RHEA SHARMA **1001679252**

CSE 6363: Machine Learning
Assignment: Paper Review

1. Learning Social Networks from Web Documents Using Support Vector Classifiers

Masoud Makrehchi, Mohamed S. Kamel Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence. IEEE Computer Society, 2006.

Objective: To generate a social network from a collection of web documents.

Introduction:

- Has a set of actors/users and a mapping for each actor to a vector space document.
- Learning relations between different actors from vector space documents. SVM used for classification.
- Using a partially explored social network to predict and learn the entire social network.
- Let $A = \{a_1,a_2,\ldots,a_n\}$ be the set of actors. Maximum number of possible relations/ties are: $M = n(n-1) / 2$. Whereas in reality, the social network is sparse. The sparsity of a network is given by: $S = 1 – (2r / n-1)$.
- Social networks are represented by making use of adjacency matrices. Let $T = \{t_1, t_2,\ldots,t_q\}$ be the set of incomplete known relations. Let $D = \{d_1, d_2,\ldots,d_n\}$ be the set of actors textual documents.

Approach:

- Makes use of actor modeling & relationship modeling.
- Actor is represented by a set of documents like website, blog, resume, portfolio, etc.
- All the documents are mapped to a single document vector. In this vector space representation of an actor, each actor is represented by a set of unique terms.
- The global weights are calculated by two methods: local weighting and global weighting technique. TF-IDF is used for calculating the weights. latent semantic indexing (LSI) can be used to model actors.
- Estimate similarity of their document vectors and aggregate the document vectors of the actors.
- Performance Evaluation Measures done using F-score as micro averaged and macro averaged F-measure. This is used for calculating performance of classifier for both classes. 2 fold cross validation is used for estimating classifier performance. (macro avg).
- As there is imbalance in the data set (FOAF Database), the database is broken into small sub-graphs.

Experimental Results:

- Increase in the percentage of majority class, recall drops linearly while precision remains almost constant.
- A text classification formulation was proposed in order to approximately predict social relations using readily available web documents.
- Observed a high class imbalance in social networks.
- Modeling of relations between actor was done.
- Document vector aggregation.
- Overall macro-averaged F-measure was used to evaluate the extracted social network instead of micro-averaged F-measure as it gave a better result.
- High recall and low precision was observed.

2. Self-taught Learning: Transfer Learning from Unlabeled Data

Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, Andrew Y. Ng, Computer Science Department, Stanford University, CA 94305 USA

Objective: New machine learning frame- work called "self-taught learning" for using unlabeled data in supervised classification tasks.

Introduction:

- Used Large number of unlabeled images (or audio samples, or text documents) randomly downloaded from the Internet to improve performance on a given image (or audio, or text) classification task.
- Supervised learning task of interest motivated by the observation that even many randomly downloaded images will contain basic visual patterns (such as edges) used to recognize such patterns from the unlabeled data.
- Labeled data for machine learning is often very difficult and expensive to obtain, and thus the ability to use unlabeled data holds significant promise in terms of vastly expanding the applicability of learning methods.

Approach:

- Self-taught learning that uses sparse coding to construct higher-level features using the unlabeled data.
- Using an SVM for classification with Fisher kernel can be learned for this representation.
- Labeled training set of m examples $\{(x_l^{(1)}, y^{(1)}), (x_l^{(2)}, y^{(2)}), \ldots, (x_l^{(m)}, y^{(m)})\}$ drawn i.i.d from some distribution D. Each labeled $x_l^{(i)}$ ^{n} is an input feature vector $y^{(i)} \in \{1, \ldots, C\}$.
- In addition, we are given a set of k unlabeled examples $x_u^{(1)}, x_u^{(2)}, \ldots, x_u^{(k)} \in$, ^{n} .
- Clearly, In Transfer learning (Thrun, 1996; Caruana, 1997), the labeled and unlabeled data should not be completely irrelevant to each other if unlabeled data is to help the classification task.

Experimental Results:

- Given the labeled and unlabeled training set, a self- taught learning algorithm outputs a hypothesis h: $^n \rightarrow$ {1,...,C} - mimic the input-label relationship represented by the labeled training data; this hypothesis h is then tested under the same distribution D from which the labeled data was drawn.
- Explained few experiments and results shown for sparse coding bases learned on handwritten digits. Here, sparse coding features alone do not perform as well as the raw features but perform significantly better when used in combination with the raw features.
- Using the Fisher kernel derived from the generative model described, obtain a classifier customized specifically to the distribution of sparse coding features.

---

3. Robust Principal Component Analysis for Computer Vision

Fernando De la Torre, Michael J. Black 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on. Vol. 1. IEEE, 2001.

Objective: Proposal of a more robust way of implementing PCA and describing a robust M- estimation algorithm.
Introduction:

- PCA is mainly used for solving problems such as object recognition, face detection, tracking and background modelling.
- The Drawback of PCA: least square estimation technique which fails to account for outliers. This technique can potentially skew the solution from the desired solution.
- Hence, the authors proposed a more robust way of implementing PCA and describe a robust M- estimation algorithm.
- Treating data set with sample outliers. Treating data set with intra-sample outliers.
- Black and Jepson's way of robustly recovering the coefficients of a linear combination that reconstructs an input image.
- Xu and Yuille address the commonly used PCA learning rules which are first related to energy functions.
- These functions are generalized by adding a binary decision field with a given prior distribution so that outliers in the data are dealt with explicitly in order to make PCA robust.
- **PROBLEM:** As view of the object changes due to motion or when motion of camera fails to detect the same.

Approach & Results:

- Given a learned basis set, B and J addressed the issue of robustly recovering the coefficients of a linear combination that reconstructs an input image.
- Problems in Previous Approaches:
  i. A single "bad" pixel value can make an image lie far enough from the subspace that the entire sample is treated as an outlier (i.e. Vi = 0) and has no influence on the estimate of B.
  ii. Xu and Yuille use a least squares projection of the data di for computing the distance to the subspace; that is, the coefficients which reconstruct the data di are ci = BT di. These reconstruction coefficients can be arbitrarily biased for an outlier.
  iii. A binary outlier process is used which either completely rejects or includes a sample.
- Performed Quantitative Comparisons and Computational Issues were discussed.
- Presented a method for robust principal component analysis that can be used for automatic learning of linear models.
- Furthermore, it extends work in the statistics community by connecting the explicit outlier formulation with robust M-estimation.
- Torre and Black are working on applications for robust Singular Value Decomposition, generalizing to robustly factorising n-order tensors, on adding spatial coherence to the outliers and on developing a robust minor component analysis.

---

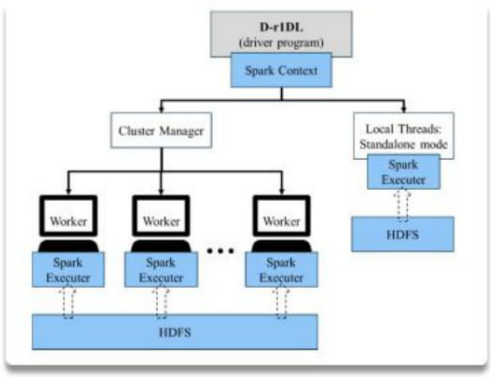
4. Scalable Fast Rank-1 Dictionary Learning for fMRI Big Data Analysis

Xiang Li1, Milad Makkie, Binbin Lin, Mojtaba Sedigh Fazli, Ian Davidson, Jieping Ye, Tianming Liu, Shannon Quinn, At 2016 KDD Conference held at SFO.

Objective: Design a novel distributed rank-1 dictionary learning (D-r1DL) model, leveraging the power of distributed computing for handling large scale fMRI big data.

Introduction & Approach:

- The learning process is a fix-point algorithm by alternating least squares updates, the memory cost - very low and very light weighted in terms of the operational complexities: besides the input data, most of the routines in the algorithm will only take one vector as input and one vector as output.
- This feature helps the r1DL algorithm to be easily parallelized to its distributed version.
- The basic Idea is the observed functional signals are the result of the linear combination from the signals of many latent sources (i.e. functional networks), plus noises.
- The methods then aim to identify the latent source signals as well as the loading matrix
- The decomposition results consist of two parts:
  i. Temporal pattern of the functional networks - regarded as basis activation patterns
  ii. Spatial pattern of the functional networks.



minimizing the following energy function $L(u, v)$:

$$L(u,v) = \|S - uv^T\|_F, \text{s.t.} \|u\| = 1, \|v\|_0 \le r. \quad (1)$$

$$v = \underset{v}{\text{argmin}} \|S - uv^T\|_F, s.t. \|v\|_0 \le r,$$

$$u = \underset{u}{\text{argmin}} \|S - uv^T\|_F = \frac{Sv}{\|Sv\|}, \quad (2)$$

Converging at step j if: $\|u^{j+1} - u^j\| < \varepsilon, \varepsilon = 0.01.$

$$R^n = R^{n-1} - v^T R^{n-1}, R^0 = S, 1 < n \le K, \quad (3)$$

- The distribution of S to each node as a series of key-value pairs is inherently straight forward: each column in S contains the T number of observations for one specific feature, to the total of P features. While S was maintained as an RDD, the vectors u and v were broadcast to all nodes.
- Vector-matrix multiplication: each node will use its portion of the updated u vector, and then estimate the v vector based on the multiplication of portions of S and u. The resulting v vectors from all the nodes will be then map-reduced by the summation operation.
- Matrix-vector multiplication: each node will use all the updated v vector then estimate its corresponding portion of the u vector.
- Experiments and results are shown. Dictionary learning model based on iterative rank-1 basis estimation.
- The model was implemented and parallelized in Spark, and then deployed using the in-house solution as well as the AWS-EC2 solution.

Result: The final goal is to provide an integrated solution for functional neuroimaging big data management and analysis, enabling high throughput neuroscientific knowledge discovery and similar parallelization scheme could be implemented on other algorithms as well.

---

5. Supervised Dictionary Learning

Mairal, Julien, et al. Advances in neural information processing systems. 2009.

Objective: Introduce supervised dictionary learning generative/discriminative framework using sparse models.

Introduction:

- In Sparse Coding, the signal x in R ^n , Dictionary D=[d1,d2,…,dk] in R where R has dimensions n x k. Initially k >n, Sparse coding with l1 regularization.

$$\mathcal{R}^\star(\boldsymbol{x}, \mathbf{D}) = \min_{\boldsymbol{\alpha} \in \mathbb{R}^k} ||\boldsymbol{x} - \mathbf{D}\boldsymbol{\alpha}||_2^2 + \lambda_1 ||\boldsymbol{\alpha}||_1$$

- Models for classification task using sparse code 1. Linear(α) and 2. Bilinear(x, α).

Approach:

- **Linear Model** has a probabilistic interpretation. Graphical Model providing probabilistic interpretation while using linear model with no bias to coefficient:

  i. Gaussian on w, $p(\mathbf{w}) \propto e^{-\lambda_2 ||\mathbf{w}||_2^2}$, constraint on D is $||\mathbf{d}_j||_2^2 = 1$ for all j and αi with Laplace prior, $p(\boldsymbol{\alpha}_i) \propto e^{-\lambda_1 ||\boldsymbol{\alpha}_i||_1}$.

- Generative training – finds maximum likelihood estimates of D and w based on Joint distribution $p(\{\boldsymbol{x}_i, y_i\}_{i=1}^m, \mathbf{D}, \mathbf{W})$.

- Discriminative training – maximum of: $p(\{y_i\}_{i=1}^m, \mathbf{D}, \mathbf{w} | \{\boldsymbol{x}_i\}_{i=1}^m)$.

- **Bilinear Model** ($f(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\theta}) = \boldsymbol{x}^T \mathbf{W} \boldsymbol{\alpha} + b$) can be interpreted in terms of kernel instead of probabilistic interpretation

- We have Kernel K: $K(\boldsymbol{x}_1, \boldsymbol{x}_2) = \boldsymbol{\alpha}_1^T \boldsymbol{\alpha}_2 \boldsymbol{x}_1^T \boldsymbol{x}_2$.

- Above kernel is a product of two linear kernels, one on α and input signal x

- Raina et in ICML 2007 ‘Self-taught learning: transfer learning from unlabeled data ‘ learn a dictionary adapted to reconstruction on a training set, then train an SVM a posteriori on the decomposition coefficients. A Supervised Sparse Coding Algorithm is explained:

Input: n (signal dimensions); $(\boldsymbol{x}_i, y_i)_{i=1}^m$ (training signals); k (size of the dictionary); $\lambda_0, \lambda_1, \lambda_2$ (parameters); $0 \leq \mu_1 \leq \mu_2 \leq \ldots \leq \mu_m \leq 1$ (increasing sequence).
Output: $\mathbf{D} \in \mathbb{R}^{n \times k}$ (dictionary); $\boldsymbol{\theta}$ (parameters).
Initialization: Set $\mathbf{D}$ to a random Gaussian matrix with normalized columns. Set $\boldsymbol{\theta}$ to zero.
Loop: For $\mu = \mu_1, \ldots, \mu_m$,
  Loop: Repeat until convergence (or a fixed number of iterations),
  • *Supervised sparse coding:* Solve, for all $i = 1, \ldots, m$,

$$\begin{cases} \boldsymbol{\alpha}_{i,-}^\star = \arg\min_{\boldsymbol{\alpha}} \mathcal{S}(\boldsymbol{\alpha}, \boldsymbol{x}_i, \mathbf{D}, \boldsymbol{\theta}, -1) \\ \boldsymbol{\alpha}_{i,+}^\star = \arg\min_{\boldsymbol{\alpha}} \mathcal{S}(\boldsymbol{\alpha}, \boldsymbol{x}_i, \mathbf{D}, \boldsymbol{\theta}, +1) \end{cases}. \quad (10)$$

  • *Dictionary and parameters update:* Solve

$$\min_{\mathbf{D}, \boldsymbol{\theta}} \Big(\sum_{i=1}^m \mu \mathcal{C}\big((\mathcal{S}(\boldsymbol{\alpha}_{i,-}^\star, \boldsymbol{x}_i, \mathbf{D}, \boldsymbol{\theta}, -y_i) - \mathcal{S}(\boldsymbol{\alpha}_{i,+}^\star, \boldsymbol{x}_j, \mathbf{D}, \boldsymbol{\theta}, y_i))\big) + (1-\mu)\mathcal{S}(\boldsymbol{\alpha}_{i,y_i}^\star, \boldsymbol{x}_i, \mathbf{D}, \boldsymbol{\theta}, y_i) + \lambda_2 ||\boldsymbol{\theta}||_2^2 \Big) \text{ s.t. } \forall j, ||\mathbf{d}_j||_2 \leq 1. \quad (11)$$

Results & Conclusion:

- Experimental Validations are performed, and error rates were given. For the Texture Classification, the authors observed that linear model works better than bilinear. The reason is simplicity of task. The BL is worth using when we do the below steps:
  i. Initially two images from Brodatz dataset are chosen.
  ii. Build two classes for these images composed of 12 x 12 patches taken from those two textures.
  iii. Comparison is made for classification performance of all methods for dictionary.
- A discriminative approach to supervised dictionary learning has been successfully introduced which efficiently exploits the exploits the corresponding sparse signal decompositions in image classification tasks.
- An effective method has been proposed for learning a shared dictionary multiple models such as linear or bilinear.
- Future work will be in the direction of adapting the proposed framework to shift-invariant models that are standard in image processing tasks, but not readily generalized to the sparse dictionary learning setting.
- Moreover, investigation is extended to unsupervised and semi-supervised learning and applications to natural image classification.

---

6. Why significant variables aren't automatically good predictors

Adeline Lo, Herman Chernoff, Tian Zheng and Shaw-Hwa Lo Proceedings of the National Academy of Sciences 112.45 (2015): 13892-13897.

Objective: Demonstrate the Inability to use the results of the identified statistically significant variables.

Introduction & Approach:

- "Why Significant Variables not leading to good predictions of the outcome?"
- Highly significant: uses assumption, but no knowledge of exact Distributions

$$\sum_{x:f_D(x)<f_H(x)} f_D(x) \text{ and } \sum_{x:f_D(x)\geq f_H(x)} f_H(x).$$

$$\text{prediction rate} = 0.5 \sum_x \max(f_D(x),\ f_H(x)).$$

- Highly Predictive uses knowledge of both fh and fd
- Few examples (predictive variable sets and Significant variable sets) are provided along with graphs. Compared Significance test with I score. The taller and lighter a bar, the more important the VS.
- Applied I score to Real Breast Cancer Data.
- **I-Score**: Y_i= i^{th} individual, $\overline{Y}$ = Mean of all Y values , $\overline{Y}$ = Mean of all Y values in cell j, s = SD of all Y values, n_j = No of individuals in cell j, n = total no of individuals,

$$I = \sum_{j=1}^{m_1} \frac{n_j}{n} \frac{(\overline{Y}_j - \overline{Y})^2}{s^2/n_j} = \frac{\sum_{j=1}^{m_1} n_j^2 (\overline{Y}_j - \overline{Y})^2}{\sum_{i=1}^{n} (Y_i - \overline{Y})^2},$$

Results & Conclusion:

- Need to know the underlying distribution, to apply efficient techniques. Real examples are difficult to analyze because of large number of variables.
- Exploration away from significance-based methodologies and toward prediction-oriented ones is encouraged.

---

7. Online Dictionary Learning for Sparse Coding

Mairal, Julien, et al., Proceedings of the 26th annual international conference on machine learning. ACM, 2009.

Objective: The objective of this paper is to learn the dictionary (basis set). It has also proposed a new online optimization algorithm for faster and better performance on large scale datasets.

Introduction:

- Sparse coding is a class of unsupervised methods for learning sets of over-complete bases to represent data effectively and efficiently. It means finding a set of basis vectors "Φ" _i which is modeled as sparse linear combinations of basis elements. It is widely used in machine learning, neuroscience, signal processing, and statistics. $\mathbf{x} = \sum_{i=1}^{k} a_i \phi_i$
- Consider a signal *x* in R^m. We say that it admits a sparse approximation over a dictionary D in R^(m×k), with k columns referred to as atoms, when one can find a linear combination of a "few" atoms from D that is "close" to the signal *x*.
- It is shown in this paper that it is possible to go further and exploit the specific structure of sparse coding in the design of an optimization procedure dedicated to the problem of dictionary learning, with low memory consumption and lower computational cost than classical second-order batch algorithms and without the need of explicit learning rate tuning.

Problem Statement:

- Classical dictionary learning techniques (Olshausen & Field, 1997; Aharon et al., 2006; Lee et al., 2007) consider a finite training set of signals X = [x1, . . . , xn] in R^(m×k) and optimize the empirical cost function: $f_n(\mathbf{D}) \triangleq \frac{1}{n}\sum_{i=1}^{n} l(\mathbf{x}_i, \mathbf{D}),$ where D in R^(m×k) is the dictionary, each column representing a basis vector, and l is a loss function such that l(x, D) should be small if D is "good" at representing the signal x.
- The problem of minimizing the empirical cost f_n(D) can be rewritten as a joint optimization problem with respect to the dictionary D and the coefficients α = [α1, . . . , αn] of the sparse decomposition, with respect to each of the two variables D and α when the other one is fixed:

$$\min_{\mathbf{D}\in\mathcal{C},\boldsymbol{\alpha}\in\mathbb{R}^{k\times n}} \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{2}||\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i||_2^2 + \lambda||\boldsymbol{\alpha}_i||_1\right).$$

Approach:

- **First Approach:** Assuming the training set composed of i.i.d. samples of a distribution p(x), its inner loop draws one element x_t at a time, as in stochastic gradient descent, and alternates classical sparse coding

steps for computing the decomposition α_t of x_t over the dictionary D_(t-1) obtained at the previous iteration, with dictionary update steps where the new dictionary D_t is computed by minimizing over C the function where the vectors α_i are computed during the previous steps of the algorithm.

- **Second Approach:** Algorithm 2 sequentially updates each column of D. Since the algorithm uses the value of D_(t-1) as a warm restart for computing D_t, a single iteration has been found to be enough.

Algorithm 1 Online dictionary learning.

Require: $\mathbf{x} \in \mathbb{R}^m \sim p(\mathbf{x})$ (random variable and an algorithm to draw i.i.d samples of p), $\lambda \in \mathbb{R}$ (regularization parameter), $\mathbf{D}_0 \in \mathbb{R}^{m \times k}$ (initial dictionary), T (number of iterations).

1: $\mathbf{A}_0 \leftarrow 0$, $\mathbf{B}_0 \leftarrow 0$ (reset the "past" information).
2: **for** $t = 1$ to T **do**
3: Draw $\mathbf{x}_t$ from $p(\mathbf{x})$.
4: Sparse coding: compute using LARS

$$\boldsymbol{\alpha}_t \triangleq \arg\min_{\boldsymbol{\alpha} \in \mathbb{R}^k} \frac{1}{2}||\mathbf{x}_t - \mathbf{D}_{t-1}\boldsymbol{\alpha}||_2^2 + \lambda||\boldsymbol{\alpha}||_1. \quad (8)$$

5: $\mathbf{A}_t \leftarrow \mathbf{A}_{t-1} + \boldsymbol{\alpha}_t \boldsymbol{\alpha}_t^T$.
6: $\mathbf{B}_t \leftarrow \mathbf{B}_{t-1} + \mathbf{x}_t \boldsymbol{\alpha}_t^T$.
7: Compute $\mathbf{D}_t$ using Algorithm 2, with $\mathbf{D}_{t-1}$ as warm restart, so that

$$\mathbf{D}_t \triangleq \arg\min_{\mathbf{D} \in \mathcal{C}} \frac{1}{t} \sum_{i=1}^{t} \frac{1}{2}||\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i||_2^2 + \lambda||\boldsymbol{\alpha}_i||_1,$$
$$= \arg\min_{\mathbf{D} \in \mathcal{C}} \frac{1}{t}\left(\frac{1}{2}\mathrm{Tr}(\mathbf{D}^T\mathbf{D}\mathbf{A}_t) - \mathrm{Tr}(\mathbf{D}^T\mathbf{B}_t)\right). \quad (9)$$

8: **end for**
9: **Return** $\mathbf{D}_T$ (learned dictionary).

Algorithm 2 Dictionary Update.

Require: $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_k] \in \mathbb{R}^{m \times k}$ (input dictionary),
$\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_k] \in \mathbb{R}^{k \times k} = \sum_{i=1}^{t} \boldsymbol{\alpha}_i \boldsymbol{\alpha}_i^T$,
$\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_k] \in \mathbb{R}^{m \times k} = \sum_{i=1}^{t} \mathbf{x}_i \boldsymbol{\alpha}_i^T$.

1: **repeat**
2: **for** $j = 1$ to k **do**
3: Update the j-th column to optimize for (9):

$$\mathbf{u}_j \leftarrow \frac{1}{\mathbf{A}_{jj}}(\mathbf{b}_j - \mathbf{D}\mathbf{a}_j) + \mathbf{d}_j.$$
$$\mathbf{d}_j \leftarrow \frac{1}{\max(||\mathbf{u}_j||_2, 1)}\mathbf{u}_j. \quad (10)$$

4: **end for**
5: **until convergence**
6: **Return** $\mathbf{D}$ **(updated dictionary).**

- **Optimizing the algorithm:** The algorithm was optimized by:
  i. Handling Fixed-Size Datasets
  ii. Mini-Batch Extension
  iii. Purging the Dictionary from Unused Atoms

Results & Conclusion:

- More experiments are of course needed to better assess the promise of this approach in image restoration tasks such as denoising, deblurring, and inpainting.
- Plan to use the proposed learning framework for sparse coding in computationally demanding video restoration tasks (Protter & Elad, 2009), with dynamic datasets whose size is not fixed.
- Plan to extend this framework to different loss functions to address discriminatory tasks such as image classification (Mairal et al., 2009), which are more sensitive to overfitting than reconstructive ones.

---

8. A Deep Learning Approach to Unsupervised Ensemble Learning

Uri Shaham, Xiuyuan Cheng, Omer Dror, Ariel Jaffe, Boaz Nadler, Joseph Chang, Yuval Kluger

Objective: Demonstrate that deep learning can be applied to crowd sourcing and unsupervised ensemble learning problems. The goal of this paper is to show that deep learning can be applied to Unsupervised Ensemble Learning in which the CI assumption is violated.

Introduction & Approach:

- DS model has an equivalent parameterization in terms of an RBM with a single hidden node. To address the general case where classifiers are possibly dependent.
- Construct an RBM based DNN (stacked RBMs).

- Use the DNN to perform Ensemble Learning:
  - **i.** Train RBM with d hidden units.
  - **ii.** Compute the singular value decomposition (SVD) of the weight matrix W, and determine its rank (large singular values)
- Rule of Thumb: m set to minimum number of singular values whose cum sum is 95% of the 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.

Results & Conclusion:

- 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.
- Demonstrated 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 the state-of the art methods, especially when the conditional independence assumption made by Dawid & Skene does not hold. Future research can include extending the approach to multiclass problems, theoretical analysis of the SVD approach.

---

9. Efficient Sparse Coding Algorithms

Lee, Honglak, et al., Advances in neural information processing systems. 2007.

Objective: To demonstrate that the inferred sparse codes exhibit end-stopping and non-classical receptive field surround suppression that is exhibited by V1 neurons by applying these algorithms to natural images.

Introduction:

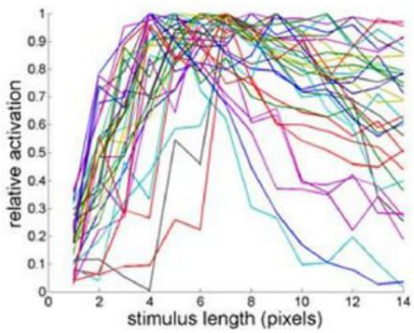
- Sparse coding is a kind of unsupervised learning in which given unlabeled data, it discovers basis functions that identifies higher-level features in the data.
- In sparse coding number of bases obtained is greater than the input dimension unlike other unsupervised learning techniques like PCA.
- In some cases, basis can be overcomplete, i.e., n>k.

Approach:

- The ultimate goal of sparse coding is to reconstruct input vectors approximately as a weighted linear combination of a small number of basis vectors.The reconstruction error must be minimized.
- The optimization problem is convex in B (while holding S fixed) and convex in S (while holding B fixed), but not convex in both simultaneously. For learning the bases B, the optimization problem is a least squares problem with quadratic constraints. For learning the coefficients S, the optimization problem is equivalent to a regularized least squares problem.
- **Feature sign search algorithm:** Maintains an active set of potentially nonzero coefficients and all other coefficients must be zero and systematically searches for the optimal active set and coefficient signs. The algorithm proceeds in a series of "feature-sign steps".

i. In every step, it is given a current guess for the active set and the signs, and it computes the analytical solution xˆnew to the resulting unconstrained QP; it then updates the solution, the active set and the signs using an efficient discrete line search between the current solution and xˆnew.

- Analysis on learning highly overcomplete natural image bases:



Conclusion:

- It is a novel algorithm.
- It is one of the first efficient sparse coding algorithms developed.
- Authors justified the explanation of neuron's end-stopping and surround suppression phenomenon using sparse coding.
- It produced great results when tested on natural images.

---

10. Support Vector Machine Active Learning for Image Retrieval

Simon Tong Proceedings of the ninth ACM international conference on Multimedia. ACM, 2001.

Objective: To demonstrate the use of a support vector machine active learning algorithm for doing relevance feedback for image retrieval.

Introduction:

- With Image databases, it is difficult to specify queries directly and explicitly. Relevance feedback is often a critical component when designing image databases. Relevance feedback interactively determines a user's desired output or query concept by asking the user whether proposed images are relevant or not.
- For this to be effective, it must grasp a user's query concept: accurately & quickly. To be done only by asking the user to label a few images.
- The algorithm selects the most informative images to query a user and quickly learns a boundary that separates the images that satisfy the user's query concept from the rest of the dataset.

Approach:

- **SVM_{active}:**
  i. It works by combining the following three ideas: SVM_{Active} regards the task of learning a target concept as one of learning a SVM binary classifier.
  ii. Captures the query concept by separating the relevant images from the irrelevant images.

iii. SVM_{Active} learns the classier quickly via active learning.

iv. The active part of SVM_{Active} selects the most informative instances with which to train the SVM classifier. This step ensures fast convergence to the query concept in a small number of feedback rounds. Once the classifier is trained, SVM_{Active} returns the top-k most relevant images.

v. SVM and Version Space are discussed. SVMs find the hyperplane that maximizes the margin in the feature space F.

$$\begin{aligned} &\text{maximize}_{\mathbf{w}\in\mathcal{F}} && \min_i\{y_i(\mathbf{w}\cdot\Phi(\mathbf{x}_i))\} \\ &\text{subject to:} && \|\mathbf{w}\| = 1 \\ & && y_i(\mathbf{w}\cdot\Phi(\mathbf{x}_i)) > 0 \quad i = 1\ldots n \end{aligned}$$

vi. This helps us to find the point in the version space that maximizes the minimum distance to any of the delineating hyperplanes.

- **Active Learning:**
  i. It is assumed that the instances x are independently and identically distributed according to some underlying distribution F(x) and label according to some conditional distribution P(y|x).
  ii. An unlabeled pool U, an active learner has three components: (f ,q , X). f is a classifier, f : X -> {-1,1} trained on the current set of labeled data X (and possibly unlabeled instances in U too). q(X) is the querying function that, given a current labeled set X, decides which instance in U to query next.
  iii. The active learner can return a classifier f after each pool-query (online learning) or after some fixed number of pool-queries.
  iv. The main difference between an active learner and a regular passive learner is the querying component q.
  v. We wish to reduce the version space as fast as possible. One good way of doing this is to choose a pool-query that halves the version space.
  vi. The hyperplane that is closest to wi is b, so we will choose to query b.
- **Active Learning Algorithm:**
  i. Our SVM_{Active} system performs the following for each round of relevance feedback:
    1. Learn an SVM on the current labeled data.
    2. If this is the first feedback round, ask the user to label twenty randomly selected images. Otherwise, ask the user to label the twenty pool images closest to the SVM boundary.
  ii. After the relevance feedback rounds have been performed SVM_{Active} retrieves the top-k most relevant images:
    1. Learn a final SVM on the labeled data.
    2. The final SVM boundary separates “relevant” images from irrelevant ones. Display the k “relevant” images that are farthest from the SVM boundary.

Results & Conclusion:

- Experiments are performed and results are shown. Active learning with SVM can provide a powerful tool for searching image databases, outperforming a number of traditional query refinement schemes.
- SVM_{Active} not only achieves consistently high accuracy on a wide variety of desired returned results, but also does it quickly and maintains high precision when asked to deliver large quantities of images.
- The running time of the algorithm scales linearly with the size of the image database both for the relevance feedback phase and for the retrieval of the top-k images.
- SVM_{active} is only practical when image database contains a few thousand images, authors are looking for ways for storing to larger sized databases. Authors are also finding ways for using experiment output to explore feature space until single relevant image is identified.

- An alternative approach for finding a single relevant image is to use another algorithm to seed SVM_{Active}. For example, the MEGA algorithm that authors have developed in a separate study does not require seeding with a relevant image.
- Transduction can be combined with active learning to provide improvement in performance for the task.