**Paper Summaries**

**Swetha Vijaya Raghavan | 1001551229 | sxv1229**

The following papers have been summarized.

| **1. 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 |

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

**Methodologies Discussed in the Paper:**

- 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.

**Problem formalism & Results:**

- 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 $x_l^{(i)} \in \mathbb{R}^n$ is an input feature vector $y^{(i)} \in \{1, \ldots, C\}$.
- In addition, we are given a set of k unlabeled examples $x_u^{(1)}, x_u^{(2)}, \ldots, x_u^{(k)} \in \mathbb{R}^n$.
- 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.
- Given the labeled and unlabeled training set, a **self- taught learning** algorithm outputs a hypothesis h: $\mathbb{R}^n \rightarrow \{1,\ldots,C\}$ - 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.

**Summary:** Using the Fisher kernel derived from the generative model described, obtain a classifier customized specifically to the distribution of sparse coding features.

| **2. 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. |

- Inability to use the results of the identified statistically significant variables

**Problem Statement**

- "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 defined:**

• $Y_i$= $i^{th}$ individual, $\bar{Y}$ = Mean of all Y values , $\bar{Y}$ = 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

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

**Summary**

- 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.

- The partition retention method, helps in reducing prediction error from 30% to 8% on a long-studied breast cancer data set.

| **3.A Bayesian Approach to Filtering Junk E-Mail** |
|---|
| Sahami, Mehran, et al.Learning for Text Categorization: Papers from the 1998 workshop. Vol. 62. 1998. |

- Appears to be the straight-forward text classification problem but can produce much more accurate filters. Finally, they showcased the efficacy of such filters in a real-world usage scenario, arguing that this technology is mature enough for deployment.
- With the use of the extensible framework of Bayesian modeling, the authors can not only employ traditional document classification techniques based on the text of messages but can also easily incorporate domain knowledge about the particular task at hand through the introduction of additional features in our Bayesian classifier.

**Probabilistic Classification**

- A Bayesian network is a directed, acyclic graph that compactly represents a probability distribution. It contains a node C representing the class variable and a node Xi for each of the features.

$$P(C = c_k \mid \mathbf{X} = \mathbf{x}) = \frac{P(\mathbf{X} = \mathbf{x} \mid C = c_k)P(C = c_k)}{P(\mathbf{X} = \mathbf{x})}. \quad (1)$$

The critical quantity in Equation 1 is P(X = x| C= ck), which is often impractical to compute without imposing independence assumptions. The main assumption of the Naive Bayesian classifier which assumes that each feature Xi **is conditionally independent** of every other feature, given the class variable C.

$$P(\mathbf{X} = \mathbf{x} \mid C = c_k) = \prod_i P(X_i = x_i \mid C = c_k).$$

Few experiments performed for junk E-mail detection. Both the performance of various enhancements to the simple baseline classification based on the raw text of the messages, as well as looking at the efficacy of learning such a junk filter in an operational setting are measured.

- Authors first employed a Zipf's Law-based analysis of the corpus of E-mail messages to eliminate words that appear fewer than three times as having little resolving power between messages.
- In examining the growing problem of dealing with junk E-mail, it is found that it's possible to automatically learn effective filters to eliminate a large portion of such junk from a user's mail stream.The efficacy of such filters can also be greatly enhanced by considering not only the full text of the E-mail messages to be filtered, but also a set of hand-crafted features which are specific for the task at hand.

- By the use of extensible classification formalism such as Bayesian networks, it becomes possible to easily and uniformly integrate such domain knowledge into the learning task

| **4. 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 |

- The paper tries to generate a social network from a collection of web documents.
- With a set of actors(users) and a mapping for every person to a vector space document.
- Then we learn the relations between different actors from these vector space documents. Used SVM for classification.
- With a partially explored social network, predict and learn the rest of the social network.
- Let $A = \{a_1,a_2,\ldots,a_n\}$ be the set of actors. Maximum number of possible relations(ties) are: $M = n(n-1) / 2$. But in reality, the social network is extremely sparse. The sparsity of a network is given by: $S = 1 – (2r / n-1)$.
- Social networks are represented by adjacency matrix. 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.

**Proposed Approach: Learning social Network from Incomplete network (Actor Modeling & Relationship Modeling)**

- An actor is represented by a set of documents like Resume, portfolio, etc.
- All these documents are mapped to a single document vector. In this vector space representation of an actor, each user is represented by a set of terms.
- The global weights are calculated by local weighting and global weighting technique. We use tf-idf for calculating the weights. Latent semantic indexing is used to model actors.
- Estimate similarity of their document vectors and Aggregate the document vectors of the actors.
- Performance Evaluation Measures with F-score as Micro averaged and macro averaged F-measure used for performance of classifier for both classes. 2 fold cross validation for estimating classifier performance. (macro avg).
- Data Set(FOAF Database) as it is unbalanced, breaking the database into small sub graphs.

**Experimental Results:**

- Increased the percentage of majority class, recall linearly drops while precision remains almost constant.
- A text classification formulation to approximately predict social relations using web documents were proposed.
- High class imbalance in social networks.
- Modeling relation between actors.
- Document vector aggregation
- Overall macro-averaged F-measure was used to evaluate the extracted social network.

- High recall and low precision.

| **5. 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. |

- 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.

**Robustness of previous PCA methods**:

- 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.
- **THEIR APPROACH**: 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:**
    - 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.
    - 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.
    - 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 factorizing n-order tensors, on adding spatial coherence to the outliers and on developing a robust minor component analysis.

| **6. Supervised Dictionary Learning** |
|---|
| Mairal, Julien, et al. Advances in neural information processing systems. 2009. |

- This paper talks about Supervised dictionary learning generative/discriminative framework using sparse models. 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)
- **Linear Model** has probabilistic interpretation. Graphical Model providing probabilistic interpretation while using linear model with no bias to coefficient:
  - Gaussian on w, $p(\mathbf{w}) \propto e^{-\lambda_2 ||\mathbf{w}||_2^2}$.
    - constraint on D is $||\mathbf{d}_j||_2^2 = 1 \text{ for all } j$
    - 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** ($f(x, \alpha, \theta) = x^T \mathbf{W} \alpha + b$)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 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. An 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

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

- 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 compose 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 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.

| **7. Support Vector Machine Active Learning for Image Retrieval** |
| --- |
| Simon Tong Proceedings of the ninth ACM international conference on Multimedia. ACM, 2001 |

- 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. Also, only by asking the user to label a few number of images.
- The paper proposes the use of a support vector machine active learning algorithm for doing such relevance feedback for image retrieval.
- This 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.

**$SVM_{active}$**

- 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 classier.
- Captures the query concept by separating the relevant images from the irrelevant images.
- $SVM_{Active}$ learns the classier quickly via active learning.
- 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.

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

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

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

**Active Learning**

- 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).
- 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.
- The active learner can return a classifier f after each pool-query (online learning) or after some fixed number of pool-queries.
- The main difference between an active learner and a regular passive learner is the querying component q.
- 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.
- The hyperplane that is closest to wi is b, so we will choose to query b.

**Active Learning Algorithm:**

- Our $SVM_{Active}$ system performs the following for each round of relevance feedback:
  - Learn an SVM on the current labeled data.
  - 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.

- After the relevance feedback rounds have been performed $SVM_{Active}$ retrieves the top-k most relevant images:
  - Learn a final SVM on the labeled data.
  - The final SVM boundary separates “relevant” images from irrelevant ones. Display the k “relevant” images that are farthest from the SVM boundary.

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 our 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.

| **8. Multinomial Naïve Bayes for Text Categorization Revisited** |
|---|
| Ashraf M. Kibriya, Eibe Frank, Bernhard Pfahringer, and Geoffrey Holmes |

- Comparing Standard Multinomial Naïve Bayes with Transformed Weight-Normalized Complement Naïve Bayes classifier (TWCNB).
- TWCNB is easy to implement, has good running time and claims to be as accurate as SVMs.

**Feature Generation:**

- Bag of words approach
- Documents represented as a set of words
- Reuters dataset treated different than others.
- TFIDF(word) = $log(f+1) \times log(\frac{D}{df})$

**Multinomial Naïve Bayes**

- It is used to compute class probabilities by using Bayes Rule: $\Pr(c|t_i) = \frac{\Pr(c)\Pr(t_i|c)}{\Pr(t_i)}, \qquad c \in C$
- Prior: Pr(*c*) = # of document of class C / Total # of documents
- Pr(*ti*/*c*) = $: (\sum_n f_{ni})! \prod_n \frac{\Pr(w_n|c)^{f_{ni}}}{f_{ni}!},$

**Transformed Weight-Normalized Complement Naive Bayes**

- TWCNB estimates the parameters of class c by using data from all classes apart from c and uses word weight rather than probability.

$$w_{nc} = log(\frac{1 + \sum_{k=1}^{|C|} F_{nk}}{N + \sum_{k=1}^{|C|} \sum_{x=1}^{N} F_{xk}}), \qquad k \neq c \wedge k \in C$$

- class(*ti*)

$$= \text{argmax}_c[\log(\Pr(c)) - \sum_n (f_{ni} w_{nc})],$$

$$= \text{argmin}_c[\sum_n (f_{ni} w_{nc})]$$

- class(*ti*)

**Evaluating Standard Multinomial Naïve Bayes**

- Comparing MNB with TWCNB, TCNB and linear SVMs
- In case of SVM, Sequential minimal optimization (SMO) algorithm is used.
- Results with full and reduced vocabulary are presented. Reuters dataset results reported as precision-recall break-even points.
- Improving Multinomial Naïve Bayes by Evaluating ways to increase the accuracy of MNB.
- Comparing the results when using MNB alone, with TFIDF (not normalized), normalized with respect to average vector length (TFIDFN*a)* and normalized to unit vector (TFIDFN).

**Locally Weighted Learning**

- Use only a subset of the training documents.
- Documents are selected based on their Euclidean distance to the test document.
- Select k (neighborhood size) nearest documents.
- Divide the distance by the kth nearest neighbor.
- *f*(*di*) = $= \begin{cases} 1 - d_i \text{ if } d_i <= 1 \\ 0 \quad \text{if } d_i > 1 \end{cases}$

- Experiments results are shown and Standard MNB can be improved substantially by applying TFIDF transformation and normalizing the feature vector to the average vector length.
- MNB with TFIDF outperforms TWCNB with TFIDF. Effect of weight normalization on complement Naïve Bayes is negligible.

- MNB with locally weighted learning is slightly better but not competitive with linear SVM.

| **9. 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 |

- The main objective of this paper is a novel distributed rank-1 dictionary learning (D-r1DL) model, leveraging the power of distributed computing for handling largescale fMRI big data.
- Compared to the gradient-based dictionary learning algorithms and the K-SVD, the proposed rank-1 dictionary learning algorithm has a few critical advantages.

**Advantages of the Proposed Algorithm**

- 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 source (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: 1. temporal pattern of the functional networks - regarded as basis activation patterns, and 2. spatial pattern of the functional networks.

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

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

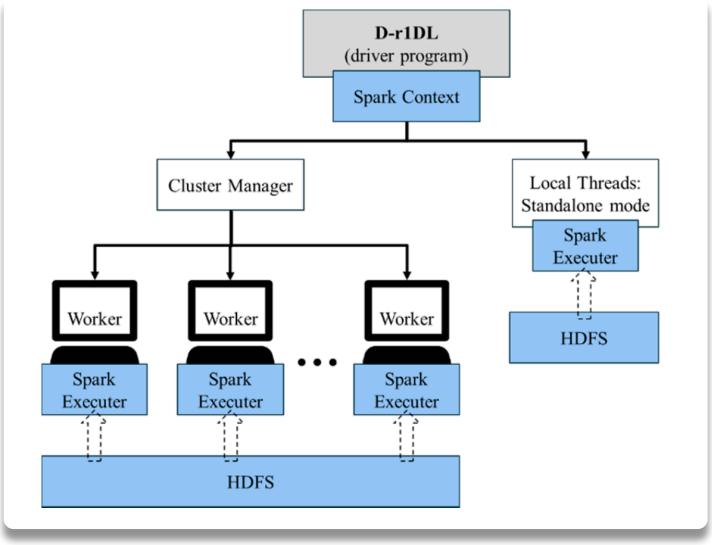
$$v = \underset{v}{\text{argmin}} \|S - uv^T\|_F, s.t. \|v\|_0 \le r,$$

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

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

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

Algorithm parallization and Deployment on SPARK




- 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.
- The Ultimate 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.

| **10. Overview and Recent Advances in Partial Least Squares** |
|---|
| Rosipal, Roman, and Nicole Krämer. "Overview and recent advances in partial least squares." Lecture notes in computer science 3940 (2006): 34. |

- Partial Least Squares (PLS) is a method for modeling relations between sets of observed variables by means of latent variables. Its general form creates orthogonal score vectors, commonly referred to as latent vectors by maximising the co-variance between different sets of variables.
- Latent Variables are those variables that are hidden and are not directly observed rather they are inferred using mathematical models on other variables.

- PLS is mainly used in fields like Chemometric, Bioinformatics, Food Research, Social Sciences etc.
- The scope of this paper is based on two blocks of variables. The paper gives an introduction to different variants of PLS.
- PLS is a general linear PLS algorithm models relation between two data sets X and Y where X⊆ $RN \ and \ Y \subseteq RM$ are the two blocks of variables.
- The relation is modeled by calculating score vectors between these two blocks.
- PLs decomposes n-samples from each data set and composes them in the form:
    - **$X = TP^T + E$** and **$Y = UQ^T + F$** where T, U are extracted score vectors P, Q are matrices of Loading E, F are matrices of Residuals

**PLS using NIPALS Algorithm**

- X and Y datasets are decomposed in two ways using NIPALS algorithm as:
- **First Method** is the Convergence Approach which is a Y-space score vector, u is randomly initialised and then follows the following steps of converges: $W = X^Tu/(u^Tu)$, $||w|| \rightarrow 1$, $t = Xw$, $c = Y^Tt/(t^Tt)$, $||c|| \rightarrow 1$, $u = Yc$ .
- **Second Way:** Eigen Value Problems: Weight vector, w corresponds to first eigenvector problem using equation: X and Y space score vectors t and u are given by:
  **t = Xw** and **u = Yc** Where the computation of t, u and c is given in steps 4 and 5 of NIPALS. The users can compute the values of these eigenvector problems and can readily compute relation between weight and score vectors.
- Different forms of PLS were discussed such as PLS-SB, SIMPLS etc.

**Projection Method: PCA vs. CCA vs. PLS**

- There are several methods of projection for latent variables apart from PLS namely:
    - Principal Component Analysis (PCA)
    - Canonical Correlation Analysis (CCA)
- In PCA, the original variables are projected onto a direction of maximal variance that is called principal direction.
- In CCA, the optimization criterion is based on finding maximum correlation.
- PLS optimization is a form of CCA where latent vectors are orthogonal directions that capture maximal co-variance in a single dataset.
- PLS Regression along with its Shrinkage Properties, PLS Discrimination and Classification, PLS Two-class Discrimination and Classification, PLS multi-class Discrimination and Classification were discussed.

**Non-Linear PLS: First Approach**

- This approach is based on reformulating the linear relation between score vectors t and u by as non-linear model as: $U = g(t) + h = g(X, w) + h$ where g(.) is a continuous function and h is a vector of residuals.

- This approach works on the assumption that score vectors t ,u are linear projections of the original variables.
- The function, g(.) is modeled using several methods namely: polynomial functions, smoothening splines, artificial neural networks and radial basis functions.
- This approach elaborates about the need for linearization of nonlinear mapping g(.) and successive iterative updating of weight vector, w.

**Non-Linear PLS: Second Approach**

- This is based on mapping of the original data by means of a nonlinear function to a new representation i.e., data space where linear PLS is applied and is based on kernel-based learning of non-linear PLS by modeling relation between observed variables, regression and classification problem.
- This approach maps original X-space data into high-dimensional feature space F. Kernel trick then applied to estimation of PLS and it reduces feature space, F into linear PLS.
- The kernel form is formed using NIPALS algorithm as:

$$1)\ \mathbf{t} = \mathbf{\Phi}\mathbf{\Phi}^T\mathbf{u} = \mathbf{K}\mathbf{u} \qquad 4)\ \mathbf{u} = \mathbf{Y}\mathbf{c}$$
$$2)\ \|\mathbf{t}\| \to 1 \qquad 5)\ \|\mathbf{u}\| \to 1$$
$$3)\ \mathbf{c} = \mathbf{Y}^T\mathbf{t}$$

- **Non-Linear PLS: Comparison** Kernel PLS approach is easily implementable, computationally less demanding and is capable of modeling difficult non-linear relations. However, there is a loss of interpretability of results with respect to original data.
- The first Approach focusses on keeping latent variables as linear projections of the original data which is inadequate in construction of some data situations.

The PLS method projects original data onto a more compact space of latent variables. This can leads to the deletion of unimportant variables so as the result we will have a significant cost reduction. PLS tools based on score and loadings plots allows to better understand data structure, observe existing relations among data sets and to detect outliers in the measured data.