Invariant Information Clustering for Unsupervised Image Classification and Segmentation
(2019)

Xu Ji, Joo F. Henriques, Andrea Vedaldi

Notes

Contributions

The authors presents a novel clustering objective based on mutual information (MI) maximization, the training objective is to maximize the mutual information between the two outputs of two similar input, i.e. two images containing the same objects or the same image and it transformed version, pushing the model to focus on what is common between the two inputs and discarding low level and irrelevant details. Using the outputs in the form of discrete probabilities over the classes (or the clusters), the MI can be computed exactly.

Method

Invariant Information Clustering

Given two inputs x and x', the goal is to learn a representation Φ that only preserves what is common between the two similar inputs and discards what low level details, this can be achieved by maximizing the MI between the two outputs: $\max_{\Phi} I\left(\Phi(\mathbf{x}), \Phi\left(\mathbf{x}'\right)\right)$.

The two outputs, using a softmax, are in the form of probability distributions over C classes, where $\Phi(\mathbf{x}) \in [0,1]^{C}$, for a pair of cluster assignment z and z' of the two inputs x and x', the joint probability of these two predictions can be independent when conditioned on the inputs: $P\left(z=c, z'=c' | \mathbf{x}, \mathbf{x}'\right)=\Phi_{c}(\mathbf{x}) \cdot \Phi_{c'}\left(\mathbf{x}'\right)$. Which gives a zero mutual information, but in reality these the two variables are dependent given that the convet is trained on a set pairs over the whole dataset.

Computing the loss Given the two outputs $\Phi(\mathbf{x})$ and $\Phi(\mathbf{x}')$ which are probabilities of C possible classes, the objective is to compute the mutual information between the two output, first we need to compute the joint probability $\mathbf{P}_{\mathbf{c}\mathbf{c}'}$ and the two marginals $\mathbf{P}_{\mathbf{c}}$ and $\mathbf{P}_{\mathbf{c}'}$. Computing the marginal can be infeasible for continuous RV, but in this case the probabilities are discrete and simple and are easy to compute the marginals over. First given two vectors $\Phi(\mathbf{x})$ and $\Phi(\mathbf{x}')$ of size $C \times 1$, we compute the dot product to a get a matrix $\mathbf{P}$ of size $C \times C$, where each element is the joint probability $\mathbf{P}_{cc'} = P\left(z=c, z'=c'\right)$, this matrix is:

$$\mathbf{P}=\frac{1}{n} \sum_{i=1}^{n} \Phi\left(\mathbf{x}_{i}\right) \cdot \Phi\left(\mathbf{x}_{i}'\right)^{\top}$$

For the marginals $\mathbf{P}_{c}=P(z=c)$ and $\mathbf{P}_{c'}=P\left(z'=c'\right)$ we simply compute the sum over the rows and columns respectively. And given that we want to maximize the MI of both pairs: (x, x') and (x', x), the matrix $\mathbf{P}$ is symmetrized $\left(\mathbf{P}+\mathbf{P}^{\top}\right)/2$. Finally the MI is computed by the following:

$$I\left(z, z'\right)=I(\mathbf{P})=\sum_{c=1}^{C} \sum_{c'=1}^{C} \mathbf{P}_{cc'} \cdot \ln \frac{\mathbf{P}_{cc'}}{\mathbf{P}_{c} \cdot \mathbf{P}_{c'}}$$

Avoiding degenerate solutions In case of clustering, we might end up with all of the samples in the same cluster, or the other extreme case where each example is a cluster on its own, by using the MI: $I\left(z, z^{\prime}\right)=H(z)-H\left(z \mid z^{\prime}\right)$ we can avoid these problems. To maximize the MI, the conditional entropy needs to be minimized and the entropy $H(z)$ needs to be maximized, $H(z)$ is maximized when we have a maximum level of uncertainty, in this case $H(z)=\ln \frac{1}{C}$ and this occurs when all the data is randomly assigned to the C clusters, this way we avoid having only one clusters and also a maximum number of cluster, because in both cases the uncertainty is minimized. Now to counter this affect and this entropy maximization, which will give us uniform distributions, resulting in no clustering. The second term come into play, to minimize the conditional, the cluster of assignments must be exactly predictable from each other, and this can only be obtained if the outputs are one-hot vectors, so the two terms balance each other, between reinforcing the predictions and having equal mass between the clusters.

Overclustering or certain datasets (e.g.STL10), training data comes in two types: one known to contain only relevant classes and the other known to contain irrelevant or distractor classes, the authors use two head, the main head with C output classes, trained only using the relevant classes, and a second overclustering head (with output $C^{\prime} >> C$) that is trained using the istractor classes

0.0.1 Image Segmentation

For image segmentation, instead of having probabilities for each input, we have probability densities over each pixel, $\Phi(\mathbf{x}) \in [0,1] C \times H \times W$, the the goal is to maximize the MI between two patches, a path x_u centered at a pixel location u, and neighboring patch $x_(u+t)$ subject to a small translation t. Given a number of translations, we need to maximize the MI between x_u and all its neighbors that are t translation away. Now the second input can also be subject to a form of transformations, be it photometric or geometric, in case of geometric transformation g, the outputs need to be readjusted by the inverse of the geometric transformation g^{-1} (if we have the transformation function, we can use its inverse, or if not, we can add a bilinear upsampling network that will learn the inverse of the transformation). And the MI can then be computed over all locations t and transformation g:

$$\max_{\Phi} \frac{1}{|T|} \sum_{t \in T} I\left(\mathbf{P}_{t}\right)$$
$$\mathbf{P}_{t}=\frac{1}{n|G||\Omega|} \sum_{i=1}^{n} \sum_{a \in G} \sum_{u \in \Omega} \Phi_{u}\left(\mathbf{x}_{i}\right) \cdot\left[g^{-1} \Phi\left(g \mathbf{x}_{i}\right)\right]_{u+t}^{\top}$$

To compute the probability matrix, we can use convolution, where we convolve the two outputs aver the batch, H and W, so with two tensors of size C, B, H, W, we pad the output tensor computed using the second patch with d pixels ($2d+1=t$ are all the possible translation of the second patch over the first) that are then taken from the original image and convolve it with the second tensor, resulting in $[C, C, 2d+1, 2d+1]$, and we sum over all the locations $2d+1$.

Results

Evaluation: based on accuracy, true positives divided by sample size, the true positives are the best one-to-one permutation mapping between learned and ground-truth clusters

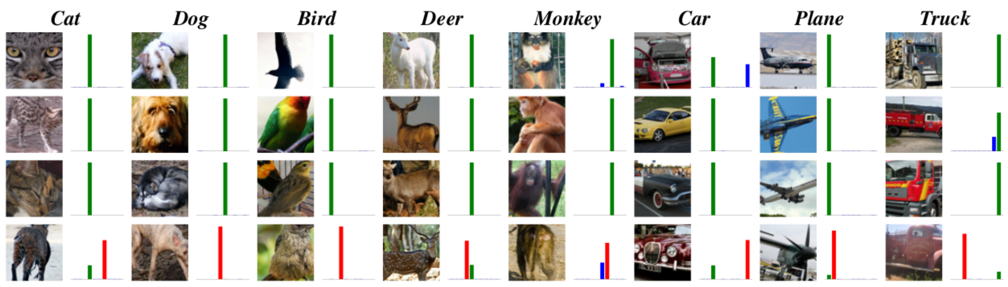

Figure 5: **Unsupervised image clustering (IIC) results on STL10.** Predicted cluster probabilities from the best performing head are shown as bars. Prediction corresponds to tallest, ground truth is green, incorrectly predicted classes are red, and all others are blue. The bottom row shows failure cases.

	STL10
Dosovitskiy 2015 [18]†	74.2
SWWAE 2015 [54]†	74.3
Dundar 2015 [19]	74.1
Cutout* 2017 [15]	87.3
Oyallon* 2017 [42]†	76.0
Oyallon* 2017 [42]	87.6
DeepCluster 2018 [7]	73.4⋆
ADC 2018 [24]	56.7⋆
DeepINFOMAX 2018 [27]	77.0
IIC plus finetune†	**79.2**
IIC plus finetune	**88.8**

Table 3: **Fully and semi-supervised classification.** Legend: *Fully supervised method. ⋆Our experiments with authors' code. †Multi-fold evaluation.

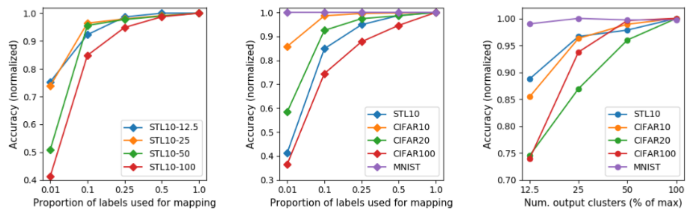

Figure 6: **Semi-supervised overclustering.** Training with IIC loss to overcluster ($k > k_{gt}$) and using labels for evaluation mapping only. Performance is robust even with 90%-75% of labels discarded (left and center). STL10-r denotes networks with output $k = \lceil 1.4r \rceil$. Overall accuracy improves with the number of output clusters k (right). For further details see supplementary material.

	STL10	CIFAR10	CFR100-20	MNIST
Random network	13.5	13.1	5.93	26.1
K-means [53]†	19.2	22.9	13.0	57.2
Spectral clustering [49]	15.9	24.7	13.6	69.6
Triplets [46]‡	24.4	20.5	9.94	52.5
AE [5]‡	30.3	31.4	16.5	81.2
Sparse AE [40]‡	32.0	29.7	15.7	82.7
Denoising AE [48]‡	30.2	29.7	15.1	83.2
Variational Bayes AE [34]‡	28.2	29.1	15.2	83.2
SWWAE 2015 [54]‡	27.0	28.4	14.7	82.5
GAN 2015 [45]‡	29.8	31.5	15.1	82.8
JULE 2016 [52]	27.7	27.2	13.7	96.4
DEC 2016 [51]†	35.9	30.1	18.5	84.3
DAC 2017 [8]	47.0	52.2	23.8	97.8
DeepCluster 2018 [7]† ‡	33.4⋆	37.4⋆	18.9⋆	65.6 ⋆
ADC 2018 [24]	53.0	32.5	16.0⋆	99.2
IIC (lowest loss sub-head)	**59.6**	**61.7**	**25.7**	**99.2**
IIC (avg sub-head ± STD)	59.8 ± 0.844	57.6 ± 5.01	25.5 ± 0.462	98.4 ± 0.652

Table 1: **Unsupervised image clustering.** Legend: †Method based on k-means. ‡Method that does not directly learn a clustering function and requires further application of k-means to be used for image clustering. ⋆Results obtained using our experiments with authors' original code.

	STL10
No auxiliary overclustering	43.8
Single sub-head ($h = 1$)	57.6
No sample repeats ($r = 1$)	47.0
Unlabelled data segment ignored	49.9
Full setting	**59.6**

Table 2: **Ablations of IIC (unsupervised setting).** Each row shows a single change from the full setting. The full setting has auxiliary overclustering, 5 initialisation heads, 5 sample repeats, and uses the unlabelled data subset of STL10.