ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

Privacy and Clustering Based Online Feature Selection with Public Auditing

K.Poornima*, C.Grace Padma

* Research Scholar in M.Phil Computer Science, RVS College of Arts & Science, Sulur, Coimbatore – 641 402, Tamil Nadu, India

Research Guide in M.Phil Computer Science, Associate Professor & HoD, Department of Computer Applications (MCA), RVS College of Arts & Science, Sulur, Coimbatore – 641 402, Tamil Nadu, India

Abstracts

Feature selection is essential topic in data mining. Although its importance, most studies of feature selection are limited to batch learning. The online feature selection is used to make accurate prediction for using small number and fixed number of active features . We deal with this challenge by studying scarcity regularization and truncation techniques. we estimate the performance of proposed algorithms for online feature selection and its applications .we propose Two-Gaussian algorithm for clustering a search result. Our predicted informations are separately grouped in the basis of classification. This clustering technique done by using Two Gaussian mixtures algorithm. And we implement Blowfish algorithm for reduce privacy issues. All informations are stored in our system as an encrypted format. And also we implement public auditing for audit a user contents. Because it is used to avoid fake informations.

Keywords: Feature Selection & Prediction, Clustering, Privacy.

Introduction

Feature selection (FS) is an important topic in data mining and machine learning, and has been extensively studied for many years in literature. [3], [4], [5], [6], [9]. For classification, the objective of feature selection is to select a subset of relevant features for building effective prediction models. By removing irrelevant and redundant features, feature selection can improve the performance of prediction models by alleviating the effect of the curse of dimensionality, enhancing the generalization performance, speeding up the learning process, and improving the model interpretability. Feature selection has found applications in many domains, especially for the problems involved high dimensional data. Despite being studied extensively, most existing studies of feature selection are restricted to batch learning, which assumes that the feature selection task is conducted in an offline batch.

The learning fashion and all the features of training instances are given a priori. Such assumptions may not always hold for real-world applications in which training examples arrive in a sequential manner or it is expensive to collect the full information of training data.

For example, in an online spam email detection system, training data usually arrive sequentially, making it difficult to deploy a regular batch feature selection technique in a timely, efficient, and scalable manner. Another example of feature selection is in bioinformatics, where acquiring the entire set of features/ attributes for every training instance is expensive due to the high cost in conducting wet lab experiments.[15]

This paper ,we address four different types of online feature selection tasks: 1) OFS by learning with full inputs, 2) OFS by learning with partial inputs. 3) Clustering in the basis of Classification.4) Each Information's are verified by auditor. For the first task, we assume that the learner can access all the features of training instances, and our goal is to efficiently identify a fixed number of relevant features for accurate prediction. In the second task, we consider a more challenging scenario where the learner is allowed to access a fixed small number of features for each training instance to identify the subset of relevant features. To make this problem attractable, we allow the learner to decide which subset of features to acquire for each training instance. In the third task, retracing information are creates by a group, i.e. clustering. The clustering technique is done by using classification. In fourth task, user uploads articles or information are verified by auditor, because it used to avoid fake information's.[1],[2],[10].

Literature survey

Distributional Word Clusters vs. Words for Text Categorization

Descriptions: We study an approach to text categorization that combines distributional clustering of words and a Support Vector Machine (SVM) classifier. This word-cluster representation is computed using the recently introduced Information Bottleneck method, which generates a compact and efficient representation of documents. When combined with the classification power of the SVM, this method yields high performance in text categorization. This novel combination of SVM with word-cluster representation is compared with SVM-based categorization using the simpler bag-of-words (BOW) representation. The comparison is performed over three known datasets. On one of these datasets (the 20 Newsgroups) the method based on word clusters significantly outperforms the word-based representation in terms of categorization accuracy or representation efficiency. On the two other sets (Reuters-21578 and WebKB) the word-based representation slightly outperforms the word-cluster representation. We investigate the potential reasons for this behavior and relate it to structural differences between the data sets. Author: Ron Bekkerman, Ran El-Yaniv, Naftali Tishby & Yoad Winter.[7]

Dimensionality Reduction via Sparse Support Vector Machines

Descriptions: We describe a methodology for performing variable ranking and selection using support vector machines (SVMs). The method constructs a series of sparse linear SVMs to generate linear models that can generalize well, and uses a subset of non zero weighted variables found by the linear models to produce a final nonlinear model. The method exploits the fact that a linear SVM (no kernels) with `1-norm regularization inherently performs variable selection as a side-effect of minimizing capacity of the SVM model. The distribution of the linear model weights provides a mechanism for ranking and interpreting the effects of variables. Star plots are used to visualize the magnitude and variance of the weights for each variable. We illustrate the effectiveness of the methodology on synthetic data, benchmark problems, and challenging regression problems in drug design. This method can dramatically reduce the number of variables and outperforms SVMs trained using all attributes and using the attributes selected according to correlation coefficients. The visualization of the resulting models is useful for understanding the role of underlying variables.
Author: Jinbo Bi, Kristin P. Bennett, Mark Embrechts, Curt M. Breneman & Minghu Song

The Projectron: a Bounded Kernel-Based Perceptron

Descriptions: We present a discriminative online algorithm with a bounded memory growth, which is based on the kernel-based Perceptron. Generally, the required memory of the kernel based Perceptron for storing the online hypothesis is not bounded. Previous work has been focused on discarding part of the instances in order to keep the memory bounded. In the proposed algorithm the instances are not discarded, but projected onto the space spanned by the previous online hypothesis. We derive a relative mistake bound and compare our algorithm both analytically and empirically to the state-of-the-art Forgetron algorithm (Dekel et al, 2007). The first variant of our algorithm, called Projectron, outperforms the Forgetron. The second variant, called Projectron++, outperforms even the Perceptron. Francesco Orabona, Joseph Keshet & Barbara Caputo Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy Feature selection is an important problem for pattern classification systems. We study how to select good features according to the maximal statistical dependency criterion based on mutual information. Because of the difficulty in directly implementing the maximal dependency condition, we first derive an equivalent form, called minimal-redundancy-maximal-relevance criterion (mRMR), for first-order incremental feature selection. Then, we present a two-stage feature selection algorithm by combining mRMR and other more sophisticated feature selectors (e.g., wrappers). This allows us to select a compact set of superior features at very low cost. We perform extensive experimental comparison of our algorithm and other methods using three different classifiers (naive Bayes, support vector machine, and linear discriminate analysis) and four different data sets (handwritten digits, arrhythmia, NCI cancer cell lines, and lymphoma tissues). The results confirm that mRMR leads to promising improvement on feature selection and classification accuracy.
Author: Hanchuan Peng, Fuhui Long, and Chris Ding.[11]

Online Feature Selection using Grafting
Descriptions: In the standard feature selection problem, we are given a fixed set of candidate features for use in a learning problem, and must select a subset that will be used to train a model that is “as good as possible” according to some criterion. In this paper, we present an interesting and useful variant, the online feature selection problem, in which, instead of all features being available from the start, features arrive one at a time. The learner’s task is to select a subset of features and return a corresponding model at each time step which is as good as possible given the features seen so far. We argue

that existing feature selection methods do not perform well in this scenario, and describe a promising alternative method, based on a stage wise gradient descent technique which we call grafting.[15]
Author: Simon Perkins & James Theiler

Forward Semi-Supervised Feature Selection
Descriptions: Traditionally, feature selection methods work directly on labeled examples. However, the availability of labeled examples cannot be taken for granted for many real world applications, such as medical diagnosis, forensic science, fraud detection, etc, where labeled examples are hard to find. This practical problem calls the need for "semi-supervised feature selection" to choose the optimal set of features given both labeled and unlabeled examples that return the most accurate classifier for a learning algorithm. In this paper, we introduce a "wrapper-type" forward semi-supervised feature selection framework. In essence, it uses unlabeled examples to extend the initial labeled training set. Extensive experiments on publicly available datasets shows that our proposed framework, generally, outperforms both traditional supervised and state of-the-art "filter-type" semi-supervised feature selection algorithmsby 1% to 10% in accuracy.
Author: Jiangtao Ren, Zhengyuan Qiu, Wei Fan, Hong Cheng3 and Philip S. Yu.[13]

Learning with Missing Features
Descriptions: We introduce new online and batch algorithms that are robust to data with missing features, a situation that arises in many practical applications. In the online setup, we allow for the comparison hypothesis to change as a function of the subset of features that is observed on any given round, extending the standard setting where the comparison hypothesis is fixed throughout. In the batch setup, we present a convex relaxation of a non-convex problem to jointly estimate an imputation function, used to fill in the values of missing features, along with the classification hypothesis. We prove regret bounds in the online setting and Rademacher complexity bounds for the batch i.i.d. setting. The algorithms are tested on several UCI datasets, showing superior performance over baseline imputation methods.
Author: Afshin Rostamizadeh, Alekh Agarwal & Peter Bartlett.[14]

A review of feature selection techniques in bioinformatics
Descriptions: Feature selection techniques have become an apparent need in many bioinformatics applications. In addition to the large pool of techniques that have already been developed in the machine learning and data mining fields, specific applications in bioinformatics have led to a wealth of newly proposed techniques. In this paper, we make the interested reader aware of the possibilities of feature selection, providing a basic taxonomy of feature selection techniques, and discussing their use, variety and potential in a number of both common as well as upcoming bioinformatics applications.[12]
Author: Yvan Saeys, Inaki Inza and Pedro Larranaga

Randomized Online PCA Algorithms with Regret Bounds that are Logarithmic in the Dimension
Descriptions: We design an online algorithm for Principal Component Analysis. In each trial the current instance is centered and projected into a probabilistically chosen low dimensional subspace. The regret of our online algorithm, that is, the total expected quadratic compression loss of the online algorithm minus the total quadratic compression loss of the batch algorithm, is bounded by a term whose dependence on the dimension of the instances is only logarithmic. We first develop our methodology in the expert setting of online learning by giving an algorithm for learning as well as the best subset of experts of a certain size. This algorithm is then lifted to the matrix setting where the subsets of experts correspond to subspaces.
Author: Manfred K. Warmuth & Dima Kuzmin

Online feature selection

Feature Selection & Prediction:

In our system, we address two different types of online feature selection tasks:

1. OFS by learning with full inputs, and
2. OFS by learning with partial inputs.

1) For the first task, we assume that the learner can access all the features of training instances, and our goal is to efficiently identify a fixed number of relevant features for accurate prediction. In this task, we assume the learner is provided with full inputs of every training instances. To motivate our algorithm, we first present a simple but non effective algorithm that simply truncates the features with small weights. The failure of this simple algorithm motivates us to develop effective algorithms for OFS.

2) In the second task, we consider a more challenging scenario where the learner is allowed to access a fixed small number of features for each training instance to identify the subset of relevant features. To make this problem attractable, we allow the learner to decide which subset of features to acquire for each training instance.

Problem found in existing system

In our existing system is not always appropriate for real-world applications when data instances are of high dimensionality and it is more expensive for retrieving fullest of information's. It contain only retrieving technique. So It is not proper for all applications like online feature selection. In this system allows Fake information's also.

Most existing studies of online learning require accessing all the attributes and features of training instances. Such a classical setting is not always appropriate for real-world applications when data instances are of high dimensionality or it is expensive to acquire the full set of attributes and features. To address this limitation, we investigate the problem of online feature selection in which an online learner is only allowed to maintain a classifier involved only a small and fixed number of features. The main objective of our system is to create a best online feature selection method, clustering and provide a high security for data's.

Proposed system

In our proposed system is focusing four tasks in online feature selection. These are mentioned given below.

- Prediction
- Classification & Clustering
- Security
- Auditing

Example Pseudo Code:

```
### gaussian_example.R ###
# usage: R --no-save < gaussian_example.R

library(mclust)                 # load mclust library
x1 = rnorm(n=20, mean=1, sd=1)  # get 20 normal distributed points for x axis with mean=1 and std=1 (1st class)
y1 = rnorm(n=20, mean=1, sd=1)  # get 20 normal distributed points for x axis with mean=1 and std=1 (2nd class)
x2 = rnorm(n=20, mean=5, sd=1)  # get 20 normal distributed points for x axis with mean=5 and std=1 (1st class)
y2 = rnorm(n=20, mean=5, sd=1)  # get 20 normal distributed points for x axis with mean=5 and std=1 (2nd class)
rx = range(x1,x2)               # get the axis x range
ry = range(y1,y2)               # get the axis y range
plot(x1, y1, xlim=rx, ylim=ry)  # plot the first class points
points(x2, y2)                  # plot the second class points
mix = matrix(nrow=40, ncol=2)   # create a dataframe matrix
mix[,1] = c(x1, x2)             # insert first class points into the matrix
mix[,2] = c(y1, y2)             # insert second class points into the matrix
mixclust = Mclust(mix)          # initialize EM with hierarchical clustering, execute BIC and EM
plot(mixclust, data = mix)      # plot the two distinct clusters found
```

The two well separated data sets generated through a gaussian distribution function (Normal). The points are showed in the first chart.

Advantages:

1. We propose **Two** Gaussian mixtures algorithm for clustering technique.
2. We Propose Blowfish algorithm for more security.
3. We validate their empirical performance by conducting an extensive set of experiments;
4. Finally, we apply our technique to solve real-world problems in text classification, computer vision, and bioinformatics.

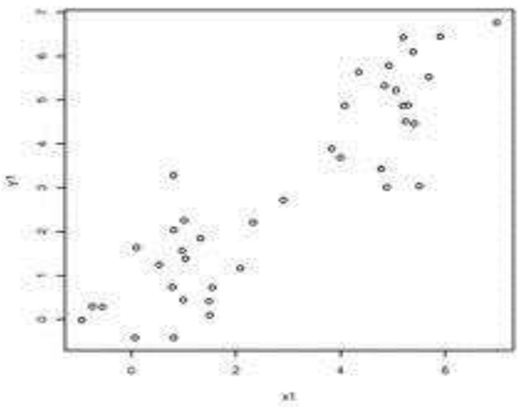
Algorithm details

Online feature selection is addresses two different tasks of online feature selection: 1) learning with full input, where an learner is allowed to access all the features to decide the subset of active features, and 2) learning with partial input, where only a limited number of features is allowed to be accessed for each instance by the learner.[8] 3) We present TwoGaussian technique for clustering a search result, 4) Blowfish algorithm for data privacy and also we implement public auditing for audit a user contents.

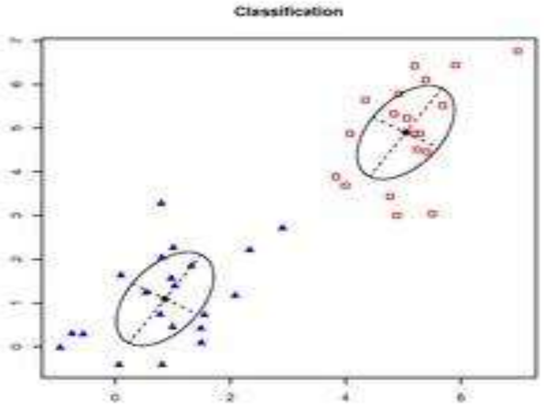
Two Gaussian mixtures:

This scenario is composed by two well separated data sets generated through a gaussian distribution function (Normal). As we can see, the EM clustering obtain two gaussian models that is in conformed to the data.[14]

(SEPARATED DATA SETS)



The EM clustering is applied and the results are also showed in the graphs below. As we can see, the EM clustering obtain two gaussian models that is in conformed to the data.



(CLASSIFICATION)

Blowfish Algorithm

The data transformation process for Pocket Brief uses the Blowfish Algorithm for Encryption and Decryption, respectively. The details and working of the algorithm are given below.

- Blowfish is a symmetric block cipher that can be effectively used for encryption and safeguarding of data. It takes a variable-length key, from 32 bits to 448 bits, making it ideal for securing data. Blowfish was designed in 1993 by Bruce Schneier as a fast, free alternative to existing encryption algorithms. Blowfish is unpatented and license-free, and is available free for all uses.
- Blowfish Algorithm is a Feistel Network, iterating a simple encryption function 16 times. The block size is 64 bits, and the key can be any length up to 448 bits. Although there is a complex initialization phase required before any encryption can take place, the actual encryption of data is very efficient on large microprocessors.
- Blowfish is a variable-length key block cipher. It is suitable for applications where the key does not change often, like a communications link or an automatic file encryptor. It is significantly faster than most encryption algorithms when implemented on 32-bit microprocessors with large data caches.

Description of the algorithm:

Blowfish is a variable-length key, 64-bit block cipher. The algorithm consists of two parts: a key-expansion part and a data- encryption part. Key expansion converts a key of at most 448 bits into several sub key arrays totaling 4168 bytes. Data encryption occurs via a 16-round Feistel network. Each round consists of a key dependent permutation, and a key- and data-dependent substitution. All operations are XORs and additions on 32-bit words. The only additional operations are four indexed array data lookups per round.

Sub keys:

Blowfish uses a large number of sub keys. These keys must be pre-computed before any data encryption or decryption.

- The P-array consists of 18 32-bit sub keys:
  P1, P2,..., P18.
- There are four 32-bit S-boxes with 256 entries each:
  S1,0, S1,1,..., S1,255;
  S2,0, S2,1,..,, S2,255;
  S3,0, S3,1,..., S3,255;
  S4,0, S4,1,..,, S4,255.

Encryption:

Blowfish has 16 rounds.

The input is a 64-bit data element, x.
Divide x into two 32-bit halves: xL, xR.
Then, for i = 1 to 16:
xL = xL XOR Pi
xR = F(xL) XOR xR
Swap xL and xR

After the sixteenth round, swap xL and xR again to undo the last swap.

Then, xR = xR XOR P17 and xL = xL XOR P18.

Finally, recombine xL and xR to get the ciphertext.

Decryption is exactly the same as encryption, except that P1, P2,..., P18 are used in the reverse order. Implementations of Blowfish that require the fastest speeds should unroll the loop and ensure that all subkeys are stored in cache.

Generating the Sub keys:

The sub keys are calculated using the Blowfish algorithm:

1. Initialize first the P-array and then the four S-boxes, in order, with a fixed string. This string consists of the hexadecimal digits of pi (less the initial 3): P1 = 0x243f6a88, P2 = 0x85a308d3, P3 = 0x13198a2e, P4 = 0x03707344, etc.

2. XOR P1 with the first 32 bits of the key, XOR P2 with the second 32-bits of the key, and so on for all bits of the key (possibly up to P14). Repeatedly cycle through the key bits until the entire P-array has been XORed with key bits. (For every short key, there is at least one equivalent longer key; for example, if A is a 64-bit key, then AA, AAA, etc., are equivalent keys.)
3. Encrypt the all-zero string with the Blowfish algorithm, using the sub keys described in steps (1) and (2).
4. Replace P1 and P2 with the output of step (3).
5. Encrypt the output of step (3) using the Blowfish algorithm with the modified sub keys.
6. Replace P3 and P4 with the output of step (5).
7. Continue the process, replacing all entries of the P array, and then all four S-boxes in order, with the output of the continuously changing Blowfish algorithm.

In total, 521 iterations are required to generate all required sub keys. Applications can store the sub keys rather than execute this derivation process multiple times.

Design decisions:

- A 64-bit block size yields a 32-bit word size, and maintains block-size compatibility with existing algorithms. Blowfish is easy to scale up to a 128-bit block, and down to smaller block sizes.
- The fundamental operations were chosen with speed in mind. XOR, ADD, and MOV from a cache are efficient on both Intel and Motorola architectures. All sub keys fit in the cache of a 80486, 68040, Pentium, and PowerPC.
- The Feistel Network that makes up the body of Blowfish is designed to be as simple as possible, while still retaining the desirable cryptographic properties of the structure.
- Our algorithm design, there are two basic ways to ensure that the key is long enough to ensure a particular security level. One is to carefully design the algorithm so that the entire entropy of the key is preserved, so there is no better way to crypt analyzes the algorithm other than brute force. The other is to design the algorithm with so many key bits that attacks that reduce the effective key length by several bits are irrelevant. Since Blowfish is designed for large microprocessors with large amounts of memory, the latter has been chosen. But it works equally well on Handheld systems with a decent microprocessor.
- The sub key generation process is designed to preserve the entire entropy of the key and to distribute that entropy uniformly throughout the sub keys. It is also designed to distribute the set of allowed sub keys randomly throughout the domain of possible sub keys. The digits of pi were chosen as the initial sub key table for two reasons: because it is a random sequence not related to the algorithm, and because it could either be stored as part of the algorithm or derived when needed. But if the initial string is non-random in any way (for example, ASCII text with the high bit of every byte a 0), this non-randomness will propagate throughout the algorithm.
- In the sub key generation process, the sub keys change slightly with every pair of sub keys generated. This is primarily to protect against any attacked of the sub key generation process that exploit the fixed and known sub keys. It also reduces storage requirements. The 448 limit on the key size ensures that the every bit of every sub key depends on every bit of the key.
- The key bits are repeatedly XORed with the digits of pi in the initial P-array to prevent the following potential attack: Assume that the key bits are not repeated, but instead padded with zeros to extend it to the length of the P-array. An attacker might find two keys that differ only in the 64-bit value XORed with P1 and P2 that, using the initial known sub keys, produce the same encrypted value. If so, he can find two keys that produce all the same sub keys. This is a highly tempting attack for a malicious key generator. To prevent this same type of attack, the initial plaintext value in the sub key generation process is fixed.

- The sub key-generation algorithm does not assume that the key bits are random. Even highly correlated key bits, such as an alphanumeric ASCII string with the bit of every byte set to 0, will produce random sub keys. However, to produce sub keys with the same entropy, a longer alphanumeric key is required.
- The time-consuming sub key-generation process adds considerable complexity for a brute-force attack. The sub keys are too long to be stored on a massive tape, so they would have to be generated by a brute-force cracking machine as required. A total of 522 iterations of the encryption algorithm are required to test a single key, effectively adding 29 steps to any brute-force attack.
- The most efficient way to break Blowfish is through exhaustive search of the key space.

Results

Authentication:

- Authentication is the act of confirming the truth of an attribute of a datum or entity. This might involve confirming the identity of a person or software program, tracing the origins of an artifact, or ensuring that a product is what it's packaging and labeling claims to be. Authentication often involves verifying the validity of at least one form of identification. We authenticate our system by using username and password. The username-password authentication flow can be used to authenticate when the consumer already has the user's credentials.

Feature Selection & Prediction:

- In our system, we address three different types of online feature selection tasks:
  1) OFS by learning with full inputs, and
  2) OFS by learning with partial inputs.
- For the first task, we assume that the learner can access all the features of training instances, and our goal is to efficiently identify a fixed number of relevant features for accurate prediction.
- In the second task, we consider a more challenging scenario where the learner is allowed to access a fixed small number of features for each training instance to identify the subset of relevant features. To make this problem attractable, we allow the learner to decide which subset of features to acquire for each training instance.[13]

Clustering:

- When our predicted information's are separately grouped in the basis of classification.
- This clustering technique done by using **Two Gaussian mixtures algorithm.**

Audition:

- When uses upload their article or educational oriented texts into our system mean, the auditor will audit their information's, if those information's are current in a sense, auditor will approve their information's in our system.

Privacy:

- In our system, we use **Blowfish algorithm** for reduce privacy issues.
- That is, all information's are stored in our system as an encrypted format.

Conclusion

Conclusion of this phase provides we research a problem in online features selection and how to rectify it. This system we implement two algorithms for prediction, clustering and privacy implementation. First one is Two-Gaussian technique for OFS (online Feature Selection) implementation, implementing classification technique and clustering retrieved information's. Second one is Blowfish algorithm for rectifying privacy issues. The data transformation process for Pocket Brief uses the Blowfish Algorithm for Encryption and Decryption, respectively. Blowfish is a symmetric block cipher that can be effectively used for encryption and safeguarding of data. Both two algorithms are used to improve a performance of online Feature Selection and provide a high security implementation for informations.

Future work

Our Future work could extend our frame work to other settings. we introduce a system generated auditing technique for auditing informations. This could provide better utility because of reduce a time delay and man power which indicate superior performance of online feature selection.

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

Reference

1. David J. Sharp , Fani Deligianni, Gaël Varoquaux, Bertrand Thirion, , Christian Ledig, Robert Leech, and Daniel Rueckert.”A Framework for Inter-Subject Prediction of Functional Connectivity from Structural Networks.
2. C. Gentile,“A New Approximate Maximal Margin Classification Algorithm,”Machine Learning Research, vol. 2, pp. 213-242, 2001.
3. I. Guyon and A. Elisseef,J ,“An Introduction to Variable and Feature Selection,”. Machine Learning Research, vol. 3, pp. 1157-1182, 2003.
4. Y. Saeys, I. Inza, and P. Larranaga, “A Review of Feature Selection Techniques in Bioinformatics,” Bioinformatics, vol. 23, no. 19, pp. 2507-2517, 2007
5. Jinbo Bi, Kristin P. Bennett, Mark Embrechts, Curt M. Breneman & Minghu Song,”Dimensionality Reduction via Sparse Support Vector Machines”
6. Z. Xu, I. King, M.R. Lyu, and R. Jin, “Discriminative Semi-Supervised Feature Selection via Manifold Regularization,” IEEE Trans. Neural Networks, vol. 21, no. 7, pp. 1033-1047, July 2010
7. Ron Bekkerman, Ran El-Yaniv, Naftali Tishby & Yoad Winter.“Distributional Word Clusters vs. Words for Text Categorization.
8. P. Zhao and S.C.H. Hoi, “Bduol: “Double Updating Online Learning” on a Fixed Budget,” Proc. European Conf. Machine Learning and Knowledge Discovery in Databases (ECML/PKDD ’12),no. 1, pp. 810-826, 2012.
9. J. Duchi and Y. Singer, “Efficient Online and Batch Learning Using Forward Backward Splitting, J. Machine Learning Research, vol. 10, pp. 2899-2934, 2009.
10. S. Yang, L. Yuan, Y.-C. Lai, X. Shen, P. Wonka, and J. Ye, “Feature Grouping and Selection over an Undirected Graph, Proc. 18th ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining(KDD ’12), pp. 922-930, 2012
11. H. Peng, F. Long, and C.H.Q. Ding, “Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy,” IEEE Trans. Pattern Analysis Machine Intelligence, vol. 27, no. 8, pp. 1226-1238, Aug. 2005.
12. M. Dash and H. LiuIntelligent Data Analysis, “Feature Selection for Classification,” vol. 1, nos. 1-4, pp. 131-156, 1997.
13. Jiangtao Ren, Zhengyuan Qiu, Wei Fan, Hong Cheng3 and Philip S. Yu. ”Forward Semi-Supervised Feature Selection”.
14. A. Rostamizadeh, A. Agarwal, and P.L. Bartlett, “Learning with Missing Features,” Proc. Conf. Uncertainty in Artificial Intelligence(UAI ’11), pp. 635-642, 2011.
15. S. Perkins and J. Theiler, “Online Feature Selection Using Grafting,” Int’l Conf. Machine Learning (ICML ’03), pp. 592-599,2003.
16. http://www.google.com
17. http://asp.net-tutorials.com/