Pacific Symposium on Biocomputing 7:41-52 (2002)

INFERRING GENOTYPE FROM CLINICAL PHENOTYPE THROUGH A KNOWLEDGE BASED ALGORITHM

B.A. MALIN, L.A. SWEENEY, Ph.D.
School of Computer Science, Carnegie Mellon University
Pittsburgh, PA 15213, USA

Genomic information is becoming increasingly useful for studying the origins of disease. Recent studies have focused on discovering new genetic loci and the influence of these loci upon disease. However, it is equally desirable to go in the opposite direction – that is, to infer genotype from the clinical phenotype for increased efficiency of treatment. This paper proposes a methodology for such inference. Our method constructs a simple knowledge-based model without the need of a domain expert and is useful in situations that have very little data and/or no training data. The model relates a disease's symptoms to particular clinical states of the disease. Clinical information is processed using the model, where appropriate weighting of the symptoms is learned from observed diagnoses to subsequently identify the state of the disease presented in hospital visits. This approach applies to any simple genetic disorder that has defined clinical phenotypes. We demonstrate the use of our methods by inferring age of onset and DNA mutations for Huntington's disease patients.

1 Background

1.1 Genotype-Phenotype Relationships

Over the past decade, growing interest has surfaced in recognizing relationships between the genotype and clinical phenotype of an individual. It is believed that more efficient treatment of many diseases can be achieved through tailoring drug administration to specific genotypes.[1] With this in mind, one must consider that the etiology of many diseases resides in a combination of genetic predispositions, environmental variables, and random chance. Genetic influence varies among diseases, ranging from a weak influence on Alzheimer's disease to a deterministic effect on sickle cell anemia.

This paper addresses the relationship in single gene mutation diseases known as simple Mendelian traits. The traits are an interesting study from the standpoint that their DNA mutation is considered the direct cause of the disease and permit a wide range of genotype-phenotype relationships. For example, the autosomal recessive disease cystic fibrosis is caused by mutations in the cystic fibrosis transmembrane conductance regulator gene, of which over 750 mutations have been documented. While cystic fibrosis' clinical expression is variable, the phenotypes have been demonstrated to relate to particular mutations of the gene.[2]

Though the relationship between the genotype and some aspect of the clinical phenotype is known, the relationship is often obscured in standardized medical information. This paper describes a generally applicable method for discovering the particulars of this relationship from standardized medical information as they relate to individual patients. We demonstrate our general method to determine clinical phenotype of the autosomal dominant disorder known as Huntington's disease in a group of patients. Huntington's disease is caused by a CAG trinucleotide repeat expansion of the HD gene, a feature relatively independent of the observed clinical features.[3] Rather, the size of the repeat harbors an inverse exponential relationship to the age of onset of the disease, a feature of the clinical phenotype not recorded in general medical information.[4] Our methods therefore, are utilized to infer non-recorded features of the clinical phenotype (such as age of onset) from standardized hospital information to reveal characteristics of the genotype.

1.2 Knowledgebase and Statistical Learning Approaches

Nowadays knowledge-based systems, which gained tremendous popularity in the 1980's, and data mining techniques, which gained tremendous popularity in the 1990's, are often viewed as rival approaches. The era of expert systems began with exciting systems like DENDRAL, which inferred molecular structure from information provided by a mass spectrometer[5], and MYCIN, which diagnosed blood infections[6]. But the era of expert systems ended with disillusionment as the costs of constructing real-world knowledge-based systems far exceeded their perceived benefits. Excitement shifted to neural networks and statistical data mining techniques, in part, because they provided results using standardized learning models with little or no explicit representation of domain knowledge required. But as the problem space becomes more complex, the advantages of a knowledge-based approach become apparent. Examples include games such as chess, checkers, and backgammon where efforts to learn an evaluation method using knowledge of the game are much more successful then methods void of domain knowledge[7].

In this paper, we tackle a problem in which a neural network, for example, could be used if training data were available; but in this environment, we assume no training data are available. Our approach constructs an initial knowledge-based model with minimal effort by relating diagnoses to a disease's symptoms and relating those symptoms, in turn, to stages of the disease. The represented knowledge is not from a domain "expert" but from simple extractions drawn from common literature and so, these initial mappings may be inconsistent. We therefore calibrate the model by generalizing, specializing and partitioning the initial mappings as needed. Finally, we apply the model to infer genotype for patients.

2 Methods

Materials needed for this approach are standardized hospital data, general facts about the clinical presentation of the disease, and for a sample of individuals, known features of the clinical phenotype. Following are descriptions of these materials, which are necessary for use with our approach.

INDIV1	AGE1	DOB1	SEX1	ZIP1	ADMIT1	{DIAGNOSES}
INDIV2	AGE2	DOB2	SEX2	ZIP2	ADMIT2	{DIAGNOSES}
INDIV2	AGE2	DOB2	SEX2	ZIP2	ADMIT2	{DIAGNOSES}
INDIV2	AGE2	DOB2	SEX2	ZIP2	ADMIT2	{DIAGNOSES}
INDIV3	AGE3	DOB3	SEX3	ZIP3	ADMIT3	{DIAGNOSES}

Figure 1. Longitudinal medical profiles from clinical information databases. Multiple visit profile is shaded.

2.1. Inference Algorithm Definition

First, we consider the collections of inpatient hospital visits. The National Association of Health Data Organizations reported that 44 of 50 states have legislative mandates to gather hospital-level data on each patient visit. As shown in Figure 1, patient demographics, hospital identity, diagnosis codes, and procedure codes are among the attributes stored with each hospital visit. Previous research has demonstrated that publicly available discharge data permits the formation of genetic population subsets.[8]

INPUT	Patient profiles of clinical data (basic hospital visit information)
ASSUMES	Disease is known to be temporal or constrained to an exclusionary status of clinical phenotype for the duration of a profile
Step I	Manually map diagnoses to symptoms to clinical phenotype states and diagnoses to clinical phenotype states
do	
Step II	Automatically adjust symptoms to cover clinical phenotype states
Step III	Learn accuracy of diagnosis codes using the defined mappings
Step IV	Automatically classify each patient visit into a clinical phenotype state based on the mappings
Step V	Align the predicted clinical phenotype states for each set of patient visits to optimally respect temporal or disease stage constraints
until	predictions converge
OUTPUT	Specific inferences and/or constraints regarding genotype of patient

Figure 2. Genotype inference algorithm.

Second, we consider the current corpora of knowledge about particular diseases. General information about symptoms and clinical presentation of the disease are needed. Such information can be found in articles on MEDLINE, on authoritative web sites devoted to the disease, and in general medical texts. From the study of this information, the symptoms of the disease in question can be defined. Symptoms are defined narrowly enough, so that diagnoses map to a single symptom. With the previous information available, we present an algorithm for knowledge representation and inference in Figure 2. Steps of the algorithm are discussed in the following subsections.

2.2 Manual Mapping of Diagnoses and Symptoms (Step I)

The simplest model is the direct mapping of each diagnosis to its corresponding disease states. Such a model permits the use of a neural network for parameter estimation for a defined model as depicted in Figure 3a. However, such learning tools require ample training data. Yet, this study uses a small amount of data and no training data, and as such, we attempt to use known knowledge about a disease to tune the model by adding an internal layer of nodes. Each node in the additional layer represents a group of diagnoses, as shown in Figure 3b. The question becomes, “What criteria is best for grouping the raw diagnoses?” Three possibilities are explored.

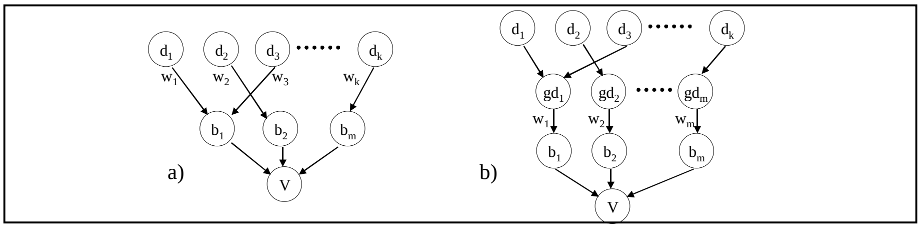

Figure 3. Clinical phenotype inference models. **a)** Diagnoses (d_i) are directly mapped to vectors describing clinical phenotype states (b_i). **b)** Diagnoses are grouped (gd_i). Basis for grouping may be generalization of code, semantic text description, defined symptoms, or some other binning feature. *V* is the final combined result.

One possibility is to generalize on diagnosis codes. In the case of ICD-9 diagnosis codes, codes having the same leftmost digits are semantically related. The fewer the number of leftmost digits found in common between codes, the greater the number of codes to which those digits refer. By "generalizing" diagnosis codes, codes that share the same left most digits are grouped together. A second possibility uses the textual descriptions that accompany the ICD-9 diagnosis codes. String matching words or phrases found in the text descriptions provide the basis for grouping diagnosis codes together. Finally, a third possibility identifies symptoms based on published articles and books about the disease in question. Each symptom then has an associated set of diagnoses not dependent on the coding structure or common words in the description. In all three possibilities, diagnosis codes are grouped into sets, which we generally refer to as "symptoms." Tools for performing each of these ways of grouping diagnosis codes into symptoms were constructed.

Each clinical phenotype may be thought of as a different state of the disease. Of the states, only one may be presenting at any particular point in time. Thus, the disease can be characterized as a vector of 0s and 1s with k positions, where k is the number of distinct phenotypes associated with the disease. From published descriptions of the disease, groups of diagnoses ("symptoms"), which are reported as being related to states of the disease, are designated a 1 in each vector position that corresponds to those states and a 0 in all other vector positions. Each symptom therefore has a non-zero "state vector" associated with it that identifies the states in which the symptom is expected to appear. For example, the vector [1,1,0,0] represents a disease having 4 states and the symptom related to this vector has diagnoses that only appear in the first two states. Each b_i in Figure 3 represents such a vector.

At this point, we have an initial knowledge-based model that relates each diagnosis code to states of the disease through a symptom as shown in Figure 3b. We now look at an alternative mapping directly from diagnosis codes to disease states as shown in Figure 3a. Using literature about the disease, each diagnosis code appearing in the hospital visits is directly mapped to corresponding disease states without the use of symptoms. These are manual mappings that are not calibrated by actual observations or mapped by a real domain expert. They provide a second guess at the relationship between diagnosis and disease states.

2.3 Mapping Adjustment (Step II)

The question becomes, “Do the mappings consistently define the clinical phenotype?” For example, a hospital visit with diagnoses that when binned into their respective symptoms (using the model in Figure 3b) could provide the vectors {[0,1,1,0] , [0,1,1,0] , [0,0,1,0]}, while those same diagnoses mapped directly to disease states (using the model in Figure 3a) could yield only the fourth state. Using the vectors resulting from the symptoms, the fourth clinical phenotype would never be considered. Thus, the scenario exists where a symptom may under-represent the states provided in the diagnosis-to-state mappings. Similarly, the situation could occur where a symptom over-represents the diagnosis-to-state mappings. If a symptom vector presents [0,1,1,0], but the corresponding diagnosis-to-state mappings appear only in the second clinical phenotype, then the symptom falsely assumes the third phenotype.

We address these scenarios, with a method to update the symptoms, such that our model becomes consistent in its mappings. Our approach involves finding maximally specific mappings in the space defined by the initial brute-force mappings. This approach builds on prior work in the area of concept learning using general-to-specific ordering[9]. Let Sy_i be the state vector for the i^{th} symptom. Let D_i be the state vector for the set of diagnoses mapped to Sy_i, but whose state is determined from the diagnosis to disease state mappings. In each position j of the vector D_i, the number is 1 if any diagnosis, within D_i, maps to clinical state j, and 0 otherwise. For each Sy_i and D_i, there are four possible scenarios each with a specific action as defined in Table 1.

Table 1. Symptom refinement based on diagnoses vector comparison

Scenario	Example	Action
$D_i = Sy_i$	$D_i = [0,1,0,0]$, $Sy_i = [0,1,0,0]$	None
$D_i < Sy_i$	$D_i = [0,1,0,0]$, $Sy_i = [0,1,1,1]$	$Sy_i = D_i$
$D_i > Sy_i$	$D_i = [0,1,1,1]$, $Sy_i = [0,1,0,0]$	
Not($D_i = Sy_i$) And Not($D_i < Sy_i$) And Not($Sy_i < D_i$)	$D_i = [1,1,0,0]$, $Sy_i = [0,1,1,0]$	Partition symptoms

The first scenario, $D_i=Sy_i$, is trivial, the symptom does not change. When there are more states found for a symptom than the diagnoses provide, $D_i>Sy_i$, or the symptom covers too many states, $D_i<Sy_i$, we set Sy_i equal to D_i. The final scenario, when the two vectors are unequal we partition Sy_i into several symptoms and redefine the state mappings. The partitioning rule is explained with an accompanied example. Let D_i= [1,1,0,0] and Sy_i = [0,1,1,0]. Diagnoses within D_i that are contained by Sy_i (≤) remain mapped to Sy_i. So, diagnoses that provide [0,1,0,0], [0,0,1,0], or [0,1,1,0], remain mapped to Sy_i. Next, create new symptoms with the vectors defined to be equal to the largest range of states spanning the remaining diagnoses. Thus, if there existed diagnoses with vectors [1,0,0,0] and [1,1,0,0] a new symptom would be created with the vector [1,1,0,0]. Yet, if the remaining diagnoses all had the vector [1,0,0,0], the new symptom would have a vector of [1,0,0,0]. Partitioning modifies the structure of the model. We now have mappings of diagnoses to symptoms and symptoms to states that are generally specific to the mapping of diagnoses to states.

2.4 Learning Diagnosis Weights (Step III)

For any particular hospital visit, we are interested in what state of the disease a patient is presenting. To accomplish this, we must determine how accurate a diagnosis code is for predicting any particular state. Initially we cannot make such a determination because we have no training data. Hospital visits are not initially classified as representing a particular phenotype of the disease. So initially, we assume all mappings are equally accurate. On subsequent iterations however, hospital visits are classified as representing certain disease states thereby revealing some diagnosis codes as being more accurate then others. The accuracies of diagnosis code mappings, once hospital visits are classified, are determined as follows.

From the classified hospital visits, the frequency of each diagnosis is calculated for each state. The frequency vectors are compared with the state vectors from symptoms to determine the accuracy of each diagnosis code in the prediction of disease state. A vector containing the frequency of diagnosis codes appearing in the hospital visits and a symptom's state vector are incomparable. For comparison, the frequency vector is thresholded to transform it into vector of 0s and 1s. The threshold was calculated as the average number of non-zero counts per stage:

$$\text{Threshold} = \frac{\sum_{i\in|S|} x_i}{\sum_{i\in|S|} \theta(x_i)}, \qquad \theta(x) = \begin{cases} 0, & if\ x = 0 \\ 1, & if\ x > 0 \end{cases}$$

where x represents the frequency counts for state i, S is the number of states, and $\theta(x)$ is the indicator function. Each position of the frequency vector is changed to a bit, where a 1 is assignment to frequencies greater than the threshold and 0 otherwise. For example, the frequency vector [1,9,1,1] would provide a threshold of 3 ((1+1+9+1) / (1+1+1+1)), and the frequency vector after thresholding would be [0,1,0,0]. All frequency data is now on the order of a string of bits. The corresponding vectors are termed "sample vectors."

From the sample and symptom vectors, we can determine an accuracy score for each diagnosis code. Because each vector position has a binary choice, we consider the accuracy score as the following. True positives and true negatives add a score of +1, while false positives add –1. The score is normalized by the total number of states *k*. Formally, accuracy is defined as:

$$\text{Accuracy} = \frac{\sum_{i\in|S|} \phi(t_i - f_i)}{|S|}, \quad \phi(x) = \begin{cases} -1, & if\ x = -1 \\ 0, & if\ x = 1 \\ +1, & if\ x = 0 \end{cases}$$

where t and f representative of the positions in the state and thresholded frequency vectors, respectively, and $\phi(x)$ a bit function. By the above definition, the accuracy of any particular feature must reside in the range [(*k*-1)/*k* , 1]. The lower bound is dependent on the nature of assigning a state pattern for a symptom. there must be a minimum of one position in the vector that is defined as 1.

2.5 Patient Visits Mapped to Disease States (Step IV)

The question becomes, "How do we use these accuracy values to relate distinct hospital visits to disease states?" Consider the feed forward schematic depicted in

Figure 4. The useful information from a hospital visit, diagnosis codes (though procedure and discharge status codes could also be used), denoted as *d*, are mapped to their respective symptoms, denoted as *s*. Rather than feed the raw frequency vector associated with a symptom, the accuracy value of each constituent diagnosis code is used. This is represented by w_i, which recognizes the degree to which the hospital visit belongs to the disease state based on the appearance of the diagnosis. At each symptom, the maximum accuracy *max* w_i, a scalar value, is used to scale the symptom's state vector. The scaling is simple multiplication of a vector by a scalar, which converts the state vector into a vector of 0s and the max weight. For example, let s_i have a state vector [0,1,1,0] and the corresponding set of diagnoses weights that fed forward to s_i be [0.3, 0.4, 0.3, 0.7]. Because the max weight is equal to 0.7, the weighted state vector is [0, 0.7, 0.7, 0].

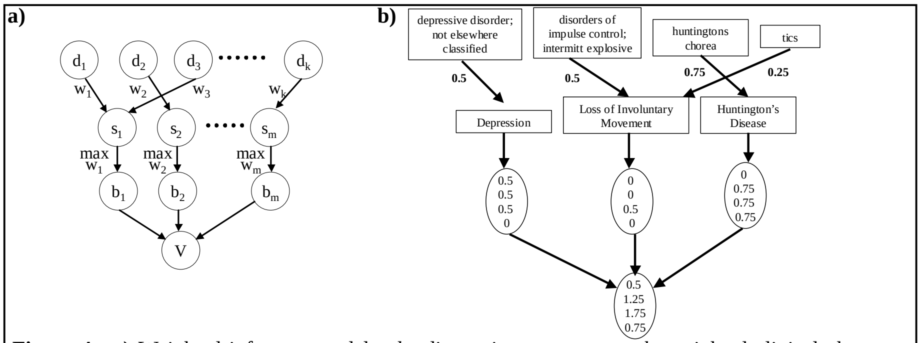

Figure 4. **a)** Weighted inference model. d_i= diagnosis, s_i= symptom, b_i=weighted clinical phenotype vector, *V* = final weighted clinical phenotype vector for a hospital visit. **b)** An example relating a hospital visit to disease states.

The weighted vectors, noted as b_i in Figure 4a, for each symptom are then combined via vector addition. Since the vectors are of the same dimension, with each position corresponding to the same state of the disease, the vector addition results in a vector the same size as the number of disease states. The final vector *V*, is a weighted score of the certainty in each state that a particular hospital visit exhibits. This score is determined for each hospital visit independently.

2.6 Temporal Constraints for Optimizing Disease State Alignments (Step V)

Now that each hospital visit has been converted into a weighted vector of disease states, we must determine how to relate visits belonging to a single patient with time dependent aspects. There exist some diseases that have a defined progression pattern, or one that may be inferred. One example of such progression inference has been demonstrated with partially observable Markov decision processes for studying heart disease.[11] Other diseases, such as Huntington's disease, are currently untreatable and therefore have a direct progression towards death. This section of the algorithm is not necessary for clinical information inference of diseases with an unknown progression status, such as cystic fibrosis. However, because progression of disease is an issue that helps define the current state of some diseases, time constraints must be taken into account.

Consider the profile depicted in Figure 5a. Each row corresponds to a state *s* of the disease. We are attempting to determine the optimal alignment of states for this profile. If we assume that the max value in each *v* corresponds to the actual state, then the predicted progression would be as depicted in Figure 5b. However, if the disease could only have a forward progression without remission, we would have to consider the maximum sum of single vector positions under this constraint. The result is depicted in Figure 4c. Yet, to prevent impossible stage alignments for longitudinal medical profiles, we must utilize knowledge about the disease.

a)	$\mathbf{v_1}$	$\mathbf{T_1}$	$\mathbf{v_2}$	$\mathbf{t_2}$	$\mathbf{v_3}$	$\mathbf{t_3}$	$\mathbf{v_4}$	$\mathbf{t_4}$	$\mathbf{v_5}$		**Alignment**				
$\mathbf{s_0}$	5.8		1.3		0.8		0.5		2.7	b)	0	2	1	2	3
$\mathbf{s_1}$	2.5	1.5	3.5	1	5.1	1	1.3	3	1.2	c)	0	1	1	2	3
$\mathbf{s_2}$	0.9		6.2		1.3		1.8		3.4	d)	2	2	2	2	3
$\mathbf{s_3}$	0		5.0		1.8		0.8		7.5						

Figure 5. a) Sample patient profile with the time elapsed shown between each vertical state vector. The time elapsed is in years. **b-d)** Various statement alignments as the result of varying degrees of time constraints. In **b)** max cell values are assumed to be stage of disease; in **c)** linear forward progression constraint is enforced, and in **d)** time dependency for each stage is considered.

Many different state sequences could have given rise to the observed sets of diagnoses. For example, the state sequences could have been generated from the underlying states (1,1,1,1,1), (2,2,2,2,2), (2,2,3,3,3), or (1,1,2,2,2), though there would be many more possible alignments of the sequence. The difference between these alignments is that they would present diagnoses at each visit with different probabilities. We are interested in the path through the sequence alignment that provides the highest probability by considering the most probable state sequence given the set of compressed tuple vectors *V*, the set of time constraints *T*, and a state path through the profile.

The diagnosis-state mappings are updated from the hospital visit classifications and then steps II through IV of the algorithm repeat until no further refinement in the classification of hospital visits is realized. Metrics were defined to determine convergence.

3 Results on Huntington's Disease

Materials included hospital discharge data from the State of Illinois, for the years 1990 through 1997. There were approximately 1.3 million hospital discharges per year. Collection information has compliance with greater than 99% of discharges occurring in hospitals in the state.[8] As a sample set, a Huntington's disease registry from Rush Presbyterian Hospital of Chicago was used. The registry consisted of demographic information and the age of onset of the disease for each listing.

Longitudinal medical profiles were constructed as described in previous reports.[8] Profile construction was performed with an estimate of 100% uniqueness and identifiability of individuals based on {*ZIP*, *date of birth*, *gender*}. The resulting profiles were crossed with the registry. The resulting join, yielded a sample of 22 individuals, with a total number of 69 hospital visits.

The literature review provided a list of symptoms related to four stages. Clinically, there are three stages of the disease known to exist; an early stage, middle stage, and late stage, as well as the asymptomatic period of the disorder. It is worth noting there are two types of Huntington's disease that have different progression rates. One type is a juvenile onset that presents before the age of 20, while the other is normal adult presentation above the age of 20. Furthermore, the disease has an untreatable forward progression toward death. There is no remission, thus you could not backtrack from the middle to the early stage of the disease.

Step I of the approach involves mapping symptoms to phenotype states. Based on literature review, a list of 36 symptoms was constructed with each symptom mapping to any of 4 possible stages of the disease. Diagnosis codes and discharge status codes were identified and then mapped to the generalized diagnoses based on the methods described in section 2.2. For Step II, partitioning the symptom model resulted in a total of 45 symptoms.

Table 2. Comparison of Models for Generalizing Diagnoses

Model	**Number of Nodes**	**Number of States Encountered**
Diagnoses (direct)	156	477
Generalized Codes	60	791
Text Semantics	8	975
Symptoms	45	752

As shown in Table 2, the diagnosis model, which maps each diagnosis directly to its disease state, resulted in 477 states being encountered. Of the generalized models, we found the symptom model to be more specific than the models resulting from generalizing codes or using textual descriptions. Henceforth, we continue with the symptom model.

Step III of the approach involves learning corresponding weights for the diagnosis and discharge status codes. Step IV of the approach involves automatically classifying each patient visit into a known disease state. Finally, with each hospital visit independently classified as exhibiting a state of the disease, we had to align visits pertaining to the same patient in order to respect the temporal constraints specific to Huntington's Disease. The method used is described in the next paragraphs.

For Huntington's disease, the expected length of time is approximately 5 years per stage for the adult type, and 3 years for the juvenile type.[10] Furthermore, there is a linear progression of the disease. Once the patient reaches stage (or state) 2, remission to stage 1 is not possible. For the adult type, time up to the age of onset is defined as stage 0, from onset up to five years afterwards defines stage 1, five to ten years defines stage 2, greater than ten years defines stage 3.

Based on knowledge about the particular disease, we can construct a set of rules governing the time-dependency of the disease that overlaps the stages in time to account for transitions in the disease as well as in age reporting. The noted ranges are: 1 to 6 years for stage 1, 4 to 11 years for stage 2 and 9 to 15 years for stage 3.

Based on these constraints, we predicted the age of onset for each patient in the following manner: list the times for each visit as an inequality in the time ranges noted. This provides a set of inequalities, one inequality for each visit. Expand the list by reporting the time lapses between visits. Solve the resulting set of inequalities to get time bounds and then modify hospital visit classifications accordingly. Final results, converging after three iterations of the algorithm, appear in Figure 6.

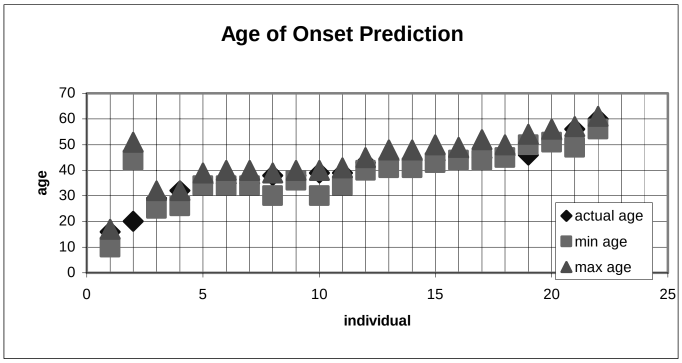

Figure 6. Results from Huntington's data. Age of onset is accurately predicted in 20 of the 22 cases.

The relationship between the age of onset and the trinucleotide repeat size has been shown to have an inverse relationship. The relationship has an r^2 regression value of 0.73 as noted in previous works.[12] The equation used to determine the relationship between age of onset and CAG repeat size is ln(age of onset) = 5.4053 - 0.0377*(trinucleotide repeat size). Out of the Rush Presbyterian dataset, there were 3 subjects that we knew the repeat size for. Predictions of the repeat size yielded matches.

4 Discussion

Our approach is biased by the initial disease and symptom mappings, but such bias is analogous to having incorrect knowledge in a knowledge base or incorrect classifications assigned in training data. The iterative nature of our approach further refines its models to better fit the data but cannot always overcome initial bias.

The methodology described above is general enough to be compatible with many single gene disorders. For each of these diseases, the number of states will be dependent on the number of clinical types.

References

1. F.M. De La Vega, M. Kreitman, and I.S. Kohane. “Human genome variation: linking genotypes to clinical phenotypes” In: *Pacific Symposium on Biocomputing 2001*, R.B. Altman et.al (Eds.) (World Scientific, Singapore, 2001).
2. M.R. Knowles, K.J. Friedman, and L.M. Silverman, “Genetics, Diagnosis, and Clinical Phenotype” Ed. J.R. Yankaskas and M.R. Knowles. (Lippincott-Raven, Philadelphia, 1999).
3. R.R. Brinkman, et al., “The likelihood of being affected with Huntington disease by a particular age, for a specific CAG size.” *Am. J. Hum. Genet.* **60**. 1202-1210 (1997).
4. A.R. La Spada. “Trinucleotide repeat instabilitiy: genetic features and molecular mechanisms.” *Brain Pathol*. **7**. 943-963 (1997).
5. B.G. Buchanan, et al. "Heuristic DENDRAL: a program for generating explanatory hypotheses in organic chemistry." In *Machine Intelligence 4,* B. Meltzer et. al (Eds.) (Edinburgh University Press, Edinburgh, 1969).
6. E.H. Shortliffe. *Computer-Based Medical Consultations: MYCIN*. Elsevier/North-Holland, Amsterdam, London, New York (1976).
7. G. Tesauro. "Practical issues in temporal difference learning" *Machine Learning*, 8 (3-4):257-277 (1992).
8. B.A. Malin, and L.A. Sweeney. “Determining the Identifiability of DNA Database Entries” *Proc AMIA Symp*. 537-541 (2000).
9. T.M. Mitchell, Machine Learning (McGraw-Hill, New York, 1997).
10. O. Quarrell, Huntington’s Disease: The Facts. (Oxford University Press, New York, 1999).
11. M. Hauskrecht, and H. Fraser. “Modeling Treatment of Ischemic Heart Disease with Partially Observable Markov Decision Processes” *Proc AMIA Symp*. 538-532 (1998).
12. S.E. Andrew, et. al. “The relationship between trinucleotide (cag) repeat length and clinical features of Huntington’s disease” *Nature*. **4**. 398-403 (1993).