A Deep Learning Approach to Unsupervised Ensemble Learning

Presented By:
Shipra Saini
Aishwarya Pothula

Abstract

Goal(S)

To show that DL can be applied to crowd sourcing and unsupervised ensemble learning problems. One of the main aims of the paper is to show that DL can be applied to Unsupervised Ensemble Learning in which the CI assumption is violated

Refresher(A)

Unsupervised ML : no ground truth

Crowdsourcing : obtaining data/predictions from different sources/classifiers

Ensemble Learning : combining the predictions of several models/classifiers/annotators into a single prediction

Restricted Boltzman Machine : special class of boltzman machines. They are restricted in terms of the connections

Conditional Independence : Given C, A & B are independent. Occurance of A does not indicate the likelihood of B

David and Skene Model : Ensemble Learning. Two assumptions

i) The classifiers make perfectly independent errors

ii) The errors are uniformly distributed across the input space

Related Work & Motivation(A)

Related Work(A)

[1] DS model is a non-convex optimization. Models that are asymptotically consistent under the DS model were researched

[2] Tried to address the second assumption of DS (uniform distribution of errors across all instances) by
Building richer models with more parameters such as instance difficulty
Using annotators with varying skills across the input space etc

[3] Comparatively, few works have considered addressing the first assumption of DS(conditional independence of classifiers). Work by Jaffe being the closest to the paper approach.

Motivation (A)

DL has performed extremely well in ML and AI applications. The motivation of this paper is to show that DL can be applied to crowdsourcing and ensemble learning as well to achieve state-of-the-art results.

Reasons for choosing DL approach(A)

> Ability to disentangle factors of variation in the inputs- hidden units become less statistically dependent

> Expressive power - studies show that some functions are more efficiently(fewer units) represented by deeper networks than shallower ones

Approach, Contributions and Result Summary

Approach(S)

-> Show that DS model has an equivalent parameterization in terms of an RBM with a single hidden node.
-> Now, the posterior probability of true labels (determining class) can be estimated via a trained RBM
-> To address the general case where classifiers are possibly dependent
> Construct an RBM based DNN (stacked RBMs)
> Use the DNN to perform Ensemble Learning

Contribution(S)

Proving that the DS model is equivalent to an RBM with a single hidden node
Show that RBM based DNN can be applied to unsupervised ensemble learning to tackle violations of CI
Putting forward a heuristic for determining the DNN architecture

Result Summary(A)

> Approach compared to several state-of-the-art methods based both on condition of CI and relaxations of it
> Found DNN approach performs better than other methods, in most cases, both on simulated & real data
> Demonstrate that in some cases, learned features in the last hidden layer of the DNN are perfectly uncorrelated while the raw data contained correlated features.

Notation and Problem Setup(A)

X, H, Y are random *variables, pθ, pλ are probability densities parametrized θ,λ respectively.*

Pθ as distribution generating the data and pλ as the RBM model distribution.

d and n are the input data dimensions and sample size respectively.

Let X ∈ {0, 1}d, Y ∈ {0,1} be random variables. The pair (X,Y) has joint distribution parameterized by θ given by

pθ(X,Y) = pθ(Y)pθ(X|Y). The joint distribution pθ(X,Y) and marginals pθ(X),pθ(Y) are not known.

In unsupervised ensemble learning we observe x1,x2,..,xn and learning task is to recover y1,y2,....,yn.

The binary vector X = (X1, X2, ….., Xd)T contains the predictions of d classifiers on instance whose label Y is unobserved.

RBM in Conditional Independence Case(A)

Lemma 4.1. *The joint probability $p_\lambda(X = x, H = y)$ of a RBM with parameters $\lambda = (a, b, W)$ is equivalent to the joint probability $p_\theta(X = x, Y = y)$ of a conditional independence model with parameters $\theta = (\{\psi_i\}, \{\eta_i\}, \pi)$ given by*

$$\psi_i \equiv \sigma(a_i + W_i), \ \eta_i \equiv 1 - \sigma(a_i)$$

$$\pi \equiv \frac{\sum_{x \in \{0,1\}^d} e^{a^T x + b + x^T W}}{\sum_{x \in \{0,1\}^d} \left(e^{a^T x} + e^{a^T x + b + x^T W}\right)}$$

Furthermore, the map $\lambda \mapsto \theta$ is a bijection.

When the parameters of the DS model are as specified, the DS model is equivalent to an RBM with a single hidden node

To show equivalence, the authors depend on a special case of a result proved by Chang(2015) that makes the parameters of DS model identifiable.

Lemma 4.2. *Let $x^{(1)}, ..., x^{(n)}$ be observed data from the conditional independence model, specified by p_θ. Assume that θ is such that for each $i = 1, \ldots, d$, X_i is not independent of Y (i.e., each classifier is not just a random guess), and that $d \geq 3$. Let $\hat{\lambda}_{MLE}$ be a maximum likelihood parameter estimate of a RBM with a single hidden node. Then the RBM posterior probability $p_{\hat{\lambda}_{MLE}}(H = 1 | X = x)$ converges to the true posterior $p_\theta(Y = 1 | X = x)$, as $n \to \infty$.*

If we know the maximum likelihood λMLE of RBM and d>=3; joint prob p(x,y), RBM distribution converges to true posterior prob of DS

Remark 4.4. *Lemma 4.2 assumes that we found the MLE of the RBM parameters. Obtaining such a MLE is problematic for two main reasons. First, RBMs are typically trained to maximize a proxy for the likelihood, as the true likelihood is not tractable. Second, the RBM likelihood function is not concave, hence there are no guarantees that after training a RBM one obtains the maximum likelihood parameter $\hat{\lambda}_{MLE}$.*

Obtaining λMLE is not possible as RBM estimate a proxy of max likelihood-not tractable. Also, RBM likelihood is not concave.

RBM based DNN (S)

In such DNN, the hidden layer of each RBM is the input for successive RBM. The RBMs are trained one at a time from bottom to top i.e., DNN is trained in a layer-wise fashion.

Given training data x1,..,xn $\in$ {0,1}d, we start by training the bottom RBM, then obtain

the first layer hidden representation of the data by sampling h(i) from pλ(H|X = x(i)).

The vectors h(1),.....,h(n) are then used as a training set for the second RBM & so on.

In this paper, where the true label y is binary, the upper-most RBM in the DNN has

a single hidden unit, from which posterior probability *pθ(Y |X) can be estimated.*

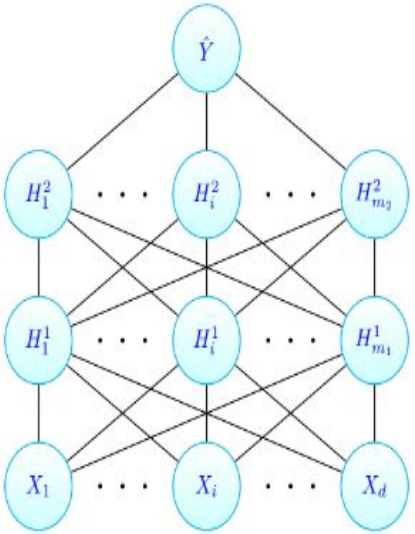


Figure 3. A sketch of RBM-based DNN with two hidden layers.

Choosing DNN Architecture(A)

Procedure

1. Train RBM with *d* hidden units.
2. Compute the singular value decomposition of the weight matrix *W,* and determine its rank(large singular values)
   Rule of Thumb : m set to min number of singular values whose cum sum is 95% of total sum

Rank is some m <= d, we re-train the RBM setting the number of hidden units to m. If m > 1, add another layer on top of current layer and proceed recursively. The process stops when m = 1, so last layer of DNN contains single node. This method is known as SVD approach.

Why SVD

This method takes advantage of the co-adaptation of hidden units. Co-adaptation is a situation where several hidden units tend to behave similarly. The rank of weight matrix might be small, although the number of hidden units may be larger.

Results- Compared with(S)

Compared with

VOTE - majority voting, assumed equal classifier accuracies, Ci of classifiers

DS - CI of classifiers,

CUBAM - CI of classifiers assumed,classifier accuracies vary across input domain

L-SML - CI assumption relaxed to a depth of 2 tree model

DNN - paper approach, depth and number of each layer determined by SVD.

Accuracy

$$\frac{\sum \mathbb{I}\{\text{true label is 0 and predicted label is 0}\}}{2\sum \mathbb{I}\{\text{true label is 0}\}} + \frac{\sum \mathbb{I}\{\text{true label is 1 and predicted label is 1}\}}{2\sum \mathbb{I}\{\text{true label is 1}\}},$$

Results- Datasets(S)

Simulated Datasets

CondInd- CI holds. 10 out 15 classifiers random guess

Tree15-3-1 - generated from depth2 tree, every node in intermediate layer connected to 5 nodes in base layer. CI violated

Layered Graph 15-5-5-1 - generated from a tree of depth 3 CI does not hold but data dependence not too high

TruncatedGaussian - x = (1+sign(Z))/2. Random variable Z -follows gaussian with different means. Data highly dependent

RealWorld Datasets

DREAM - gene mutation predictions of classifiers S1: d1 = 124, n = 92362; S2: d2 = 114, n = 70,561; S3 :d3 = 99, n= 78643

MAGIC 40 - physical measures of gamma particles, predict whether background or high energy. d = 16 n = 19020. 16 classifiers grouped into random forest, logistic trees, naive-bayes and SVM classifiers.

Results (A)

method	condInd	Tree15-3-1	LG15-5-5-1	TG
Vote	75.93 ± 0.5	93.45 ± 0.19	76.61 ± 0.09	80.14 ± 0.4
DS	$\mathbf{94.78 \pm 0.13}$	92.68 ± 0.14	86.36 ± 0.2	82.03 ± 0.27
CUBAM	91.96 ± 0.18	90.74 ± 0.3	77.12 ± 0.26	83.43 ± 0.31
L-SML	55.94 ± 21.88	$\mathbf{95.83 \pm 0.15}$	85.87 ± 0.21	79.5 ± 1.35
DNN	$\mathbf{94.78 \pm 0.13}$ (15-1)	95.13 ± 0.71 (15-3-1)	$\mathbf{86.83 \pm 0.2}$ (15-4-1)	$\mathbf{88.09 \pm 0.52}$ (15-3-1)
SUP	94.45 ± 0.11	95.54 ± 0.27	87.01 ± 0.18	90.8 ± 0.4
Bayes-Opt	95.32	96.12	87.05	91.39

DNN always outperforms Majority Vote and CUBAM

CondInd : Ds and DNN perform better than the supervised models and closer to the performance bayes. Architecture determined Single RBM with single hidden layer determined by SVD

Tree15-3-1 : L-SML outperforms the DNN and it is expected as it is tailored to the data generated by the tree. However, the performance of DNN is better than other unsupervised models and close to SUP and bayes

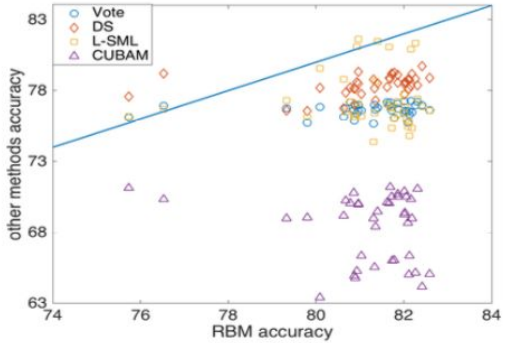
LG15-5-51 : DNN outperforms other methods but equal to SUP and Bayes. Architecture different from that of true data generation model

TG : In this dataset, CI is strongly violated and you can see that DNN outperform all other by a big margin

Results -continued

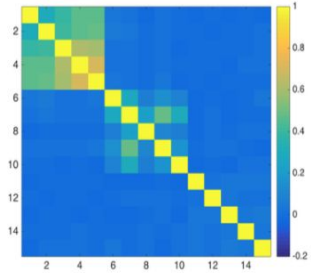
Dataset	Vote	DS	CUBAM	L-SML	DNN
S1	97.2 *	98.3 *	92.31	**98.4** *	**98.42** $\pm$ **0.0** (124-1)
S2	96 *	97.2 *	69.19	**97.7** *	97.55 $\pm$ 0.01 (114-1)
S3	95.7 *	97.7 *	87.65	98.2 *	**98.51** $\pm$ **0.01** (99-25-1)

DREAM dataset . L-SML and DNN outperform others, perform equally on s1. DNN better on s3 and L-SMl on S2

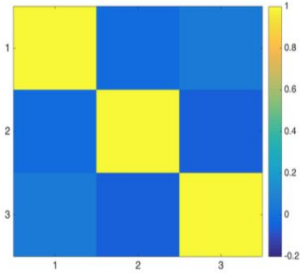


DNN outperforms. Architecture is 15-3-1. Performance compare with others using a t-test with p value<10p(-13).Null rejected in all 4 tests

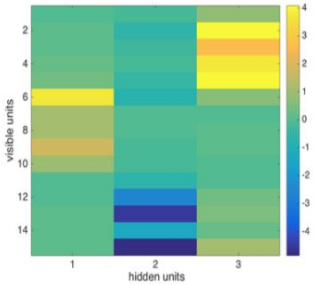
Tree15-3-1 experiment



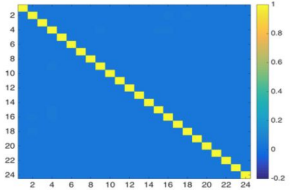
Correlation matrix for data for label y=0. Middle 5 x not CI



Correlation matrix of hidden layer of DNN. Hidden units uncorrelated



Weight matrix of bottom RBM of DNN



Hidden representation in s3 is perfectly uncorrelated.

Summary and Future Research Direction(A)

We demonstrate that deep learning techniques can be used for unsupervised ensemble learning and the DNN approach proposed in this paper often performs at least as well and often better than state-of the art methods, especially when the conditional independence assumption made by Dawid & Skene does not hold.

Future research include extending the approach to multiclass problems, theoretical analysis of the SVD approach.

THANK YOU