2. Assign each data point to the closest of the k means
3. For each of the sets of data points assigned to one of the k means, calculate the mean among each group of assignments

Repeat steps 2 and 3 until the means are not changing anymore or within a certain margin.

3 Discussed papers

When trying to find a suitable representation, we are trying to have as much information from the input within the output, i.e. loose as little information from the input as possible. Thus, the *transinformation*

or *mutual information* (MI) has to be maximized also known as the infomax principle (Linsker, 1988). Hence, making MI the objective function for representation learning and clustering seems to be a suitable approach. However, mutual information is generally quite difficult to compute, especially in continuous and high dimensional settings.

Different approaches are trying to maximize mutual information between different points of the network, e.g., between input and the output, or between different representations.

Let $\mathcal{X}, \mathcal{Y}$ be the domain and the range of a parametric function $E_\psi : \mathcal{X} \to \mathcal{Y}$ with parameters ψ. Additionally, let $\mathcal{E}_\Psi = \{E_\psi\}_{\psi \in \Psi}$ over Ψ. Then we want to find the set of parameters ψ, such that MI is maximized between two given points.

3.1 Learning Deep Representations by mututal Inforamtion Estimation and Maximization – Deep InfoMax (DIM) (Hjelm et al., 2019)

3.1.1 Approach

In DIM find the set of parameters ψ, such that the mutual information $\mathcal{I}(X; E_\psi(X))$ is maximized. Ideally the marginal distribution induced by forwarding samples from the empirical distribution for a given ψ should match a prior distribution, thus encouraging the output of the encoder to have desired characteristics, e.g., independence.
The authors maximize the mutual information between spatially-preserved features, that have to be precomputed and their high-level representation. DIM simultaneously estimates and maximizes the mutual information.

The authors present various techniques to estimate the MI, eventually deciding for the Noise-Contrastive Estimation (infoNCE).

Deep Infomax' objective function can be written as follows:

$$\arg \min_{\omega_1, \omega_2, \psi} (\alpha \hat{\mathcal{I}}_{\omega_1, \psi}(X, E_\psi(X)) + \frac{\beta}{M^2} \sum_{i=1}^{M^2} \hat{\mathcal{I}}_{\omega_2, \psi}(X^{(i)}, E_\psi(X))) + \arg \min_{\psi} \arg \max_{\phi} \gamma \hat{\mathcal{D}}_\phi(\mathbb{V} | \mathbb{U}_{\phi, \mathbb{P}}) \quad (1)$$

Hereby the following notations are used:

$\hat{\mathcal{I}}_{\omega,\phi}$ Approximation of mutual information determined by a discriminator with parameters ω

M^2 number of features with preserved locality extracted from the input images

$X^{(i)}$ Feature i of input image X

α, β, γ weights to each part of the objective function, chosen as hyperparameters. Choices in these hyperparameters affect the learned representations in meaningful ways.

$\hat{\mathcal{D}}$ discriminator for the minimization of KL-divergence between output distribution $\mathbb{U}_{\psi,\mathbb{P}}$ and prior distribution $\mathbb{V}$ with certain desired properties .

This equation can be divided into several parts that cope with

1. global, and
2. local MI maximization, and
3. prior matching

The global part approximates the mutual information between the whole input image and its representation, as depicted in Figure 1 of the paper. For training, the first discriminator tries to determine whether the given global feature matches the given input image, which can be either the real or a random one. The global feature hereby is the output of the neural network and thus its representation.
For the local information, a second discriminator tries to distinguish between real and fake local feature vectors, concerning the given global feature, hence checking whether this local feature fits the global one.
A third discriminator is used to minimize the Kullback-Leitner-divergence between some prior with some wanted property, that the output shall gain, and the output of the network.

3.1.2 Results

The authors find that incorporating knowledge about locality in the input significantly improved performance in downstream tasks. It outperforms the very limited amount of other unsupervised clustering algorithms.

Major drawbacks are:

- Only applicable to images,
- Needs preprocessing creating the images features,
- needs k-means after representation calculation for eventual clustering, and is thus prone to degeneration,
- calculates over continuous random variables, which requires complex estimators, making an approximation necessary,
- DeepInfomax estimator reduces to entropy (stated by IIC paper), and
- doesn't give a hint on best amount of classes, hence making the number of clusters a hyperparameter for k-means.

3.2 Invariant Information Clustering (Ji, Henriques, Vedaldi, 2018)

3.2.1 Approach

The IIC paper focuses on two issues that previous papers, such as DIM and IMSAT, weren't addressing.

The first issue is the general degeneracy of clustering solutions. As widely known, k-means degenerates for unfortunate initializations, thus leading to one cluster capturing the whole data. This issue is addressed through the nature of the mutual information, as it is not maximized for degenerate solutions. Second, noisy data with unknown or distractor classes such as in *STL10* bother some clustering algorithms. This is addressed through the concept of auxiliary overclustering.

Let $x, x' \in \mathcal{X}$ be a paired data sample from a joint probability distribution $P(x, x')$, e.g., two images containing the same object. We now want to maximize the mutual information between representation of those similar entities, dismissing instance specific information and preserving similarities. The goal is to make the representations of paired samples the same, which is not the same as minimizing the representation distance.

$$\max_{\psi} \mathcal{I}(E_\psi(x), E_\psi(x')) \tag{2}$$

In contrast to DIM, the representation space $\mathcal{Y} = \{1, \dots, C\}$ is discrete and finite, hence making the calculation of the MI precise and fast. The authors distinguish between semantic clustering and representation learning, as DIM is learning a representation which can be utilized for clustering

downstream, while IIC performs the clustering task end to end.

We denote the output $E_\psi(x)$, that can be interpreted as the distribution of a discrete random variable z over C classes, with $P(z = c|x) = E_{\psi,c} \in [0, 1]^C$.
Now consider a pair of such cluster assignment variables z, z' for two inputs x, x'. Their conditional joint distribution is given by $P(z = c, z' = c'|x, x') = E_{\psi,c}(x) \cdot E_{\psi,c'}(x)$.

Thus, one is able to build a $C \times C$ matrix $\mathbf{P}$, with $\mathbf{P}_{cc'} = P(z = c|z' = c')$, computed by

$$\mathbf{P} = \frac{1}{n} \sum_{i=1}^{n} E_{\psi,c}(x_i) \cdot E_{\psi,c}(x_i')^T \tag{3}$$

Why degenerate solutions are avoided: $I(z, z') = H(z) - H(z|z')$ ($H(z)$ maximized when all equally likely to happen, $H(z|z')$ is 0 when assignments are exactly predictable from each other. This encourages deterministic one-hot predictions. Whether the smoothness of the clustering comes from the effects of that formula or from the model itself has yet to be determined.

As for totally unsupervised tasks no labels are available and thus no information about pairs among the input data is present, perturbation of that input is used. The objective function becomes

$$\max_{\psi} \mathcal{I}(E_\psi(x), E_\psi(gx)) \tag{4}$$

where g denotes a random perturbation, such as rotation, skewing, scaling or flipping (and other).

Furthermore, IIC is capable of auxiliary overclustering, adding additional output heads, that are trained with the whole dataset. These capture unrelevant or distractor classes, or noise within the data.

3.2.2 Results

Benefits:

- Only reliant on the information itself,
- As range is discrete, exact computation of MI is possible and fast, and
- set state of the art performance on various datasets (e.g., IMSAT, DIM, ...)

4 Perturbation models

4.1 Wasserstein distance

Properties:

- Robust to rotation and scaling
- also called Wasserstein ground metric or Earth Movers distance
- implementations available from OpenCV and Scikit

For metric sample space $(\mathcal{X}, d_\mathcal{X})$, define Earth Movers distance of images as follows

$$d_\mathcal{X}(X, Y) := W_{q,d_\Omega}(X, Y) = \inf_{\pi} \{(\mathbb{E}_{(x,y)\sim\pi} d_\Omega(x, y)^q)^{\frac{1}{q}}\} \tag{5}$$

Usually we denote this with Wasserstein-q. In general we will only examine Wasserstein-2.

- sample images from
$$p(\xi|x) := \exp(-d_W(x, x+\xi)^2/\eta^2)d(\xi) \tag{6}$$
(See Lin, ..., Montufar, 2019, Chapter 3)

Thus for a given perturbation ξ we can determine its probability.

4.2 Sinkhorn

In order to obtain those perturbations ξ we will have a look at this paper (Wong et al., 2019), that also provides code on Github for its algorithms. We therefor build a ball with respect to the Wasserstein ground metric as previously defined. For a data point (x, y) at position x with label y we build the projection of this point onto the ball around the point x formalized as follows.

The projection of an arbitrary point w onto the ball $\mathcal{B}(x, \epsilon)$ around x with radius ϵ is described by

$$proj_{\mathcal{B}(x,\epsilon)}(w) := \arg\min_{x \in \mathcal{B}(x,\epsilon)} ||w - z||_2^2 \tag{7}$$

Starting at $x^{(0)} = x$ or any other randomly initialized point within $\mathcal{B}(x, \epsilon)$, we iteratively take steps with a step size α and some loss l (e.g., cross-entropy loss) according to this update formula

$$x^{(t+1)} := proj_{\mathcal{B}(x,\epsilon)}(x^{(t)} + \arg\max_{||v|| \leq \alpha} v^T \nabla l(x^{(t)}, y)) \tag{8}$$

As clustering is a fully unsupervised task, we do not have a label y, and the ordinary MI from the IIC paper cannot be used as the codomain is not differentiable.

4.3 Image deformation

A solution to this issue might be the ADef algorithm proposed by Rima et al. (2019). The authors build ADef iteratively for an existing, arbitrary model, aiming for an adversarial attack. This algorithm can be used to simulate the steps with a given step size, as proposed by the iterative Sinkhorn method.

As mentioned this relies on on a given, already existing model. Whether this is suitable **still needs to be discussed**.

4.4 Algorithm

The algorithm may be as mentioned in the sections above, but may be condensed as follows. For every input image $x \in X$:

Algorithm 1 Wasserstein metric based perturbation

```
procedure W2-perturbation(x, ε)
    x' := x
    while True do
        x'' := ADefStep(x', ...)          ▷ Single ADef step with step size α
        if W2(x'', x) ≤ ε then            ▷ x'' within ball
            x' := x''
            if With probability p(ξ|x) choose then   ▷ ξ dependent on W2(x', x)
                Return x'
        else
            Return x'
```