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 the 1D-case)

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 Random deformation

Deformations can be modeled by a 2D vector field $\tau : \mathbb{R}^2 \to \mathbb{R}^2$, where the resulting image is defined for each pixel $x \in \{1, \ldots, width\} \times \{1, \ldots, height\}$ with value $I(x) \in [0, 1]$ by $I(x+\tau(x))$. We compare the results with and without smoothing, under usage of Gaussian filters over the vector field.

While a manual implementation not efficiently parallelizable, the performance of elastic deformation has been heavily tested in the last few years. The procedure generates a coarse displacement grid with a random displacement for each grid point. This grid is then interpolated to compute a displacement for each pixel in the input image. The input image is then deformed using the displacement vectors and a spline interpolation. [1][2]

Elastic deformation can also be backpropagate the learned as it gradient can be backpropagated.

The performance of elastic search on MNIST has shown minor performance gains on MNIST[3]

4.3 Sinkhorn

In order to obtain 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) and project it back onto the ball by calculating Wasserstein-2 according to this update formula

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

The algorithm runs until a maximum step amount is reached or when the ball is left. As we are especially interested in the unsupervised case, we introduce a random, non-trainable model with an arbitrary amount of potential classes. This classifier then gets fed into the proposed Sinkhorn algorithm, stated in the paper.

Parameters hereby are

- maximum number of steps
- step size
- **radius of the ball**
- norm for both the distance and the radius of the ball itself

4.4 Image deformation

The ADef algorithm proposed by Rima et al. (2019) provides another approach, utilizing adversarial deformations. The authors build ADef iteratively for an existing model, aiming for an adversarial attack. We can use this algorithm over an arbitrary model as in the Sinkhorn case to generate deformations for the given images, in an unsupervised manner.

In contrast to the Sinkhorn approach, the ADef algorithm needs the correct label of the given image and will try to perturb the image in direction of the second most likely classification. As there is no ground truth label in the unsupervised case, we will just assume a random label over a random classifier. As the classification space of a random model is often quite patchy, this isn't really playing to the full strengths of this algorithm.

Additionally they do not use the Wasserstein norm, but rather the l_2 and l_{inf} respectively. I applied the W_2 from the Sinkhorn implementation for this purpose, too.

4.5 Tikhonov Regularizer

The same can be applied under usage different regularizers as done in [4]. As this work is performing on the same tasks, I omitted experiments and tried took it as a comparison.

4.6 Experiments

In this section I will present some results of the formerly mentioned approaches. The datasets were perturbed first and the put into the best performing models according to the IIC-paper. While they achieved their results using about 2000 epochs, making only marginal, but significant gains in the last 1500 epochs, I run all algorithms with about 300 epochs. This run for about 36 hours in the normal case and up to about 96 hours in the other cases on 4 V100 (128 GB of VRAM). Experiments were run with different Sinkhorn radii and varying number of added perturbed images. Amount of perturbed added images ranged from 1 to 10, while no additional gains were achieved after the first 5.

Tested datasets were MNIST and CIFAR-10.

Approach	normal IIC	random def	Sinkhorn				ADef
Radius (W_2)			0.001	0.01	0.1	1	
MNIST	97.5	97.7	97.5	97.7	97.8	97.0	97.5
CIFAR-10	57.0	57.4	57.0	57.6	57.8	58.0	57.8

Table 1: Maximum mean accuracies for each approach

While these results seem to be quite good at first sight, there are some major problems here. The behaviour for larger epoch sizes didn't show any significant difference to the original/normal results on MNIST. On CIFAR the standard deviation was was about 5 and thus too large to prove any significance. Addtionally, Even though the results seem very smooth on CIFAR-10, only the best results are displayed, thus representing completely different test cases, and could thus suffer heavily from validation overfitting. The results are therefor taken with caution.

5 Loss function

I didn't find that much room for improvement regarding the mutual information based loss function. However, with respect to [4] one could mollify the regularizer with a Wasserstein gradient of the loss

as described in the paper in section 4. This could lead to further smoothing of the surrounding of the classification space. The author did not perform any experiments regarding the loss function and this hypothesis.

6 Model alteration

I found it especially hard to find any model that is able to have the same expressivity as the original model, given in the IIC paper. Altering the depth and structure of the network lead to minor and larger performance drops.

7 Conclusion

While the results weren't that groundbreaking I do see the appilcation of those perturbations in data augmentation e.g. in medical imaging, when not that many images are given, as it provides reasonable training samples. See an exception from this below.

8 Perturbed IIC for Image segmentation

While we were focused on unsupervised image clustering, IIC also provides a successful approach to iamge segmentation. However, a inversion of the perturbation is needed in order to heavily speed up the segementation calculation itself. In general such an inversion is not given for the mentioned papers. If such an addition could be figured out, it would definitely help some people. This point is open for discussion.

References

[1] Ronneberger, Fischer, and Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation" (https://arxiv.org/abs/1505.04597)

[2] Çiçek et al., "3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation" (https://arxiv.org/abs/1606.06650)

[3] Sebastien C. Wong, Adam Gatt, Victor Stamatescu and Mark D. McDonnell. Understanding data augmentation for classification: when to warp?, 2016; arXiv:1609.08764.

[4] Alex Tong Lin, Yonatan Dukler, Wuchen Li and Guido Montufar. Wasserstein Diffusion Tikhonov Regularization, 2019; arXiv:1909.06860.