TEXT CATEGORIZATION USING Q-LEARNING ALOGRITHM

Dr.S.R.Suresh[1], T.Karthikeyan[2], D.B.Shanmugam[3],J.Dhilipan[4]

1-Principal, Arul College of Technology, Radhapuram, Email-id: sur1971@gmail.com

2-Assitant Professor,Dept of CSE, Sri Balaji Chockalingam Engg. College, tkns123@rediff.com

3-Assitant Professor, Dept of MCA, Sri Balaji Chockalingam Engg. College, dbshanmugam@yahoo.co.in

4-Assitant Professor, Dept of MCA, SRM University ,Chennai , jd_pan@yahoo.co.in

ABSTRACT

This paper aims at creation of an efficient document classification process using reinforcement learning, a branch of machine learning that concerns itself with optimal sequential decision-making. One strength of reinforcement learning is that it provides formalism for measuring the utility of actions that gives benefit only in the future. An effective and flexible classifier learning algorithm is provided, which classifies a set of text documents into a more specific domain like Cricket, Tennis and Football. This novel approach has been evaluated, with standard information retrieval techniques.

Recent work in reinforcement learning it has been proved that a quantitative connection between the expected some of rewards of a policy and binary classification performance on a created sub problem. Without any unobservable assumption, the resulting statement is independent of the number of states or actions.

Keywords- Reinforcement learning (RL), naïve bayes, a-priori, spidering,

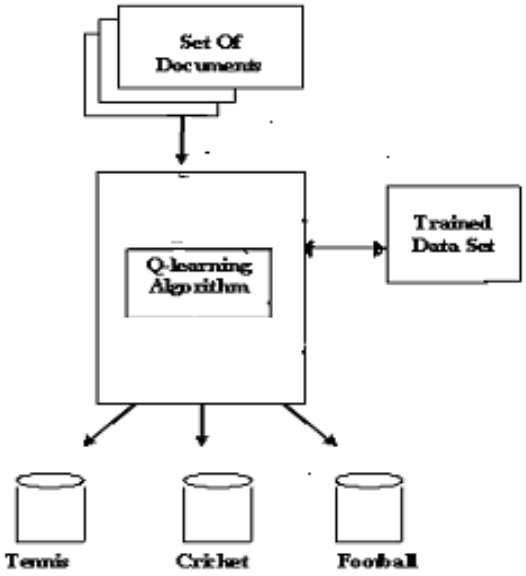
INTRODUCTION

As the amount of information on the World Wide Web grows, it becomes difficult to find pages of a particular kind or on a particular topic. The general-purpose search engines use breadth-first search to find as many web pages as possible for a given detailed query that lead to off-topic areas. In order to find topic directed web documents, reinforcement learning techniques are used classifying efficiently the text documents. This paper presents our work in an efficient domain specific text document classification. We argue that the creation of efficient domain specific classifier is best framed and solved by reinforcement learning, a branch of machine learning that concerns itself with optimal sequential decision making. The strength of reinforcement learning is that it provides formalism for measuring the utility of actions that gives benefit only in the future. Reinforcement learning agents represent this delayed benefit by learning a mapping from each available action to a scalar value indicating the sum of discounted rewards expected from executing the action. The “discount” makes later rewards less valuable than immediate rewards, thus encouraging efficiency. In our current work with document classification, we present an algorithm for learning a value function that maps text to rewards using a naïve bayes text classifier. Q-learning technique then uses these rewards to increases the efficiency of classifier by maximizing the expected sum of rewards.

Over all model

The overall model of the project shown in the block diagram depicts the steps involved in implementation of the project:

- Collect all relevant documents related to different domain (cricket, football, tennis).
- Create a training data set using documents of different domain.



Block Diagram for Document Classification

- Structure the new document in order to identify the keywords.
- Apply Temporal Difference Learning technique to the new document, to calculate Q-Value of the document.
    - During learning process, naïve bayes classifier maps text to Q- value by calculating the maximum probability that the new document would belong to the particular domain.
    - With the Q- values of different domains, Q-Learning algorithm (off policy temporal difference learning technique) with a document as state ‘s’ and moving from one document to another document as action ‘a’ finds an optimal Q-value of the document.
    - New document which has the optimal Q-value for a domain is displayed with the domain tag.

EXISTING SYSTEM

As the amount of information on the World Wide Web grows, it becomes difficult to find pages of a particular kind or on a particular topic. The general-purpose search engines use breadth-first search to find as many web pages as possible for a given detailed query that lead to off-topic areas. In order to find topic directed web documents, reinforcement learning techniques are used classifying efficiently the text documents.

The existing system in document classification using reinforcement learning technique.

- In creation of efficient web spider, reinforcement learning technique is used to explore the hyperlinks of the web graph for finding topic related web documents. This efficient spidering

task by reinforcement learning is also employed to automate the creation and maintenance of domain specific search engine.

- Domain specific search engines searches the World Wide Web with increased accuracy and extra features that is not possible in other general search engines.
- In another work Web Watcher, reinforcement learning technique is used to develop a system that accompanies, as the user browses the web. Web watcher learns to become an expert for the parts of the web it visited in the past for the type and topic of user's interest.

Recent work in reinforcement learning it has been proved that a quantitative connection between the expected some of rewards of a policy and binary classification performance on a created sub problem. Without any unobservable assumption, the resulting statement is independent of the number of states or actions.

PROPOSED SYSTEM

This paper presents our work in an efficient domain specific text document classification. We argue that the creation of efficient domain specific classifier is best framed and solved by reinforcement learning, a branch of machine learning that concerns itself with optimal sequential decision making. The strength of reinforcement learning is that it provides formalism for measuring the utility of actions that gives benefit only in the future. Reinforcement learning agents represent this delayed benefit by learning a mapping from each available action to a scalar value indicating the sum of discounted rewards expected from executing the action. The "discount" makes later rewards less valuable than immediate rewards, thus encouraging efficiency.

In our current work with document classification, we present an algorithm for learning a value function that maps text to rewards using a naïve bayes text classifier. Q-learning technique then uses these rewards to increases the efficiency of classifier by maximizing the expected sum of rewards. Application of reinforcement learning to classification problem involves encountering large state and action space. It encapsulates not only on-topic documents remaining to be found, but also the set of other documents that are available as actions.

IMPLEMENTATION WORK

Q-learning is a reinforcement learning technique that works by learning an action-value function that gives the expected utility of taking a given action in a given state and following a fixed policy thereafter. One of the strengths of Q-learning is that it is able to compare the expected utility of the available actions without requiring a model of the environment. A recent variation called delayed Q-learning has shown substantial improvements, bringing Probably approximately correct learning (PAC) bounds to Markov decision processes .

Algorithm

The problem model consists of an agent, states S and a set of actions per state A. By performing an action $a \in A$, the agent can move from state to state. Each state provides the agent a reward (a real or natural number). The goal of the agent is to maximize its total reward. It does this by learning which action is optimal for each state.

The algorithm therefore has a function which calculates the Quality of a state-action combination:

$$Q : S \times A \rightarrow \mathbb{R}$$

Before learning has started, *Q* returns a fixed value, chosen by the designer. Then, each time the agent is given a reward (the state has changed) new values are calculated for each combination of a state *s* from *S*, and action *a* from *A*. The core of the algorithm is a simple value iteration update. It assumes the old value and makes a correction based on the new information.

$$Q(s_t, a_t) \leftarrow \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha_t(s_t, a_t)}_{\text{learning rate}} \times \left[\overbrace{\underbrace{R(s_{t+1})}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \underbrace{\max_a Q(s_{t+1}, a)}_{\text{max future value}}}^{\text{expected discounted reward}} - \overbrace{Q(s_t, a_t)}^{\text{old value}} \right]$$

Where $R(s_{t+1})$ is the reward observed from s_{t+1}, $\alpha_t(s,a)$ ($0 < \alpha \leq 1$) the learning rate (may be the same for all pairs). The discount factor γ is such that $0 \leq \gamma < 1$

The above formula is equivalent to:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t)(1 - \alpha_t(s_t, a_t)) + \alpha_t(s_t, a_t)[R(s_{t+1}) + \gamma \max_a Q(s_{t+1}, a)]$$

An episode of the algorithm ends when state s_{t+1} is a final state (or, "absorbing state").

Note that for all final states s_f, $Q(s_f,a)$ is never updated and thus retains its initial value.

Learning

What has attracted the most interest in neural networks is the possibility of *learning*. Given a specific *task* to solve, and a *class* of functions, F, learning means using a set of *observations* to find $f^* \in F$ which solves the task in some *optimal* sense. This entails defining a cost function $C: F \to \mathbb{R}$ such that, for the optimal solution f^*, $C(f^*) \leq C(f) \forall f \in F$ (i.e., no solution has a cost less than the cost of the optimal solution). The cost function C is an important concept in learning, as it is a measure of how far away a particular solution is from an optimal solution to the problem to be solved. Learning algorithms search through the solution space to find a function that has the smallest possible cost. For applications where the solution is dependent on some data, the cost must necessarily be a *function of the observations*, otherwise we would not be modelling anything related to the data. It is frequently defined as a statistic to which only approximations can be made. As a simple example, consider the problem of finding the model f, which minimizes $C = E[(f(x) - y)^2]$, for data pairs (x, y) drawn from some distribution D. In practical situations we would only have N samples from D and thus, for the above example, we would only minimize $\hat{C} = \frac{1}{N} \sum_{i=1}^{N} (f(x_i) - y_i)^2$. Thus, the cost is minimized over a sample of the data rather than the entire data set. When $N \to \infty$ some form of online machine learning must be used, where the cost is partially minimized as each new example is seen. While online machine learning is often used when D is fixed, it is most useful in the case where the distribution changes slowly over time. In neural network methods, some form of online machine learning is frequently used for finite datasets.

Reinforcement learning

This module Reinforcement learning (RL) refers to a framework for learning optimal decision making from rewards. It differs from other computational approach in that the learner is never told the correct action for a particular state, but only told how well or bad the selected action was, expressed in the form of scalar reward.

A task in RL is defined by set of states s ε S, a set of action a ε A, state-action transition function T: S x A -> S and reward function R: S X A-> Ŕ. At each time step the learner selects an action and then as a result is given a reward and its new state. The goal of reinforcement learning is to learn a policy Π, mapping from state to actions, Π: S → A that maximizes the sum of reward over time. For policy Π, we can define the value of each state to be:

$$V^{\Pi}(s) = \sum_{t=0}^{\infty} \gamma^t r_t \qquad (1)$$

r_t → Reward received in t time step

γ^t →Discount factor $0 \leq \gamma < 1$.

The optimal policy, Π* is the one that maximizes the value, $V^{\Pi}(s)$, for all states s.

In order to learn the optimal policy, we learn its value function, V*, and its more correlate, called Q. Let Q*(s,a) be the value of selecting action a from state s, and thereafter following the optimal policy is expressed as:

$$Q^*(s,a) = R(s,a) + \gamma V^*(T(s,a)) \qquad (2)$$

The optimal policy in terms of Q is defined by selecting from each state the action with highest expected future reward:

$$\Pi^*(s) = \arg\max_a Q^*(s,a) \qquad (3)$$

Reinforcement Learning To Classification

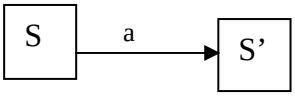
This module is in document classification task, the domain specific documents are immediate rewards. An action is moving from a document to another document. The state is the set of documents to be found and set of documents that have been classified. Reinforcement learning on real-world problem requires large state and action space. However, reinforcement learning algorithms like Q-Learning algorithm is tractable to small environment with relatively few states and actions.

Q-Learning Algorithm

This module of classification process has to maximize the Q-value of a document and does not require a policy to evaluate. This can be achieved by using off-policy TD control algorithm called Q-Learning.

Its simplest form in 1-step Q-learning is defined by

$$Q^*(s,a) = R(s,a) + \gamma \max(Q(s',a')) \qquad (5)$$



State Value

In this case, the learned action-value function Q directly approximates Q*(s,a) the optimal action-value function, independent of the policy being followed. This simplifies the analysis of the algorithm and enables early convergence proofs.

1.Initialize the Q-value function Q(s,a) to some arbitrary value.
2.Repeat forever
a)Initialize one document as current state s.
b)Repeat Select a new document 'a ' from current document 's' using policy derived from Q
i) $a = \Pi_{Q\varepsilon}(s)$.
ii) Execute action 'a'.
iii) Observe the new document s'
and receive reward r =R(s,a).
iv) Calculate
$Q^*(s,a) = R(s,a) + \gamma \max (Q(s',a'))$
v) s = s'.
c) Until s is last document

Mapping Text to Q-Values

With calculated Q-Values for documents in the training data, we represent a value function using naïve bayes text classifier that maps documents into scalar values. Bayesian classifier predicts the class membership probabilities, such as the probability that given document belongs to particular class. In present work, there are 3 classes C1, C2, C3(Cricket,Tennis, Football).Given a new document , X, the classifier will predict that X belong to the class having the highest posteriori probability conditioned on X. That is, the Naïve Bayesian classifier assigns the new document X to the class Ci, if and only if,

$P(Ci / X) > P(Cj / X) \quad ;1 \leq j < m ; i \neq m.$ (6)

Thus maximizing P(Ci/X). The class Ci for which P(Ci/X) is maximized is called the maximum posteriori hypothesis. By Bayes theorem

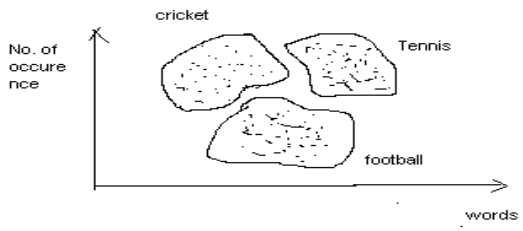
$P(Ci/X) = P(X/Ci)\, P(Ci) / P(X)$ (7)

As P(X) is constant for all classes only

P(X/Ci) P(Ci) need to be maximized. The class prior probability is estimated by

$P(Ci) = s_i / s$ (8)

Where s_i is the number training documents of class Ci and s is the total number of training documents.



Classifications of Documents

Given data set with many keywords, it would be extremely expensive to compute P(X/Ci). In order to reduce the computation in evaluating P(X/Ci), the naïve assumption of class Conditional Independence is made. Thus for n

$$P(X/Ci) = \prod_{K=1} P(x_k /Ci) \qquad (9)$$

The probability $P(x_1 /Ci)\ P(x_2 /Ci).P(x_n /Ci)$ can be estimated from the training data.

In order to classify the new document X, P(X/Ci) P (ci) is evaluated for each class Ci. The document X is assigned to m the class Ci if and only if

$$P(Ci / X) > P(Cj / X) \quad ;1 \leq j < m ; i \neq i$$

In other words, it is assigned to the class, for which P(X/Ci) is the maximum.

Bayesian inference

Bayesian inference is a method of statistical inference in which some kinds of evidence or observations are used to calculate the probability that a hypothesis may be true, or else to update its previously calculated probability. The term "Bayesian" comes from its use of the Bayes' theorem in the calculation process.

In practical usage, "Bayesian inference" refers to the use of a prior probability over hypotheses to determine the probability of a particular hypothesis given some observed evidence; that is, the probability that a particular hypothesis is true given some observed evidence (the *posterior probability* of the hypothesis) comes from a combination of the *prior probability* of the hypothesis and the compatibility of the observed evidence with the hypothesis (or *likelihood* of the evidence, in a technical sense). Bayesian inference is different from frequentist inference, which uses the sampling distribution of a statistic. Most elementary undergraduate-level statistics courses teach frequentist inference rather than Bayesian inference.

Evidence and changing beliefs

The primary foundation of Bayesian inference is the Bayesian interpretation of probability, which is distinct from other interpretations of probability in that it permits the attribution of probabilities to the truth and

falsehood of events that are not random, but rather the truth or untruth of which is simply unknown. Bayesian inference uses aspects of the scientific method, the method which involves collecting evidence that will be either consistent or inconsistent with a given hypothesis. As the evidence accumulates, the degree of confidence in a hypothesis ought to change. With enough evidence, the degree of confidence should become either very high or very low. Thus, Bayesian inference can be used to discriminate between conflicting hypotheses: hypotheses with very high support can be accepted as "true", and those with very low support should be rejected as "false". As with any inference method, however, results will naturally be biased subject to a-priori notions (either explicit or implicit) that are assumed before any evidence is ever collected, yielding results that are true or false relative to given assumptions. (This is a form of inductive bias.)

Bayesian inference uses a numerical estimate of the degree of confidence in a hypothesis before any evidence has been observed, and then it calculates a numerical estimate of the degree of confidence in the hypothesis after a set of evidence has been observed. (This process is repeated whenever additional evidence is obtained.) Bayesian inference usually relies on degrees of belief, or subjective probabilities, in the induction process, and it does not necessarily claim to provide an objective method of induction. Nonetheless, some Bayesian statisticians think that probabilities can have an objective value, and therefore Bayesian inference can provide an objective method of induction. See scientific method.

Bayes' theorem modifies probabilities, given new pieces of evidence, in the following way: where

- H represents a specific hypothesis, which may or may not be some null hypothesis.
- E represents the evidence that has been observed.
- $P(H)$ is called the *prior probability* of H that was inferred before new evidence became available.
- $P(E \mid H)$ is called the *conditional probability* of seeing the evidence E if the hypothesis H happens to be true. It is also called a *likelihood function* when it is considered as a function of H for fixed E.
- $P(E)$ is called the *marginal probability* of E: the *a priori* probability of witnessing the new evidence E under all possible hypotheses. It can be calculated as the sum of the product of all probabilities of any complete set of mutually exclusive hypotheses and corresponding conditional probabilities:

- $P(H \mid E)$ is called the *posterior probability* of H given E and is the new estimate of the probability that the hypothesis H is true, taking the evidence E into account.

The factor $P(E \mid H) / P(E)$ represents the impact that the evidence has on the belief in the hypothesis. If it is likely that the evidence E would be observed when the hypothesis under consideration is true, but, when no hypothesis is assumed, it is inherently unlikely that E would have been the outcome of the observation, then this factor will be large. Multiplying the prior probability of the hypothesis by this factor would result in a larger posterior probability of the hypothesis given the evidence. Conversely, if it is unlikely that the evidence E would be observed if the hypothesis under consideration is true, but *a priori* likely that E would be observed, then the factor would reduce the posterior probability for H. Under Bayesian inference, Bayes' theorem therefore measures how much new evidence should alter a belief in a hypothesis. Bayesian statisticians argue that even when people have very different prior subjective probabilities, new evidence from repeated observations will

tend to bring their posterior subjective probabilities closer together. However, others argue that when people hold widely different prior subjective probabilities their posterior subjective probabilities may never converge even with repeated collection of evidence. These critics argue that worldviews which are completely different initially can remain completely different over time despite a large accumulation of evidence.This behavior can be well described in a Bayesian framework.

Multiplying the prior probability $P(H)$ by the factor $P(E \mid H) / P(E)$ will never yield a probability that is greater than 1, since $P(E)$ is at least as great as (where denotes "and"), which equals (see joint probability).

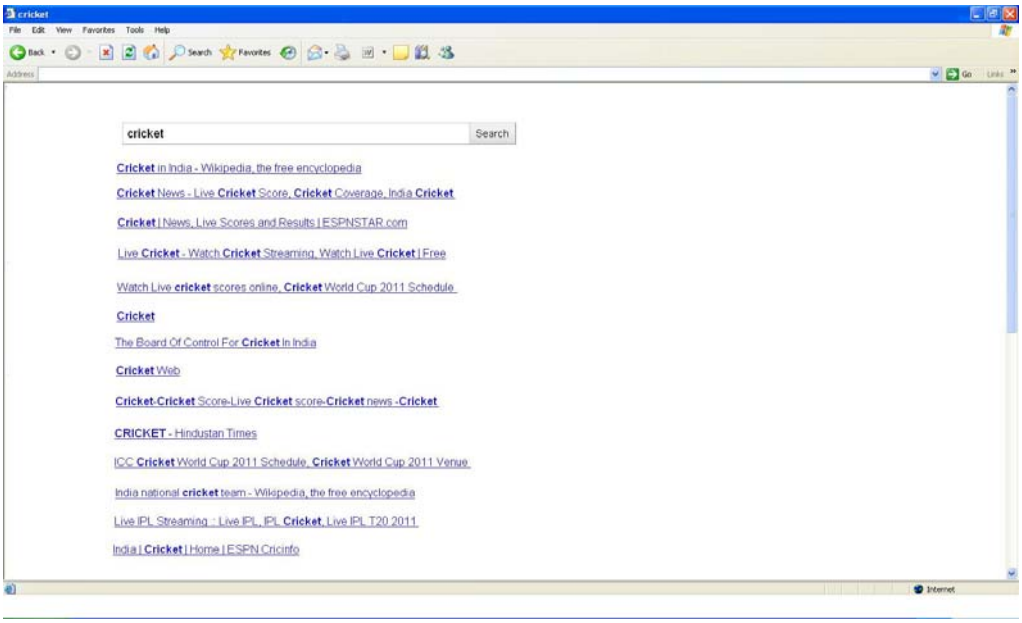
The probability of E given H, $P(E \mid H)$, can be represented as a function of its second argument with its first argument held fixed. Such a function is called a likelihood function; it is a function of H alone, with E treated as a parameter. A ratio of two likelihood functions is called a likelihood ratio, Λ. For example, where the dependence of Λ_E on H is suppressed for simplicity (as E might have been, except we will need to use that parameter below). Since H and not-H are mutually exclusive and span all possibilities, the sum previously given for the marginal probability reduces to As a result, we can rewrite Bayes' theorem as

We could then exploit the identity to exhibit $P(H \mid E)$ as a function of just $P(H)$ (and Λ_E, which is computed directly from the evidence). With two pieces of evidence E_1 and E_2, that are marginally and conditionally independent of each other, Bayesian inference can be applied iteratively. We could use the first piece of evidence to calculate an initial posterior probability, and then use that posterior probability as a new prior probability to calculate a second posterior probability given the second piece of evidence. Bayes' theorem applied iteratively yields.

CONCLUSION

The amount of information available on the Internet continues to grow. As this trend continues, this paper argues and attempts to provide an effective tool to the users to sort this bulk of increasing information on web. This work ventures to create text document classifier for efficient classification of domain specific documents using reinforcement learning technique. We also identify that the use of reinforcement learning technique significantly improve the efficiency of text document classifiers and show that reinforcement learning is a prominent technique. In this classification process text documents of specific domain are completely gone through for classification and thus improve the accuracy.

Text Categorization was deeply studied and analyzed to design the code and implement the various testing methods was done under the guidance of experienced project guide. We feel the solution provided now will suite to all the needs of various clients in the same industry but also we don’t rule the possibilities of further upgrading of this solution with the new and advance technologies and further additional requirements of the clients.



Sample Screen Shot

REFERENCES

[1] Andrew Mc Callum, Jason Rennie, Kamal Nigam and Kristie Seymour, "Building Domain-specific search engines with machine learning techniques", AAI-99 Spring Symposium on Intelligent Agents in Cyberspace, March 22-24, 1999, Stanford University in Palo Alto, California.

[2] Andrew Mc Callum, Kamal Nigam, Tom Mitchell and Sebastian Thrun, "Machine Learning", Second Edition, Kluwer Academic Publishers, Boston, 1999, PP. 1-34.

[3] Caesar Ferri, Peter Flach, Jose Herandez-orallo, "Delegating Classifiers", Proceedings of the 21 st International Conference on Machine Learning, Banff, Canada, 2004. PP. 523-536.

[4] Jason Rennie and Andrew Mc Callum, "Using Reinforcement Learning to Spider the Web efficiently", Proceedings of the 16th International conference on Machine Learning, June 27, 1999, Bled, Slovenia

[5] Jason D.M. Rennie, Lawrence Shih, Jaime Teevan, David R. Karger, "Tackling the Poor Assumptions of Naïve Bayes Text Classifier", Proceedings of the 20 th International Conference on Machine Learning, Washington DC, 2003, PP. 1023-1035.

[6] T.Joachims, T.Frietag, T. Mitchell, Web Watcher : A Tour guide for the World Wide Web. Proceedings of IJCAI, August 23-29, 1997, Nagoya, Japan .PP.770-777.

[7] John Langford and Bianca Zadronzy, "Relating Reinforcement Learning Performance to Classification Performance", Proceedings of the 22nd International Conference on Machine Learning, Bonn, Germany, 2005, PP.254-265.

[8] Michail G. Lagoudakis and Ronald Parr, "Reinforcement Learning Classification: Leveraging Moder Classifier", Proceedings of the 20 th International Conference on Machine Learning, Washington DC, 2003, PP. 345-352.

[9] Raghu Krishnapuram, Krishna P Chitrapura and Sachindra Joshi, "Classification Of Text Documents Based on Minimum System Entropy", Proceedings of the 20 th International Conference on Machine Learning, Washington DC, 2003, PP. 123-134.

[10] S. Russell and P. Norvig, Arti_cial Intelligence: A Modern Approach. Englewood Cliff, New Jersey: Prentice-Hall, Inc., 1995, ch. 20.

[11] R. S. Sutton and A. G. Barto, Reinforcement Learning: an introduction. Cambridge, Massachusetts: MIT Press, 1998.

[12] Tom Mitchell, "Machine Learning", First Edition, McGraw Hill, New York, 1997, PP.153-385.