Application of Machine Learning Techniques to Differential Diagnosis of Erythemato-Squamous Diseases

Narin Emeksiz
Havelsan, Ankara Turkey

And

H. Altay Güvenir
Bilkent University, Ankara, Turkey

ABSTRACT

This paper is about the implementation of a visual tool for Differential Diagnosis of Erythemato-Squamous Diseases based on the classification algorithms; Nearest Neighbor Classifier (NN), Naive Bayesian Classifier using Normal Distribution (NBC) and Voting Feature Intervals-5 (VFI5). This tool enables the doctors to differentiate six types of Erythemato-Squamous Diseases using clinical and histopathological parameters obtained from a patient. The program also gives explanations for the classifications of each classifier.

Keywords: Machine Learning, Classification, Dermatology, Feature Projections

1. INTRODUCTION

The aim of the project is to implement a visual tool for Differential Diagnosis of Erythemato-Squamous Diseases based on the three different classification algorithms; namely, Nearest Neighbor Classification (NN), Naive Bayesian Classifier using Normal Distribution (NBC) and Voting Feature Intervals-5 (VFI5). The program enables the doctor to see the classifications made by each classifier, along with the explanations of each classification. The data set was obtained from the Department of Dermatology of Gazi University in Ankara.

The next section gives the description of the problem. Section 3 presents the three classification algorithms incorporated in the tool. Section 4 outlines the design steps of the project. Section 5 concludes the paper.

2. PROBLEM DESCRIPTION

The differential diagnosis of erythemato-squamous diseases is an important problem in dermatology. The diseases in this group are psoriasis (C_1), seboreic dermatitis (C_2), lichen planus (C_3), pityriasis rosea (C_4), cronic dermatitis (C_5) and pityriasis rubra pilaris (C_6). They all share the clinical features of erythema and scaling, with very little differences. These diseases are frequently seen in the outpatient departments of dermatology. At the first sight all of the diseases look very much alike with the erythema and scaling. When inspected more carefully some patients have the typical clinical features of the disease at the predilection sites (localization of the skin where a disease preters) while another group has a typical localization.

Patients are first evaluated clinically with 12 features. The degree of erythema and scaling, whether the borders of lesions are definite or not, the presence of itching and koebner phenomenon, the form of the papules, whether the oral mucosa, elbows, knees and the scalp are involved or not, whether there is a family history or not are important for the differential diagnosis. For example the erythema and scaling of chronic dermatitis is less than of psoriasis, the koebner phenomenon is present only in psoriasis, lichen planus and pityriasis rosea. Itching and polygonal papules are for lichen planus and follicular papules are for pityriasis rubra pilaris. Oral mucosa is predilection site for lichen planus while knee, elbow and scalp involvements are of psoriasis. Family history is usually present for psoriasis and pityriasis rubra pilaris usually starts during childhood.

Some patients can be diagnosed with these clinical features only, but usually a biopsy is necessary for the correct and definite diagnosis. Skin samples were taken for the evaluation of 22 histopathological features. Another difficulty for the differential diagnosis is that a disease may show the histopathological features of another disease at the beginning stage and may have the characteristic features at the following stages. Some samples show the typical histopathological features of the disease while some do not. Melanin incontinence is a diagnostic feature for lichen planus, fibrosis of the papillary dermis is for chronic dermatitis, exocytosis may be seen in lichen planus, pityriasis rosea and seboreic dermatitis. Acanthosis and parakeratosis can be seen in all the diseases in different degrees.

suprapapillary epidermis are diagnostic for psoriasis. Disappearance of the granular layer, vacuolization and damage of basal layer, saw-tooth appearance of retes and a band like infiltrate are diagnostic for lichen planus. Follicular horn plug and perifollicular parakeratosis are hints for pityriasis rubra pilaris.

The features of a patient are represented as a vector of features which has 34 entries for each feature value. In the dataset, the family history feature has the value 1 if any of these diseases has been observed in the family, and 0 otherwise. The age feature simply represents the age of the patient. Every other feature (clinical and histopathological) was given a degree in the range of 0 to 3. Here, 0 indicates that the feature was not present, 3 indicates the largest amount possible, and 1, 2 indicate the relative intermediate values. Each feature has either nominal (discrete) or linear (continuous) values having different weights showing the relevance to the diagnosis.

3. SOLUTION ALGORITHMS

In this section we describe the three classification algorithms used in the tool; namely, the Nearest Neighbor Classifier, the Naïve Bayesian Classifier and the Voting Feature Intervals Classifier.

3.1. The Nearest Neighbor Classifier Algorithm

One of the classification algorithms that we used in this project is the NN classifier as it is a simple and common algorithm. The NN classification is based on the assumption that examples that are closer in the instance space are of the class. NN algorithm assumes that a new test instance belongs to the same class as its nearest neighbor among all stored training instances. In this project our aim is to classify a single test instance depending on the previously established training data set. Therefore, we did not include the training phase to the project, the methodology that we use is directly inserting the output data into arrays after performing the training process in a separate medium. So, for the implementation of the NN classification algorithm we directly stored the train data features and class values in two separate arrays as these are the data sets produced after the training process. Currently, the data set for the domain contains 366 instances. We first used all of these instances to obtain a description of the domain. The structures of the arrays are shown in Figure 1.

```
int train_value[366][34]={

{2,2,0,3,0,0,0,0,1,0,0,0,0,0,0,3,2,0,0,0
,0,0,0,0,0,0,0,3,0,0,0,1,0,55},

{3,3,3,2,1,0,0,0,1,1,1,0,0,1,0,1,2,0,2,2
,2,2,2,1,0,0,0,0,0,0,0,1,0, 8} ,
.....................

{2,1,3,1,2,3,0,2,0,0,0,2,0,0,0,3,2,0,0,0
,0,0,0,0,3,0,2,0,1,0,0,2,3,50},

{3,2,2,0,0,0,0,0,3,3,0,0,0,1,0,0,2,0,2,3
,2,3,0,2,0,2,0,0,0,0,0,3,0,35},
};
int train_class[366]=
{2,1,3,1,3,2,5,3,4,4,1,2,2,1,3,4,2,1,3,5
,6,2,5,3,5,1,6,5,2,3,

1,2,1,1,4,2,3,2,3,1,2,4,1,2,5,3,4,6,2,3,
3,4,1,1,5,1,2,3,4,2,
  .....................

1,5,5,3,1,5,5,6,6,4,4,6,6,6,1,1,1,5,5,1,
1,1,1,2,2,4,4,3,3,1};
```

Figure 1. Training data set.

All the feature values are assumed to have linear values. The distance metrics used to obtain the distance between two instances in the NN classification algorithm is the Euclidean distance metric. The NN algorithm is more effective when the features of the domain are equally important. It will be less effective when many of the features are misleading or irrelevant to classification. To overcome this problem, the features are assigned weights such that the irrelevant features have lower weights (w_f) while the strongly relevant features are given higher weights (w_f). Giving different weights to each feature modify the importance of the feature in the classification process such that a relevant feature becomes more important than a less relevant one. We used a genetic algorithm to learn the feature weights to be used with the Nearest Neighbor classification algorithm. We applied the same genetic algorithm to determine the weights of the features in our domain to be used with the VFI5 algorithm. Koebner phenomenon has the highest weight 0.0620. Inflammatory mononuclear infiltrate is also important in the classification, with the weight of 0.0527. On the other hand, the features acanthosis, follicular horn plug, munro microabcess, and age are found to be the least relevant.

3.2. Naive Bayesian Classifier Using Normal Distribution

Bayesian classifier is an algorithm that approaches the classification problem using conditional probabilities of the features. The probability of the instance belonging to a single

probabilities of classes and the feature values for an instance. Naive Bayesian Classifier (NBC) assumes that features are independent. In NBC, each feature participates in the classification by assigning probability values for each class, and the final probability of a class is the product of each single feature probabilities; and for an *n* dimensional domain, the probability of the instance belonging to a class ($P(e|C_i)$) can be computed as

$$P(e \mid C_i) = \prod_{f=1}^{n} P(e_f \mid C_i).$$

NBC estimates the conditional probability density function $P(e|C_i)$ for a given feature value e_f for the f^{th} feature using the frequency of observed instances around e_f. $P(e_f \mid C_i)$ for the nominal features is the ratio of the number of training examples of class C_i with value e_f for feature *f* over total number of training examples of class C_i. $P(e|C_i)$ for continuous features is computed by assuming normal distribution.

In this project our aim is to classify a single test instance depending on the previously established training data set. Therefore, we did not include the training phase of the NBC Algorithm to the project, we directly filled in the arrays after performing the training process in a separate medium. So, for the implementation of the NBC classification algorithm we store the variance and the mean of the linear values in two arrays called `Variance[34]` and `Mean[34]` arrays.

The NBC algorithm handles the missing feature values by ignoring the feature with the missing value instead of ignoring the whole instance. When *e* has unknown value for *f*, the conditional probability $P(e|C_i)$ of each class C_i is assigned to 1, which has no effect on the product of probabilities distributed by each feature.

3.3. Voting Feature Intervals-5 Algorithm

The VFI5 classification algorithm represents a concept description by a set of feature value intervals [2, 5]. The classification of a new instance is based on a voting among the classifications made by the value of each feature separately. It is a non-incremental classification algorithm; that is, all training examples are processed at once [3]. From the training examples, the VFI5 algorithm constructs intervals for each feature. An interval is either a range or point interval. A range interval is defined on a set of consecutive values of a given feature whereas a point interval is defined for a single feature value. For point intervals, only a single value is used to define that interval. For range intervals, on the other hand, it suffices to maintain only the lower bound for the range of values, since all range intervals on a feature dimension are linearly ordered. The lower bound of the range intervals obtained from the training instances are installed into an array called `intervalLower` and the number of segments formed for each feature value is stored in the array `NoIntervals` directly at the beginning of the `vfi` function so no training process is done. For each interval, a single value and the votes of each class in that interval are maintained. Thus, an interval may represent several classes by storing the vote for each class. The votes given to the classes for each interval for each feature values are stored in the `intervalVotes` array.

The training phase is performed in another platform and the only operation that takes place in the training process is to find the end points for each class *C* on each feature dimension *f*. End points of a given class *C* are the lowest and highest values on a linear feature dimension *f* at which some instances of class *C* are observed. On the other hand, end points on a nominal feature dimension *f* of a given class *C* are all distinct values of *f* at which some instances of class *C* are observed. There are 2*k* end points for each linear feature, where *k* is the number of classes. Then, for linear features the list of end-points on each feature dimension is sorted. If the feature is a linear feature, then point intervals from each distinct end point and range intervals between a pair of distinct end points excluding the end points are constructed. If the feature is nominal, each distinct end point constitutes a point interval. The number of training instances in each interval is counted. These counts for each class *C* in each interval *i* on feature dimension *f* are computed.

For each training example, the interval *i* in which the value for feature *f* of that training example *e* falls is searched. If interval *i* is a point interval and e_f is equal to the lower bound (same as the upper bound for a point interval), the count of the class of that instance in interval *i* is incremented by 1. If interval *i* is a range interval and e_f is equal to the lower bound of *i* (falls on the lower bound), then the count of class e_c in both interval *i* and (*i*-1) are incremented by 0.5. But if e_f falls into interval *i* instead of falling on the lower bound, the count of class e_c in that interval is incremented by 1 normally. There is no need to consider the upper bounds as another case, because if e_f falls on the upper bound of an interval *i*, then e_f is the lower

a nominal feature are point intervals, the effect of `countInstances` is to count the number of instances having a particular value for nominal feature *f*.

To eliminate the effect of different class distributions, the count of instances of class *C* in interval *i* of feature *f* is then normalized by `classCount[C]`, which is the total number of instances of class *C*. As these operations are performed in the training phase, they are not included in the program. Only the data set formed after the training phase is directly initialized to the arrays `intervalLower`, `NoIntervals` and `intervalVotes`.

The classification process starts by initializing the votes of each class to zero. The classification operation includes a separate preclassification step on each feature. The preclassification of feature *f* involves a search for the interval on feature dimension *f* into which e_f falls, where e_f is the value test example *e* for feature *f*. If that value is unknown (missing), that feature does not participate in the classification process. Hence, the features containing missing values are simply ignored. Ignoring the feature about which nothing is known is a very natural and plausible approach [1].

If the value for feature *f* of example *e* is known, the interval *i* into which e_f falls is found. That interval may contain training examples of several classes. The classes in an interval are represented by their votes in that interval. For each class *C*, feature *f* gives a vote equal to `intervalVote[f,i,C],` which is vote of class *C* given by interval *i* on feature dimension *f*. If e_f falls on the boundary of two range intervals, then the votes are taken from the point interval constructed at that boundary point. The individual vote of feature *f* for class *C*, is then normalized to have the sum of votes of feature *f* equal to 1. Hence, the vote of feature *f* is a real-valued vote less than or equal to 1. After every feature completes their voting, the individual vote vectors are summed up to get a total vote vector `totalVotes`. Finally, the class with the highest vote from the total vote vector is predicted to be the class of the test instance. The implementation of the VFI algorithm is in Figure 4:

4. DESIGN OF THE PROJECT

As this application is designed to be used by the doctors who are not advanced computer users, we aimed to implement the user interface of the Erythemato-Squamous Diseases application as user friendly. The program has been implemented in C++ and runs on Windows environment.

Being a department of a hospital, dermatology department inherits all processes that take place in a hospital. Everyday some number of patients are applied to the department as they have symptoms which are the signs of a skin disease. In order to keep track of each patient and prepare history for the hospital, we constructed a database in which the detailed information of each patient would be kept. The ByopsiNo is selected as the primary key so that it is unique for each patient in the database. Also indexes are formed for PatientName, PatientSurname and PatientName.

```
findInterval(value, feature f)
begin
  while ((intervalLower[f,s]< value)
         && (s < NoIntervals[f]))
           increase s
  if (intervalLower[f,s] == value)
     return(s)
   else
     return(s-1)
end

featureVotes(e, f, Votes[])
begin
  if e_f is known
     s = findInterval(e_f, f);
     for each class value C
        Votes[C] = intervalVote[f,s,C];
  return;
end

vfi5(e)
begin
  initialize the totalVotes array
  initialize the Votes of each feature
for each class
  for each feature f
    featureVotes(e, f, Votes);
    for each class C
       totalVotes[C] += (Votes[C] * w_f);
   return (the class C having the
           largest Votes[C])
end
```

Figure 4. The VFI5 algorithm.

In the data set constructed for this domain, the ByopsiNo is the label that is given to each patient for the differentiation, name and surname belongs to the patient, the doctor's diagnosis field stores the doctors prediction about the disease and its range is from 1 to 6 each reflecting the label of the 6 eythemato-squamous diseases, family history feature has the value 1 if any of these diseases has been observed in the family, and 0 otherwise. The age feature simply represents the age of the patient. Every other feature (clinical and histopathological) was given a degree in the range of 0 to 3. Here, a 0 indicates that the

feature was not present, a 3 indicates the largest amount possible, and 1, 2 indicate the relative intermediate values.

4.1 Database Operations

Keeping the patient records; entrance of a new patient, searching for an already recorded patient or extracting a patient from the registration are some of the operations that leads to the construction of a database. All these operations are performed by specially prepared forms. The Patient Record Entrance Form shown in Figure 5 enables the user to enter all the information about the patient.

If the buttons labeled Clinical Features or Histopathological Features is pressed one of the following forms in Figure 6 or Figure 7 is opened and enables the user to enter the feature values only by marking the corresponding values.

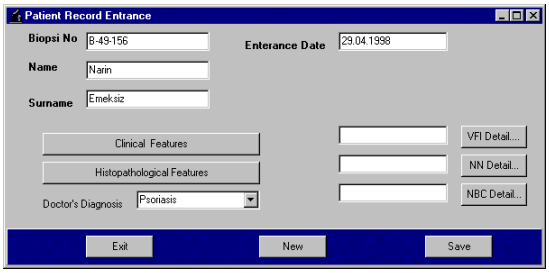

Figure 5. Patient Record Entrance.

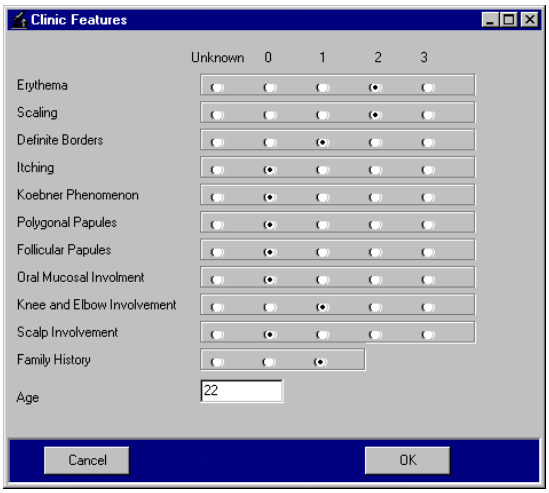

Figure 6. Clinical Features.

If a value is not entered in these forms their values are recorded as unknown to the database and each prediction algorithm handles these unknowns in a specific way depending on the handling mechanism of the algorithm. Classification algorithms make prediction even if one of the feature values of clinical or histopathological features is entered. The result of one prediction is shown in Figure8.

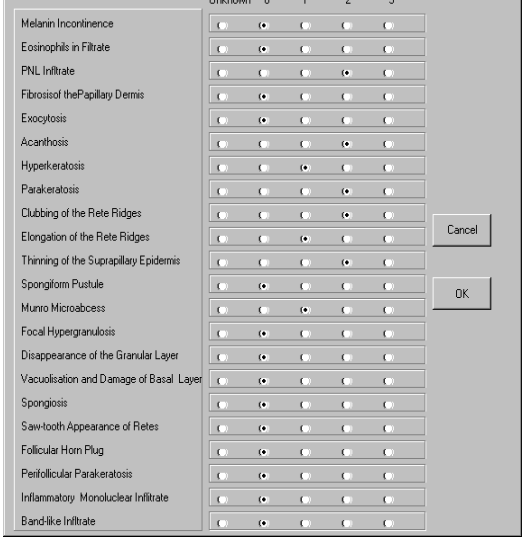

Figure 7. Histopathological Features

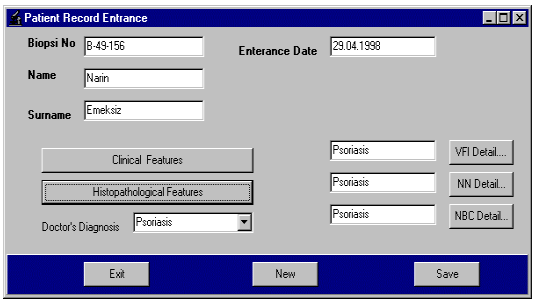

Figure 8. Results of the classifiers.

As keeping BiopsyNo in mind is a difficult task for a human being, we based our searching methodology on different indexes. We have implemented four searching craters; BiopsyNo, Name, Surname, and both Name and Surname.

For the update operation; the BiopsyNo which is on the form is taken and the database is opened as indexed by the BiopsyNo.

4.2. Explanations

As one of the main aims of the project is to be an assistant tool in the training of the dermatology students; the implementation of the three different classification algorithms are placed in both Patient Data Entrance and Searched Patient Details forms by giving the doctor the chance to compare his own classification with the prediction of the algorithms. The detailed information given for each of the classification algorithms can provide the flexibility to the application to be used both in the hospital and in the education process of the intern-doctors.

If the detail button for the NBC is pressed then the form which shows the probability of each of 34 features belonging to any erythemato-

Figure 9 which contains the detailed information about the patient is retrieved. This form also enables the doctor to make any update on the previously recorded data set; to examine the previous patients details and to see the predictions.

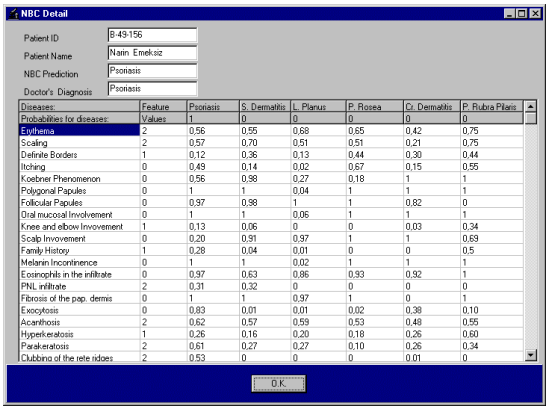

Figure 9. Explanations for NBC classification.

When the detail button for the NN classifier is pressed, the explanation for the NN algorithm's prediction is provided as shown in Figure 10. As NN algorithm assumes that a new patient has the same disease as its nearest neighbor; the design of the NN-Detail form includes both the patient for whom the NN makes classification and the patient, which has the most similar feature values.

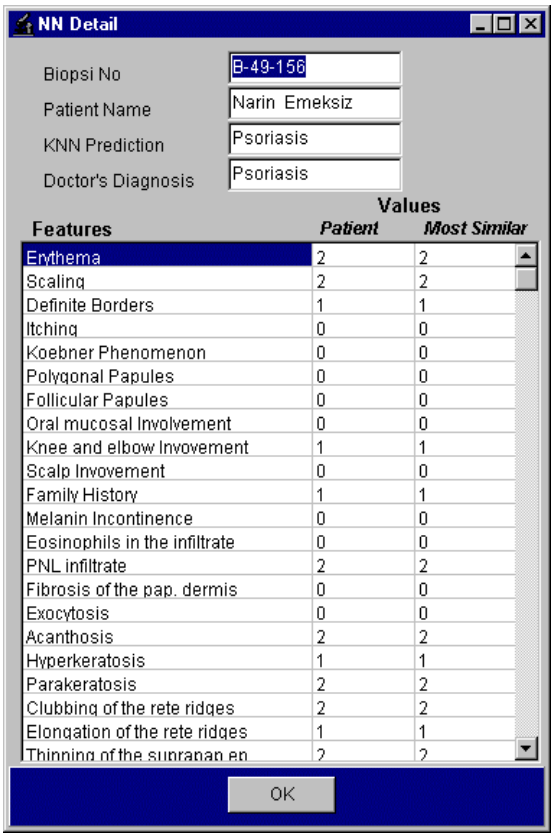

Figure10. Explanations for NN classification.

When the detail button for VFI is pressed, the explanation for the VFI-5 algorithm's classification is provided as shown in Figure 11 is displayed.

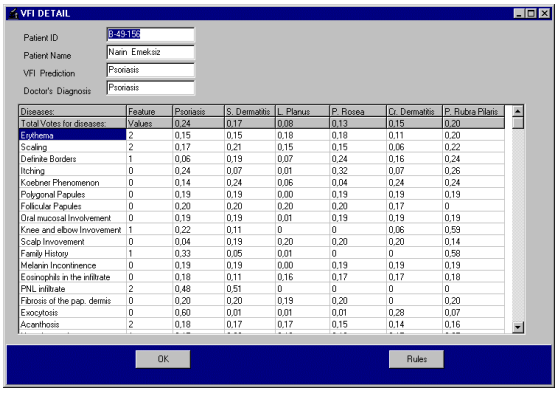

Figure 11. Explanations for the VFI classification.

The rules table in Figure 12 displays the votes given to each class for each of the 34 features. These votes are learned during the training of the VFI5 algorithm.

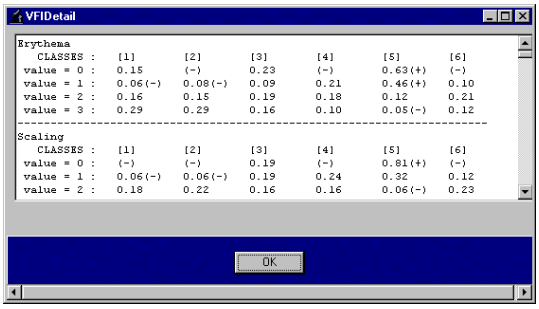

Figure 12. Rules used by the VFI classifier.

5. CONCLUSION

In our opinion using this tool in the education process provides a more colorful environment for the doctors than huge hard covered materials. Also the students of the Medical Schools can use the tool for testing their knowledge by comparing their predictions with the classifications done by the algorithms. Also another advantage of the tool is to be a guide to the doctors in constructing their own classification mechanisms by examining the working methodologies of the algorithms presented in the detail sections.

Today, this visual tool for the differentiation of erythemato squamous diseases is ready to use by the doctors and the computer scientists who are interested in machine learning algorithms.

ACKNOWLEGMENTS

We would like to thank Dr. Nilsel İlter of Gazi University in Ankara for providing the training data set and carefully evaluating the results of the classifiers.

REFERENCES

[1] G. Demiroz. "Non-Incremental Classification Learning Algorithms Based on Voting Feature Intervals." Bilkent University, Dept. of Computer Engineering and Information Science, MSc. Thesis, 1997.

[2] G. Demiroz, and H.A. Guvenir, and Nilsel Ilter. Differential Diagnosis of Erythemato-Squamous Diseases Using Feature Intervals. In *Prooceedings of the Sixth Turkish Symposium on Artificial Intelligence and Neural Networks* (TAINN'97), 190-194, 1997.

[3] H.A. Guvenir and I.Sirin, Classification by Feature Partitioning, *Machine Learning*, 23:47-67, 1996.

[4] H. A. Güvenir, G. Demiröz and N. Ilter, Learning Differential Diagnosis of Erythemato-Squamous Diseases using Voting Feature Intervals, *Artificial Intelligence in Medicine*, Vol. 13, No. 3, (1998), pp. 147-165.