(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
24 October 2002 (24.10.2002)

PCT

(10) International Publication Number
WO 02/084430 A2

(51) International Patent Classification⁷: G06F

(21) International Application Number: PCT/US02/10894

(22) International Filing Date: 8 April 2002 (08.04.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/282,751 10 April 2001 (10.04.2001) US

(71) Applicant and
(72) Inventor: **KELLER, Paul** [US/US]; 4585 Liberty Rd., South Euclid, OH 44121 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZM, ZW.

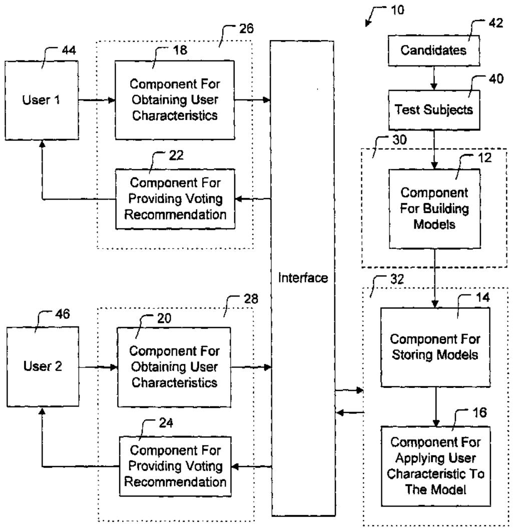
(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declaration under Rule 4.17:
— *of inventorship (Rule 4.17(iv)) for US only*

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) **Title:** VOTER ADVISORY SERVICE


10
44
User 1
26
18
Component For Obtaining User Characteristics
22
Component For Providing Voting Recommendation
46
User 2
28
20
Component For Obtaining User Characteristics
24
Component For Providing Voting Recommendation
Interface
42
Candidates
40
Test Subjects
30
12
Component For Building Models
32
14
Component For Storing Models
16
Component For Applying User Characteristic To The Model

(57) **Abstract:** One aspect of the present invention relates to using data from test subjects to develop a model encoding a generalized relationship between user characteristics and candidate preferences. The test subjects can take extensive amounts of time to investigate candidates, whereby the model can relate user characteristics to informed candidate preferences. Another aspect of the invention relates to applying the model to provide users with voting recommendations, which relate to how the users would vote if the users were to undertake an extensive investigation of the candidates such as undertaken by the test subjects. According to a further aspect of the invention, users are divided based on their characteristics among clusters, which can be defined in such a way that cluster identity is highly predictive of candidate preferences. Clusters can be used to encourage block voting, form political parties, and to conduct political research.

VOTER ADVISORY SERVICE

Technical Field

The present invention generally relates to an automated system for assisting voters in using their votes to obtain the kind of government they want.

Background of the Invention

Most people would agree that "one person, one vote" is a fine principal for a democratic government, but for government to be representative of the will of the people more is required. Generally, it is thought that voters need to have accurate information about issues and candidates. The U.S. Constitution partially addresses this matter by providing for freedom of the press. Public education is also thought to facilitate democracy. In spite of these laudable measures, there are still obstacles to obtaining representative government in the U.S. and others countries that favor democratic rule.

One obstacle is the shear size of the world. In major elections, on a percentage basis, very few voters will ever have the chance to personally meet any given candidate. The enormous burden of informing voters about candidates and issue generally falls to the candidates themselves. The result is an undue advantage to candidates who are well funded and to candidates who have broad name recognition for reasons that are often totally unrelated to their qualifications for office.

Another obstacle is the number of races in which a voter may be expected to cast a vote. In the U.S., it is not uncommon for a voter to be faced with a ballot giving the voter an opportunity to vote in twenty of more races. It is often the case that for many of the races the voter is unfamiliar with any of the choices, much less all of the choices.

A further obstacle is the information explosion. While newspapers and Internet resources can be helpful in disseminating information, they create the problem of information overload. People do not have time to sift through all the available information and often have difficulty finding reliable sources among a plethora of sources that may be unreliable.

A historical solution to these problems has been the formation of political parties that back no more than one candidate for each race. Political parties are a limited solution as, like individual candidates, they have the financial burden of informing or persuading voters and often find it necessary to choose candidates with high name recognition. Whatever the limitations of political parties, many voters have long felt dissatisfaction with them. In the United States, two political parties have come to dominate. Voters often feel that neither of these parties represents their view. Voters are hesitant to vote with a third party because their preference between the two major parties will not be expressed if they vote with a third party. The fewer people who are planning to vote for a third party candidate, the fewer people who are willing to switch their vote to the third party candidate. Hence, third party candidates may not be elected even if the majority of voters would prefer them.

Like political parties, certain special interest groups provide voters with recommendations. For example, labor unions and abortion rights groups may express their support for certain candidates, or recommend certain candidates to their constituents. However, few voters can find a special interest group that is representative of their views across the spectrum of political issues.

Another solution is to gather and summarize information about various candidates and provide this information to the voters. For example, the League of Women Voters has produced a publication providing information about various candidates. Such publications can be helpful, but most voters do not find the time to review such information summaries. Furthermore, providing adequate funding for such organizations while maintaining their neutrality is difficult.

For the foregoing reasons, there has been a long felt need for the present invention.

Summary of the Invention

The following presents a simplified summary of the invention in order to provide a basic understanding of some of its aspects. This summary is not an extensive overview of the invention and is intended neither to identify key or critical elements of the invention nor to delineate its scope. The sole purpose of this summary is to present some concepts of the invention in a simplified form as a prelude to the more detailed description that is presented later.

One aspect of the present invention relates to using data from test subjects to develop a model encoding a generalized relationship between user characteristics and candidate preferences. The test subjects can take extensive amounts of time to investigate candidates, whereby the model can relate user characteristics to informed candidate preferences.

Another aspect of the invention relates to applying the model to provide users with voting recommendations. The voting recommendations are based on candidate preferences predicted for the users by the model. Characteristics of the users are applied to the model, whereby the model predicts, with a high level of confidence, what the users' candidate preferences would be if the users were to undertake an extensive investigation of the candidates such as undertaken by the test subjects.

According to a further aspect of the invention, users are divided based on their characteristics among clusters, which can be defined in such a way that cluster identity is highly predictive of candidate preferences. Clusters can be used to encourage block voting, form political parties, and to conduct political research.

Other advantages and novel features of the invention will become apparent from the following detailed description of the invention and the accompanying drawings. The detailed description and drawings provide certain illustrative examples of the invention. These examples are indicative of but a few of the various ways in which the principles of the invention can be employed.

Brief Description of the Drawings

Fig. 1 is a high level schematic of a computer system according to one aspect of the present invention.

Fig. 2 is a high level schematic of a method according to another aspect of the present invention.

Fig. 3 is an illustration of a decision tree according to a further aspect of the present invention.

Fig. 4 is a schematic illustration of a data structure according to yet another aspect of the present invention.

Fig. 5 is a schematic illustration of a Bayesian belief network according to a still further aspect of the present invention.

Fig. 6 is an example of an operating environment for the present invention.

Detailed Description of the Invention

Fig. 1 is a high level schematic of a computer system 10 provided by the present invention. Computer system 10 includes model-building component 12, model-storing component 14, model-applying component 16, input-obtaining components 18 and 20, and recommendation-providing components 22 and 24. These components are shown as being divided among computer systems 26, 28, 30, and 32. This division is optional as described more fully below.

Model-building component 12 obtains data on test subjects 40, which includes characteristics of the test subjects and preferences of the test subjects relating to candidates 42. The preferences of test subjects 40 are generally based on information about candidates 42 that is more extensive than that which can be conveniently obtained by users, such as users 44 and 46. Model-building component 12 employs the data on test subjects 40 to develop one or more models encoding a generalized relationship between user characteristics and candidate preferences. A generalized relationship is one that can produce different outputs for different inputs. Typically, the models take into consideration a plurality of user characteristics. The models can be stored on computer-readable media of model-storing component 14, from which the models are conveniently accessed by model-applying component 16.

Input-obtaining components 18 and 20 obtain characteristics of users 44 and 46 and provide this information to model-applying component 16. Model-applying component 16 applies these characteristics to the models and provides voting recommendations for users 44 and 46 based on the results. The recommendations are provided to users 44 and 46 through recommendation-providing components 22 and 24.

As used in this application, a "computer system" is a structure comprising one or more components. A "component" is a structure comprising computer hardware and/or software. For example, a component can be, but is not limited to, a computer memory encoded with software instructions or a computer configured to carry out specified tasks. By way of illustration, both an application program stored in computer-readable memory and a server on which the application is running can be components. Due to the nature of components, multiple components can be intermingled and are often not separate from one another. Computer systems can also be intermingled and inseparable.

The components can be organized among the computer systems in various ways and can shift among the computer systems. For example, where model-storing component 14 and model-applying component 16 reside on computer system 32, computer system 32 is generally provided with an input-obtaining component. The input-obtaining component can take the form of software instructions for requesting and uploading user characteristic. A second input-obtaining component can subsequently appear on computer system 26 as a result of a download. For example, an ".html" file providing a graphical user interface for obtaining user characteristics can be downloaded from computer system 32 to computer system 26 by a web browser located on computer system 26. An input-obtaining component could also be uploaded to computer system 26 from a floppy disk or other removable media.

Computer systems 26 and 30 can be client computers, such as personal computers, associated with users 44 and 46. Computer system 32 can be a server accessible to computer systems 26 and 30 over a network interface. On the other hand, the components can be arranged among one or more computer systems in any suitable fashion. For example, all components could be placed on one computer system. In addition, two or more of the computers systems, such as computer systems 26 and 32, can be viewed as one computer system. Thus, the division of computer system components among computer systems is generally not a limitation of the invention.

Model-storing component 14 can reside on computer system 32 where it is accessible to a plurality of clients on various computers. Model-applying component 16 conveniently resides on computer system 32 along with model-storing component 14. However, due to privacy concerns or perceived opportunities for abuse, it can be desirable to avoid uploading user characteristics. In such a case, model-storing component 14 and model-applying component 16 can be provided on client computers 26 and 28. These components can be downloaded from computer system 32 or otherwise provided to the client computer systems.

The recommendation-providing components 22 and 24 may appear first on the computer system that hosts model-applying component 16. Where that computer system is a server, additional recommendation-providing components can be subsequently downloaded to one or more client computers. The voting recommendation can ultimately be provided to the user, for example, through a display on a monitor, through a printout, or over a speaker. In one aspect of the invention, a voting recommendation is provided to a user over an analog telephone transmission. Input can likewise be provided over an analog telephone transmission, with voice or touchtone signals, for example.

The voting recommendations provided by components 22 and 24 are individualized to a user. An individualized voting recommendation is one that depends on user characteristics. A user is generally a natural person, but can also be a group of individuals sharing common characteristics, such as a group or organization. For example, the user could be a political party, a union, a neighborhood association, an abortion rights group, or a religious group.

The models generated by model-building component 12 relate candidate preferences to user characteristics. In one embodiment, the models contain dependencies on two or more user characteristics. In another embodiment, the models contain dependencies on four or more user characteristics. In a further embodiment, the models contain dependencies on eight or more user characteristics.

The use characteristics can be any characteristics that are associated with users and are related, in a predictive sense, to user opinions. For example, characteristics can include demographic information such as age, sex, race, profession, marital status, and place of residence. However, characteristics can also relate more directly to the user's political views. In particular, user characteristics can comprises the user's answers to questions, especially questions related to political views such as the following:

- Are you in favor of abortion rights?
- Do you believe the government should ensure everyone is adequately housed?
- Do you favor the death penalty?
- Would you like to see a ban on handguns?
- Is legislation needed to provide equal rights to homosexuals?
- Do you feel that American markets need better protection from under-priced foreign competition?
- Does the government need to place more emphasis on environmental protection?
- Should marijuana be legalized?
- Are you willing to pay higher taxes if the additional funds go towards education?
- Do you favor school vouchers?
- Do you prefer leaders who are openly and strongly religious?
- Do you think that being State governor is a difficult job?
- Do you think the President is doing a good job?
- In your opinion, should the United States government protect oppressed people throughout the world?

- Should the United States go to war if necessary to protect its economic interests in oil producing Arab states?
- Would you like to see free trade with Cuba?
- Do you believe the government should ensure everyone in the world has enough food?

An input-obtaining component can provide any suitable set of answer choices. For example, the component can provide for "yes" or "no" answers, "yes", "no", or "no opinion" answers, or "strongly agree", "agree", "neutral", "disagree", or "strongly disagree" answers. The later contemplates questions of the form: "state whether you agree or disagree with each of the following statements."

User information is obtained from users and from test subjects. A greater number of characteristics can be obtained for the test subjects than for the users. From this greater number of characteristics, those characteristics that are most predictive of candidate preferences can be selected. The characteristics obtained for the users can be limited to those that are most predictive of candidate preferences, whereby the burden on the users can be kept to a minimum.

Test subjects, like users, are generally natural persons but can also be groups of individuals or organizations. It is advantageous to have a pool of test subjects representing diverse political views, but it is not necessary, or even necessarily advantageous, to have a pool of test subjects with views that are statistically similar to those of the users to whom voting recommendations will be provided.

Model-building component 12 employs test subject candidate preferences. These preferences can be expressed in any suitable form. For example, the preferences can be simple choices among candidates. They can also be ratings of candidates on a scale, e.g., "excellent", "very satisfactory", "satisfactory", "unsatisfactory", "very unsatisfactory", and "horrible". Optionally, these ratings can be converted to a numerical scale for use in calculating utilities, e.g., +10, +5, +2, -2, -5, or -10.

According to one aspect of the invention, the test subjects are well informed about the candidates. The test subjects can become well informed about the candidates by any suitable means. For example, the test subjects can review newspaper articles about the candidates, read candidate position papers, read what special interest groups have to say about the candidates, meet with the candidates, hire detective services, hire legislative research services, and/or hire news research services. The test subjects can also be, or become, informed about issues, job requirements, job responsibilities, authorities associated with positions, and/or other factors that might influence a candidate selection.

In another aspect of the invention, the test subjects are paid to investigate candidates and provide opinions. Paying test subjects has advantages such as providing a greater pool of test subjects, motivating test subjects to engage in thorough investigations, and developing test subjects who have expertise in evaluating candidates.

Test subject pay can be funded in any suitable manner. Options include funding from public agencies, funding from private donations, charging fees for providing voting recommendations, and paid advertisements on web sites providing or explaining voting recommendations. Paid advertisements can be accepted from candidates and/or political parties. Alternatively, advertising can be restricted to preserve the appearance of neutrality.

The candidates are choices users can make during elections. Generally, the candidates are individuals who are running for office. Candidates can also include individuals who might be persuaded to run for office. Where issues are on a ballot, positions on issues can also be candidates. For example, "Yes" to Proposition 13 can be one candidate and "No" to Proposition 13 can be another candidate.

Model-building component 12 generates one or more models encoding generalized relationships between user characteristics and candidate preferences. The output of a model can be any result that provides a suitable basis, at least in part, for a voting recommendation individualized according to user characteristics. A model can relate user characteristics to a choice among several candidates. Alternatively, there can be one model for each candidate, each model providing a measure relating to whether the user would choose the subject candidate. The measure can be, for example, a rating the user would be expected to give the candidate, a probability the user, if informed to the extent of the test subjects, would vote for the candidate, or a value to the user in having the candidate elected.

A value (cost or benefit) to the user in having a candidate elected can be determined from scores or values that test subjects place on the candidate's election. For example, the election of one candidate may, in the opinion of a test subject, warrant a small positive score while the election of another candidate may warrant a large negative score. Values can be used in calculating utilities for voting recommendations, as described more fully below.

In addition to providing a choice among several candidates or a measure relating to choice, a model can provide a confidence level with respect to a choice or measure. The confidence level can take into account uncertainty expressed by the test subjects and/or uncertainty in the data. For example, the test subjects may be asked to express their confidence in their evaluation of a particular candidate and that confidence level can become part of the model output. A model can also provide a confidence level pertaining to the sufficiency of the test subject data as a basis for predicting the preferences of a particular user.

Generally, the one or more models are probabilistic dependency models. For example, the models can be decision graphs, support vector machines, Bayesian belief networks, or neural networks. Some of these models divide users into clusters, with each user belonging to one and only one cluster. Models that divide the users into clusters have special applications, as discussed more fully below.

Fig. 2 is a flow diagram providing a generalized example of a method 200 by which model-building component 12 can operate. Actions 202 and 204 are concerned with obtaining a set of training examples. Method 200 treats these as two actions, but they can take place simultaneously. Action 202 is obtaining characteristics for a set of test subjects. Action 204 is obtaining candidate preferences for the test subjects. It may be desirable to obtain user characteristics for the test subjects before informing them about the candidates and issues to avoid the risk that information about candidates and issues will influence the data used to characterize the test subjects.

If the training data is accurate and complete, the resulting models are generally more reliable. However, probabilistic dependency models can be trained with inaccurate, incomplete or inconsistent data. For example, a probabilistic dependency model can be trained with test subject data for which apparently identical subjects prefer different candidates or for which not all user characteristics were assessed for all test subjects.

Action 206 is selecting a probabilistic dependency model type. For example, decision graphs, Bayesian belief networks, and neural networks are options. For some of these models, such as the network models, structural features may need to be specified. In a neural network, the number of layers and the number of nodes per layer may require specification. On the other hand, part of the model structure can be determined as part of the training process.

The selection of a probabilistic dependency model can limit the relationships between inputs and outputs that can be captured by the model. The set of all possible relationships is referred to as the model space. The model space is chosen so that there is a high probability that either the actual relationship between inputs and outputs or a close approximation thereto is in the model space.

Once a model space is chosen, the next action, 208, is selecting an algorithm with which to search the model space for a model consistent with the training data. The choice of a search algorithm depends on the model type, but even within each model type there are generally a host of choices. In most instances, a complete search of the model space is impractical and there is no guarantee that the model found is the most consistent with the data among those models in the model space. However, commonly used search algorithms are generally sufficient to find a model whose match to the data is reasonably good within the limits of the model space.

Action 210 is applying the search algorithm to find a model consistent with the training data. This is also referred to as training the model. Once the model is trained, or found, it is output in action 212. The output can be a display on a monitor showing the dependencies of the model. However, whether or not the model is displayed, the output is generally recorded on a computer readable media, in model-storing component 14 whereby it is ready for use by model-applying component 16.

According to one aspect of the invention, one or more decision graphs are employed to model the relationship between user characteristics and candidate preferences. Decision graphs are composed of one or more, generally a plurality, of decision nodes and a plurality of leaf nodes. Decision graphs classify users by sorting them down a tree structure from a root decision node to a leaf node. Decision graphs branch at decision nodes, which represents tests of user characteristics. The leaf nodes represent, or relate to, voting recommendations. Scores, confidence levels, and other numerical values can be calculated by averaging among the test subjects sorted to a particular leaf node. Leaf and decision nodes provide a convenient way to describe a decision graph, but a decision graph can also be expressed as a set of rules.

Decision trees are a special case of decision graphs. In the general case of a decision graph, two or more decision nodes can have branches converging on one node. In the special case of a decision tree, each node has no more than one directly descending parent node.

Fig. 3 gives an example of a decision tree 300 produced by model-building component 12. Decision tree 300 contains dependencies on answers to five yes-no questions. If the dependencies are on multinomial variables, the decision nodes can provide multiple branches. However, even with multinomial variables, all branches can be kept binary. For example, if a variable can be A, B, or C, there can be a node that branches one way if the variable equals A and the other way if the variable equals B or C. There may be an advantage in restricting the decision nodes to binary branching.

If the dependencies are on continuous variables, the decision nodes can compare the variables to critical values, for example: take the first branch if greater than or equal to 2, otherwise take second branch. Critical values can be determined as part of the decision tree learning process. The numbers in parenthesis following each test give hypothetical numbers of test subjects sorted down the corresponding branch. Decision tree 300 relates to a single candidate, and the shaded area of the rectangle at each leaf node represents a confidence level or probability that the user would vote for the candidate if the voter were informed to the same extent as the test subjects.

Decision graphs produced by model-building component 12 can be stored in computer-readable medium. A suitable data structure includes first data fields containing data representing user characteristics to test, second data fields corresponding to the first data fields and containing data representing values against which to compare the attributes, third data fields containing data representing outcomes, and fourth data fields facilitating determination of relationships among instances of the first, second, and third data fields.

Fig. 4 provides an example of a data structure 400 suitable for storing a decision graph produced by model-building component 12. In this example, the CHARACTERISTIC TYPE and CHARACTERISTIC NAME correspond to the first data field and are associated with decision nodes. CHARACTERISTIC TYPE can be the user characteristic's data type (Boolean, integer, real number, etc.) and CHARACTERISTIC NAME can be a string, index, or pointer that can be used to identify the user characteristic to be tested. A data structure of the invention generally includes dependencies on a plurality of user characteristics.

At each decision node there are two or more branches. To allow the number of branches off a decision node to vary, data structure 400 uses a critical list header to relate an arbitrary number of compare values to each decision node. The CRITICAL LIST HEADER can be, for example, the first element in an array, a pointer to an array, or a header element for a linked list. The list elements include COMPARE VALUEs, which correspond to the second data field. In this example, additional fields stored adjacent to second data fields include data fields identifying a compare operator (<, ≥, =, etc.) and data fields identifying the node to branch to if the test is satisfied. BRANCH NODE TYPE identifies whether the branch is to a decision node or a leaf node. BRANCH NODE POINTER is an index or pointer that identifies the branch node.

SCORE corresponds to third data fields and to leaf nodes. In this non-limiting example, SCORE indicates the probability, based on the training data, that the voter would actually prefer a particular candidate if informed to the extent of the test subjects.

ROOT POINTER identifies the decision node at the root of the decision graph. A root pointer is not a necessary part of the data structure. For example, a convention can be adopted whereby the first data element in the data structure is the root node. In general, there are many ways to represent decision graph structural relationships. In this example, ROOT POINTER, CRITICAL LIST HEADER, and BRANCH NODE POINTER are all fourth data fields representing decision tree relationships among the first, second, and third data fields.

Suitable algorithms for building decision trees include the ID3 algorithm, the C4.5 algorithm, and Bayesian learning algorithms. Most decision tree building algorithms use a top down greedy approach to search through the universe of all possible decision trees for one that accurately classifies examples. The algorithms begin by asking which attribute should be tested first and answer the question by selecting the attribute that, in and of itself, best classifies the training examples in a statistical sense. A branch is created for each possible value of the attribute and a new node is placed at the end of each branch. The algorithm repeats at each of the new nodes using only the training examples that would be sorted to that node.

Where an attribute is a continuous variable, part of the process of selecting the attribute that best classifies the data is selecting a value against which to compare the attribute. This can be accomplished, for example, with a gradient search starting from several randomly chosen initial values. The same variable can be tested at several decision nodes, with the value against which the variable is compared differing from one node to another.

When building a decision tree, steps are taken to avoid over-fitting the data. When data is over-fit, the model begins to capture random variations or noise that is unique to the training data. Over-fitting degrades the performance of the model when applied to items outside the training set. Over-fitting is avoided by either limiting the size of the tree or pruning the tree after its initial growth. In either case, the approach to avoiding over-fitting the data can be based on one or more of the following: a distinct set of training examples to evaluate the utility of certain branches; a statistical test to determine whether a particular branch is likely to improve the model fit outside of the training set; or an explicit measure of the complexity of a tree, whereby nodes are removed or avoided to limit the complexity.

As mentioned, a Bayesian algorithm can be employed to learn a decision tree. A Bayesian algorithm for learning decision trees involves assigning scores to various possible tree structures. For example, a Bayesian algorithm can proceed as follows:

1. Begin with one leaf node.
2. Score the current tree structure.
3. For every possible decision tree that can be generated by replacing a leaf node with a binary split of the data based on one of the user characteristics:
    4. Calculate a score for the possible structure.
5. If the best score from step 4 is better than the current score, make the corresponding possible structure the current structure and go to step 2.
6. Return the current structure.

For discrete variables, binary splits of the data are constructed by making one branch for a particular variable value and another branch for all other possible values for that variable. Binary splits for continuous variables can be accomplished, for example, by considering a finite set of possible split values, or by conducting a gradient search.

In a Bayesian algorithm, the score is the posterior probability, or an approximation thereto, of the tree structure being correct given the observed data. The posterior probability is given by:

$$p(T^h \mid D) = c \times p(D \mid T^h) p(T^h)$$

where Th is the hypothesized tree structure, D is the observed data, and c is a constant that is independent of tree structure and can therefore be ignored. p(D|Th) is the probability of observing a set of data given a particular tree structure. The data-independent probability of various tree structures, p(Th), can be taken as one (all structures equal probable) or can be given some functionality that favors simple trees. For example, p(Th) can be given by:

$$p(T^h) = \kappa^n$$

where n is the number of leaf nodes and κ is a number such that $0 < \kappa < 1$.

The probability of observing a set of data given a particular tree structure, p(D|Th), is taken as the product of the probabilities of observing each of the individual data points. The probability of an individual data point is determined by sorting it down the tree to a leaf node. The probability for the data point can be taken as the fraction of all data points sorted to that leaf node that have the same outcome as that observed for the data point in question. Alternatively, an MAP method, such as Dirichlet priors, can be employed to generate probability estimates for particular observations.

According to another aspect of the invention, one or more Bayes belief networks are employed to model the relationship between user characteristics and candidate preferences. For example, a naïve Bayes model can be employed. In a naïve Bayes model, there is a single hidden (unobserved variable) that is predictive of both user characteristics and candidate preferences. The hidden variable can be referred to as a cluster variable, as it divides users into mutually exclusive groups called clusters. In a naïve Bayes model, the user characteristics and candidate preferences are treated as conditionally independent, given the cluster identity (the hidden variable value). The number of clusters can be determined as part of the learning process by learning the best parameters for models with different numbers of clusters and selecting the model with the number of clusters that best fits (is probabilistically most consistent) with the data. To apply the naïve Bayes model, user characteristics are used to infer (or infer the probability distribution for) the cluster variable. For each cluster, there is a result relating to a candidate preference.

As an example of how the naïve Bayes model works, the case will be considered where the user characteristics, c1, c2, . . ., cm, are either one for yes or zero for no and the results cm+1, cm+2, . . ., cn are also either one for yes of zero for no. In this case, the model has parameters θij that are the probabilities that the ci are one given that the hidden variable value is hj. The model has additional parameters giving the results as functions of hj. According to the model, the probability of observing the set of user characteristics and results c1, c2, . . ., cn given that the hidden variable has value hj, P(c1, c2, . . ., cn | hj), is given by:

$$P(c_1,c_2,...,c_n \mid h_j) = \prod_{i=1}^{n}\left(\theta_{ij}c_i + (1-\theta_{ij})(1-c_i)\right) \qquad (1).$$

Applying Bayes theorem, the probability that a user belongs to cluster hj is given by:

$$P(h_j \mid c_1,c_2,...,c_n) = \frac{p_j P(c_1,c_2,...,c_n \mid h_j)}{\sum_{i=1}^{N_x} p_i P(c_1,c_2,...,c_n \mid h_i)} \qquad (2)$$

where pj is the fraction of users falling into cluster hj and NX is the number of clusters. The pj are determined from the training data as part of the training process.

In applying the naïve Bayes model, the result for a user can be taken as the average over the clusters of the result weighted by the probability that the user belongs to the cluster. For example, if the model provides an expected value for a member of a cluster in electing a particular candidate, the value for a particular user can be taken as the sum of the values given for each cluster weighted by the probabilities that the user belongs to each respective cluster. Where it is desirable to divide individual users among clusters, a user can be assigned to the cluster for which expression (2) is highest.

Figure 5 illustrates another Bayes belief network structure useful in modeling the relationship between user characteristics and candidate preferences. This structure is referred to as a bottleneck architecture. The user characteristics, C1, C2, . . ., CY, are treated as being predictive of a single hidden variable, X. The variable X is the cluster variable. R1, R2, . . ., RZ results of the model and relate to candidate preferences. The single hidden variable, X, can be replaced by several variables without fundamentally changing the model. If there are several hidden variables, there is one cluster for each possible combination of hidden variable values. The number of values for the hidden variable, as well as its relation to the user characteristics, is learned from prior knowledge and/or data. In training the model, the model parameters are tuned with the objective of selecting the parameters, and thus the cluster definitions, that are most effective in predicting the candidate preferences. This contrasts with the naïve Bayes model, wherein the model parameters are fit to the user characteristics in addition to the candidate preferences. As with the naïve Bayes model, the number of clusters is determined by trying different numbers of clusters and comparing the results.

A variety of functional forms, with varying numbers of parameters, can be used to express the relationship between the user characteristics and the probabilities for the hidden variable to take on each of its possible values. An appropriate choice of functional forms depends on the number of user characteristics, the amount of data, and any computational limitations. If the number of user characteristics is small, an unconstrained model can be manageable. An option that is effective even with a large number of user characteristics is a linear regression or Softmax model. Specifically, the probability that the hidden variable has the value hj given the model parameters θij and a user with characteristics ci, p(hj | c1, . . ., cY, θ1j . . ., θnj), is given by:

$$p\left(h_j \mid c_1,\ldots,c_Y,\theta_1,\ldots,\theta_Y\right)=\frac{\exp\left(\sum_{i=1}^{Y}c_i\theta_{ij}\right)}{\sum_{k=1}^{N_X}\exp\left(\sum_{i=1}^{Y}c_i\theta_{ik}\right)} \quad (3).$$

The relationship between the hidden variables and the results is generally unconstrained and depends on the type of result being modeled. Where the result is the probabilities the user would vote for a particular candidate, one parameter, φlj can be sufficient. Where the result is the value to the user of electing a particular candidate, or some other numerical measure, two parameters, a mean and a standard deviation for example, can be employed. Where the result is which candidate the user would be expected to vote for (assuming, as usual, that the user is as well informed as the test subjects) there could be a number of parameters equal to one minus the number of candidates, the parameters representing probabilities for voting for each of the various candidates.

Learning the bottleneck architecture model is the process of finding values for the parameters θij, φlj, and NX that tend to maximize the probability of the candidate preferences observed in the training data:

$$\prod_{i=1}^{N_D}\sum_{j=1}^{N_X}p\left(h_j \mid c_1^i,\ldots,c_Y^i,\theta_{1j},\ldots,\theta_{Yj}\right)p\left(r^i \mid h_j,\varphi_{1j},\ldots,\varphi_{Zj}\right) \quad (4)$$

where ND is the number of training examples, superscripts identify data for particular training examples, and the variable ri is the opinion associated with training example i. Expression (4) reflects the fact that the probability of observing a particular candidate preference equals the sum over all clusters j of probability that the test subject (training example) is in cluster j times probability of the observed candidate preference given that the test subject was in that cluster. The probability for the data set as a whole is the product of the probabilities for each of the independent observations the data set contains.

There are a variety of methods by which the parameters θij, φlj, and NX can be determined (learned) from the data. Suitable methods include gradient descent, stochastic gradient descent, Monte Carlo methods, and EM (expectation-maximization) algorithms. Generally, these are iterative methods that find local maximums for the Expression (4). Therefore, to increase the probability of finding the global maximum, or a better approximation thereto, it is desirable to apply whichever methods is used a plurality of times using random perturbations for the initial values of the model parameters.

An example of a generalized EM algorithm for maximizing Expression (4) can be divided into the following three steps, to be repeated until convergence:

Step 1: for each case i and cluster j calculate:

$$p\left(h^i=h_j \mid r^i,c_1^i,\ldots,c_Y^i,\theta^n,\varphi^n\right)=\frac{p\left(r^i \mid h_j,\varphi^n\right)p\left(h_j \mid c_1^i,\ldots,c_Y^i,\theta^n\right)}{\sum_{k=1}^{N_X}p\left(r^i \mid h_k,\varphi^n\right)p\left(h_k \mid c_1^i,\ldots,c_Y^i,\theta^n\right)}$$

Step 2: find the set of parameters θn+1 to maximize:

$$\sum_{i=1}^{N_D}\sum_{j=1}^{N_X}p\left(h^i=h_j \mid r^i,c_1^i,\ldots,c_Y^i,\theta^n,\varphi^n\right)\log p\left(h_j \mid c_1^i,\ldots,c_Y^i,\theta^{n+1}\right)$$

Step 3; find the set of parameters φn+1 to maximize:

$$\sum_{i=1}^{N_D}\sum_{j=1}^{N_X} p\left(h^i = h_j \mid r^i, c_1^i, \ldots, c_Y^i, \theta^n, \varphi^n\right)\log p\left(R_{k(i)} = r^i \mid c_1^i, \ldots, c_Y^i, \varphi^{n+1}\right)$$

In step 1, probabilities for the various possible states (values) of the hidden variable are calculated for each training example taking into consideration the observed candidate preference. This step is accomplished by direct application of the model using the current parameters estimates. To carry out the second step, the derivatives of the objective function with respect to the model variables are computed and small steps are taken in the direction of the gradient until a local minimum is found. In stepping along the direction of the gradient, a search is usually conducted to find the step size the gives a best, or near best, improvement in the objective function. Step 3 can be carried out with a method similar to that used for Step 2, although with the unconstrained model, a closed form mathematical expression for finding the maximum in Step 3 can often be determined readily. The bottleneck architecture model can be applied to generate candidate preferences for particular users in a similar fashion to that in which the naïve Bayes model is applied.

Model-applying component 16 provides voting recommendations, which are based at least in part on candidate preferences that are predicted by applying user characteristics to the one or more models. A voting recommendation can be any suitable information that comprises or leads to a suggestion that the user might want or prefer to vote for a particular candidate. For example, the voting recommendation can be a simple recommendation that the user vote for a particular candidate, a probability that the user would be satisfied with a particular candidate, or a utility associated with the user voting for a particular candidate. A voting recommendation can be indirect. For example, the voting recommendation can direct the user towards a political party or other group associated with certain candidate choices.

In a simple case, the output of a model is a voting recommendation and model-applying component 16 provides that output directly to the user. In another case, model-applying component 16 compares scores or values provided by several models, one for each candidate, and provides a voting recommendation according to the highest score or value. Where a model divides users into clusters, the model can provide a score for each candidate for each cluster. A recommendation for a particular user can be obtained by determining for each candidate the sum over all clusters of the score the cluster gives the candidate multiplied by the probability the user belongs to the cluster.

Model-applying component 16 can also provide a voting recommendation based on maximizing a utility for a user. The utility of voting for a particular candidate is the benefit of making a correct voting decision weighted by the probability the voting decision is correct less the cost of making an incorrect voting decision weighted by the probability the voting decision is incorrect. Any suitable definition of a correct voting decision can be employed. For example, a voting decision may be considered correct if it is the decision the user would make if informed to the extent of the test subjects.

Model-applying component can take into account additional data, such as data pertaining to the likelihood of various election outcomes. Where available, poll results and/or other data, such as the distribution of characteristic among a pool of users, can be used by model-applying component 16 to relate a user's choice to the likelihood of various election outcomes. Using these likelihood's, the expected utility to the user can be calculated for each possible candidate choice and the maximum utility recommendation determined. In this example, the utility of a vote is the sum over the candidates of the likelihood that the candidate will be elected, given the users vote, multiplied by the value the user is expected to place on electing that candidate. Although the variations in these utilities due to one user's vote are ordinarily very small, the variations are sufficient to provide the basis for a maximum utility voting recommendation.

A maximum utility vote can also be determined assuming groups of users will be influenced by the model's recommendations or similar considerations. For example, the utilities of various votes can be calculated assuming that an entire cluster to which the user belongs will vote as a block.

A maximum utility recommendation can result in recommending the user vote for what would otherwise be the user's second choice to reduce the risk that an even less favored third choice will be elected. Such second choice recommendations can have a particularly high utility when the test subject data shows the user is likely to have a strong dislike for the third choice candidate and other data shows the first choice candidate is unlikely to win.

Where utilities are based on the likelihood of various election outcomes, it is beneficial to take into account as many factors as possible in relating the likelihood's to the available data. For example, when considering poll data, the uncertainty includes not only the uncertainty in measuring current opinion, but also the uncertainty in relating current opinion to election outcomes. Historical data from other elections can be used to help estimate the uncertainty with which polling data predicts the outcome of an upcoming election.

Naïve Bayes models and bottleneck architecture Bayesian belief network models divide users among clusters that are predictive of candidate preferences. These models are examples of classifiers that can be used to divide users among clusters that are predictive of candidate preferences. Any suitable classifier can be employed. Suitable classifiers can be found among a priori rule-based classifiers, such as expert systems, and classifiers based on probabilistic dependency models learned from training data. Classifiers based on probabilistic dependency models include classifiers based on decision graph models, support vector machines, Bayesian belief networks, and neural networks. Classifiers can be trained, and clusters defined, using data pertaining to opinions about a single candidate, a single race, or data pertaining to a plurality of races.

A division of users among clusters that are predictive of candidate preferences can be used to create real political parties or define virtual political parties. Virtual political parties can be used to provide voting recommendations based on an assumption of block voting. The users belonging to virtual political parties can be informed of their characteristics and size. Virtual political parties, particularly large ones, can be encouraged to behave as, or become, real political parties. For example, a group with common views can identify and field candidates or express support for certain political issues.

Clusters can also be used for political research. Initially, voters can be polled for user characteristics, whereby the fraction of the population represented by each cluster can be determined. Subsequently, when data is wanted, a small group of test subjects with members representing the various clusters can be polled with respect to the current issue. The position of the population as a whole can than be projected by averaging the responses over each cluster and weighting the clusters according to fraction of the population that they represent. Such political research can provide accuracy without the expense of large-scale polling.

Political research can take into account the potential effect of bringing certain facts to the attention of voters. For example, if members of two clusters that form a majority of voters can be persuaded to accept a controversial position, such as abolishing the death penalty or mandatory sentencing laws, a candidate might be influenced to take a stance on the issue or a group might be motivated to fund a campaign to bring about the majority position. If members of clusters representing a majority of voters prefer a third party candidate, that fact can be touted to support a campaign by the third party candidate.

Fig. 6 illustrates an exemplary computer system 620 of the present invention. Computer system 620 is a personal or server computer, including a processing unit 621, a system memory 622, and a system bus 623 that couples various system components including the system memory 622 and processing unit 621. Processing unit 621 can be any of various commercially available processors, including dual microprocessors and other multi-processor architectures.

System bus 623 can include a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of commercially available bus architectures. The system memory 622 includes read only memory (ROM) 624 and random access memory (RAM) 625. A basic input/output system (BIOS), containing the basic routines that help to transfer information between elements within the computer 620, such as during start-up, is stored in ROM 624.

The computer 620 further includes hard disk drive 627, magnetic disk drive 628 for reading from and write to a removable disk 629, and optical disk drive 630 for reading a CD-ROM disk 631. The hard disk drive 627, magnetic disk drive 628, and optical disk drive 630 are connected to the system bus 623 by a hard disk drive interface 632, a magnetic disk drive interface 633, and an optical drive interface 634, respectively. The drives and their associated computer-readable media provide nonvolatile storage of data, data structures, computer-executable instructions, etc. for the computer 620. Although the description of computer-readable media above refers to a hard disk, a removable magnetic disk and a CD, other types of media that are readable by a computer, such as magnetic cassettes, flash memory cards, digital video disks, Bernoulli cartridges, and the like, can also be used in the exemplary operating environment. A number of program modules can be stored in the drives and RAM 625, including an operating system 635, one or more application programs 636, other program modules 637, and program data 638.

An operator can enter commands and information into the computer 620 through a keyboard 640 and pointing device, such as a mouse 642. These and other input devices are often connected to the processing unit 621 through a serial port interface 646 that is coupled to the system bus 623, but can be connected by other interfaces, such as a parallel port, game port or a universal serial bus (USB). A monitor 647 or other type of display device is also connected to the system bus 623 via an interface, such as a video adapter 648. In addition to the monitor, computer 620 typically includes other peripheral output devices (not shown), such as speakers and printers.

The computer 620 can operate in a networked environment using logical connections to one or more remote computers, such as a remote server or client computer 649. The remote computer 649 can be a workstation, a server computer, a router, a peer device or other common network node, and typically includes many or all of the elements described relative to the computer 620, although only a memory storage device 650 is illustrated in Fig. 6. The logical connections depicted in Fig. 6 include a local area network (LAN) 651 and a wide area network (WAN) 652. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet.

When used in a LAN networking environment, the computer 620 is connected to the local network 651 through a network interface or adapter 653. When used in a WAN networking environment, the server computer 620 typically includes a modem 654, or is connected to a communications server on the LAN, or has other means for establishing communications over the wide area network 652, such as the Internet. The modem 654, which can be internal or external, is connected to the system bus 623 via the serial port interface 646. In a networked environment, program modules depicted relative to the computer 620, or portions thereof, can be stored in the remote memory storage device. The network connections shown are exemplary and other means of establishing a communications link between the computers can be used.

The present invention is described with reference to components of computer system 620 and to acts and symbolic representations of operations that are performed by the computer 620, unless indicated otherwise. Such acts and operations can be referred to as being computer-executed. These acts and operations include the manipulation by the processing unit 621 of electrical signals representing data bits and result in transformation or programming of data bits represented in the memory systems (including the system memory 622, hard drive 627, floppy disks 629, and CD-ROM 631). The memory locations where data bits are maintained are physical locations that have particular electrical, magnetic, or optical properties corresponding to the data bits.

The present invention is illustrated with respect to a particular computer architecture and particular examples, however, various computer architectures are suitable for carrying out the present invention and fall within the scope of the hereto appended claims. Various aspects of the invention have been described. Obvious modifications and alterations will occur to those of ordinary skill in the art upon reading and understanding the foregoing detailed description. It is intended that the invention be construed as including all such modifications alterations, and equivalents thereof

Claims

The claims are:

1. A computer system for assisting a voter in selecting among candidates, comprising:

a computer system component configured to provide the voter with a voting recommendation based on an application of characteristics of the voter to a model encoding a general relationship between user characteristics and candidate preferences.

2. The computer system of claim 1, wherein the general relationship is a probabilistic dependency model.

3. The computer system of claim 1, wherein the voting recommendation comprises a suggestion of a particular one of the candidates.

4. The computer system of claim 1, wherein the voting recommendation comprises a suggestion that the voter be associated with a particular voting group.

5. The computer system of claim 1, further comprising a computer system component configured to obtain user characteristics through a graphical user interface.

6. The system of claim 1, further comprising:

a computer system component configured to obtain data concerning the characteristics of a group of subjects and their candidate preferences; and

a computer system component configured to employ the data in developing the model.

7. A computer system, comprising:

a first computer system component that learns, from training examples, a probabilistic dependency model for classifying voters according to user characteristics.

8. The computer system of claim 7, wherein the first computer system component learns from training examples that comprise test subject responses to questions and test subject candidate preferences.

9. The computer system of claim 7, wherein the probabilistic dependency model comprises a decision graph.

10. The computer system of claim 7, wherein the probabilistic dependency model comprises a Bayesian belief network.

11. The computer system of claim 10, wherein the Bayesian belief network has a bottleneck architecture.

12. The computer system of claim 7, further comprising:

a second computer system component that applies the probabilistic dependency model to user characteristics to provide one or more candidate recommendations.

13. A computer-readable medium having stored thereon a data structure useful in assisting voters, comprising:

first data fields containing data representing an attribute to test, wherein the attributes represented comprise voter attributes;

second data fields containing data representing outcomes; and

third data fields facilitating determination of relationships among instances of the first and second data fields.

14. The computer-readable medium of claim 13, wherein the relationships among instances of the first and second data fields comprise a probabilistic dependency model.

15. The computer-readable medium of claim 14, wherein the relationships comprise a decision graph.

16. The computer-readable medium of claim 14, wherein the relationships comprise a Bayesian belief network.

17. A method of constructing a model for providing voters with a recommendation among candidates, comprising:

obtaining training data comprising candidate preferences of a group of test subjects and characteristics of the test subjects; and

employing the training data to construct a model encoding a generalized relationship between characteristics and candidate preferences.

18. The method of claim 17, wherein the characteristics comprise responses to a series of questions directed to political views.

19. The method of claim 17, wherein the model comprises a decision graph.

20. The method of claim 17, wherein the model compares a Bayesian belief network model.

21. A method of providing a voter with a recommendation among candidates, comprising:

obtaining a model according to the method of claim 17;

obtaining characteristics of the voter;

applying the model to the characteristics of the voter to obtain model output; and

providing the voter with a recommendation among candidates that is based on the model output.

22. A computer system, comprising:

components that combine information about candidates for office with information about one or more voters to develop and provide individualized voting recommendation for the one or more voters

23. A method of providing advice to one or more voters, comprising:

combining information about candidates for office with information about the one or more voters to develop and provide individualized voting recommendation for the one or more voters.

24. The method of claim 23, wherein the information about the candidates comprises candidate preferences of test subjects.

25. The method of claim 24, wherein the test subjects are paid.

26. The method of claim 25, wherein the test subjects are paid from funds comprising government funding.

27. The method of claim 25, wherein the test subjects are paid from funds comprising private donations.

28. The method of claim 25, wherein the test subjects are paid from funds comprising fees paid by the one or more voters.

29. The method of claim 25, wherein the test subjects are paid from funds comprising advertising on a web site providing or explaining voter recommendations.

30. A method of obtaining a voting recommendation, comprising:

providing a data transmission relating to user characteristics; and

receiving a data transmission relating to a voting recommendation;

wherein the voting recommendation depends on the user characteristics.

31. The method of claim 30, wherein the data relating to user characteristics comprises data relating to at least two distinct user characteristics.

32. The method of claim 30, further comprising receiving a data transmission encoding data relating to a plurality of voting recommendations.

33. The method of claim 30, wherein the data transmissions comprise analog transmissions over a telephone.

34. The method of claim 33, wherein the analog transmissions comprise voice transmissions.

35. A computer system useful in conducting political research, comprising:

a computer system component implementing a model encoding a general relationship between user characteristics and clusters that are predictive of voting preferences.

36. The computer system of claim 35, wherein the general relationship is a probabilistic dependency model.

37. A method of projecting a result applicable to a voting population, comprising:

providing definitions, based on user characteristics, of clusters that are predictive of voting preferences;

polling a group of individual voters for responses that include user characteristics;

identifying the clusters to which the individual voters belong to obtain cluster-by-cluster responses; and

weighting the cluster-by-cluster responses according to a distribution of the voting population among the clusters to project a result applicable to the voting population.

38. The method of claim 37, wherein the individuals are provided with information before polling and the result pertains to the effect of that information.

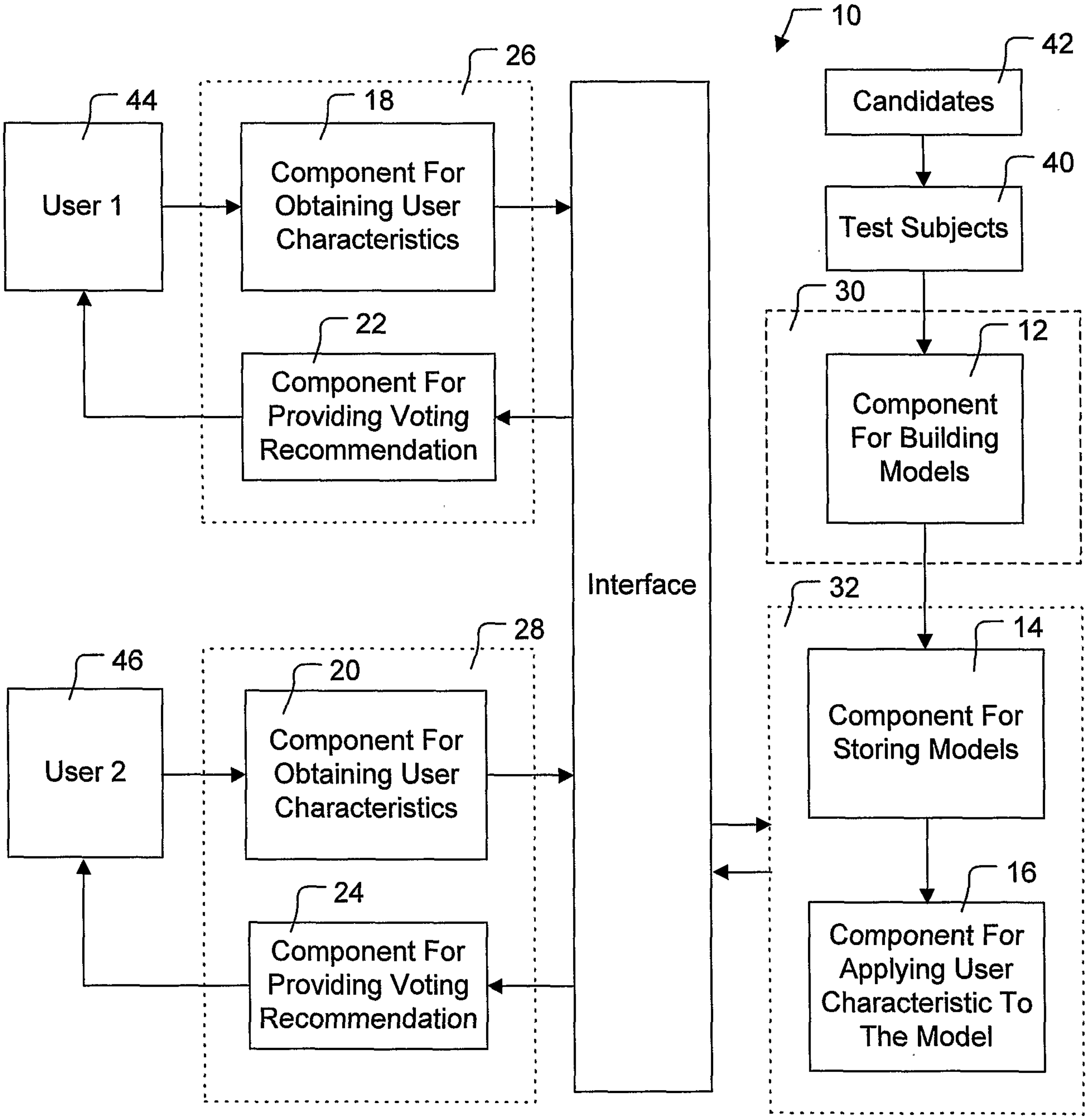

10
42
Candidates
40
Test Subjects
30
12
Component For Building Models
32
14
Component For Storing Models
16
Component For Applying User Characteristic To The Model
Interface
26
44
User 1
18
Component For Obtaining User Characteristics
22
Component For Providing Voting Recommendation
28
46
User 2
20
Component For Obtaining User Characteristics
24
Component For Providing Voting Recommendation

Fig. 1

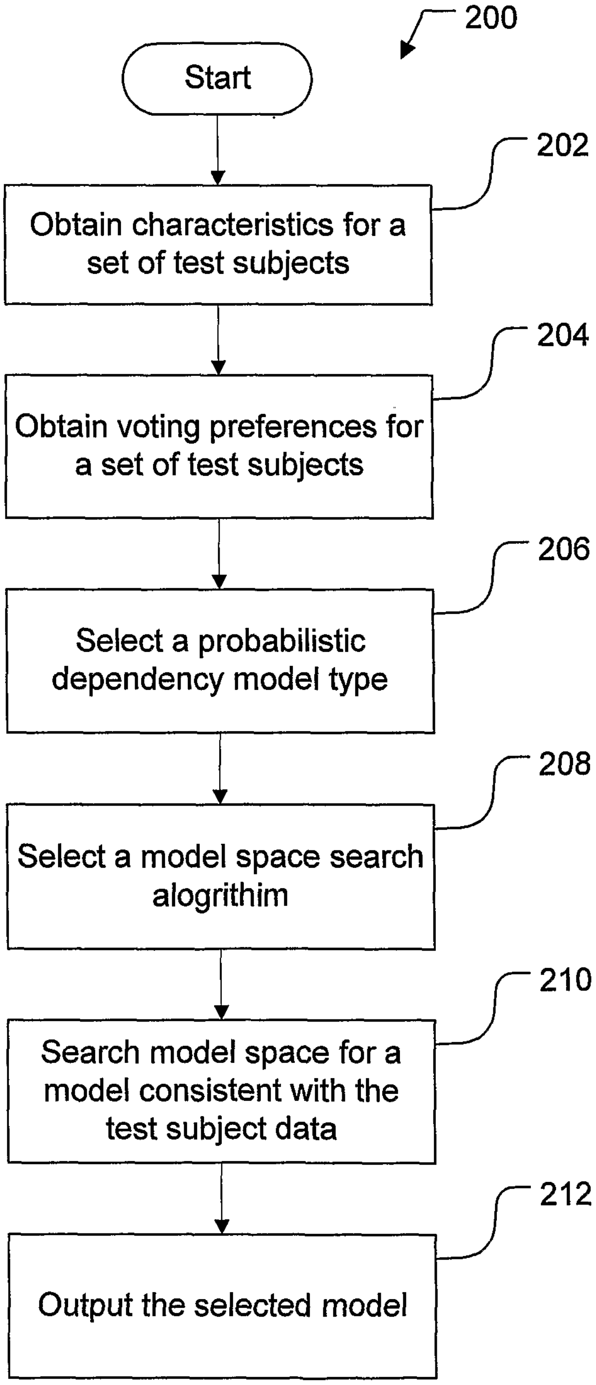

200
Start
Obtain characteristics for a set of test subjects
202
Obtain voting preferences for a set of test subjects
204
Select a probabilistic dependency model type
206
Select a model space search alogrithim
208
Search model space for a model consistent with the test subject data
210
Output the selected model
212

Fig. 2

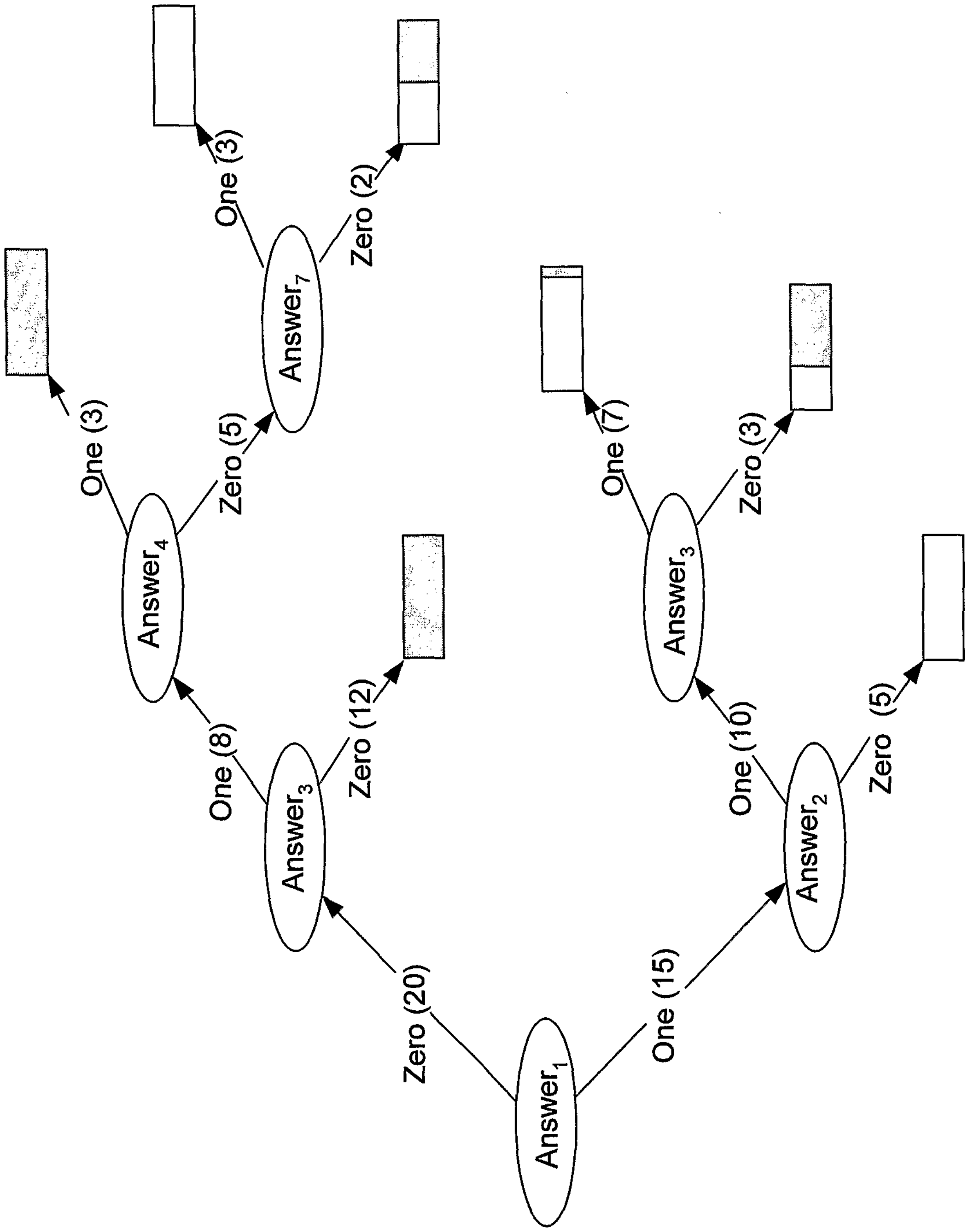

Answer$_1$
Zero (20)
One (15)
Answer$_3$
Answer$_2$
One (8)
Zero (12)
One (10)
Zero (5)
Answer$_4$
Answer$_3$
One (3)
Zero (5)
One (7)
Zero (3)
Answer$_7$
One (3)
Zero (2)

Fig. 3

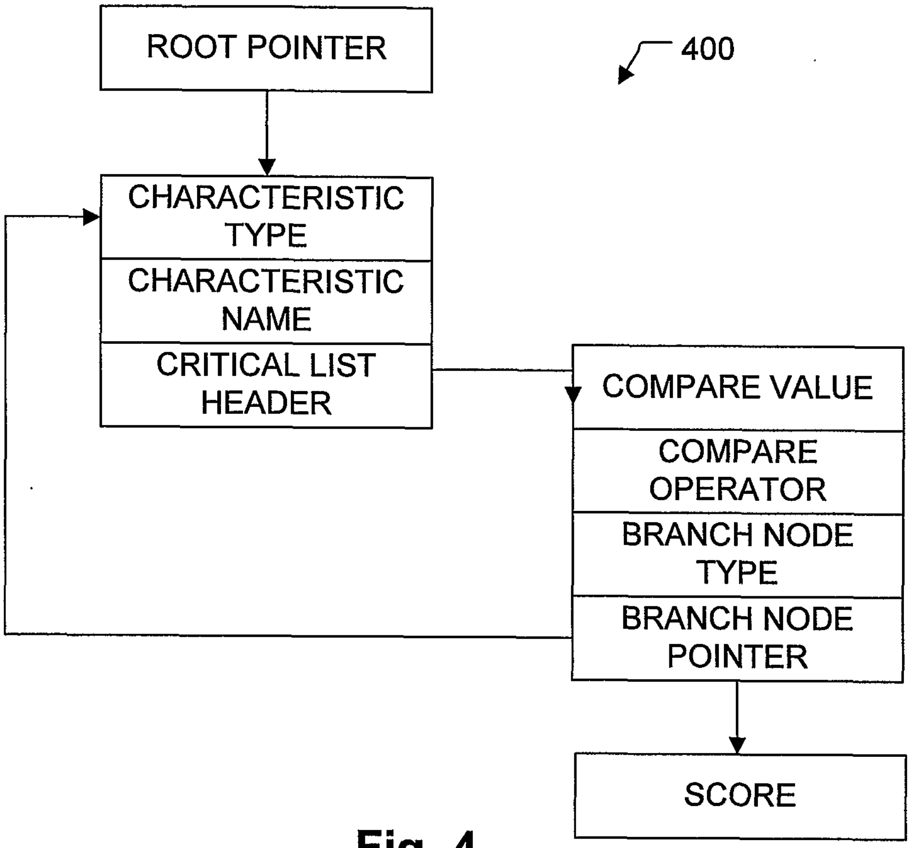

ROOT POINTER
400
CHARACTERISTIC TYPE
CHARACTERISTIC NAME
CRITICAL LIST HEADER
COMPARE VALUE
COMPARE OPERATOR
BRANCH NODE TYPE
BRANCH NODE POINTER
SCORE

Fig. 4

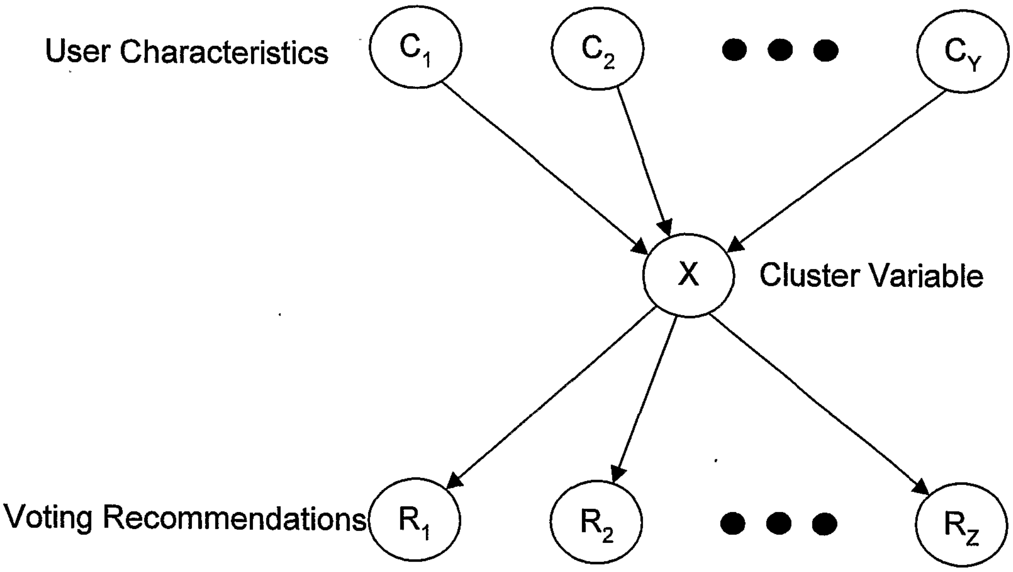

User Characteristics
C_1
C_2
C_Y
X
Cluster Variable
Voting Recommendations
R_1
R_2
R_Z

Fig. 5

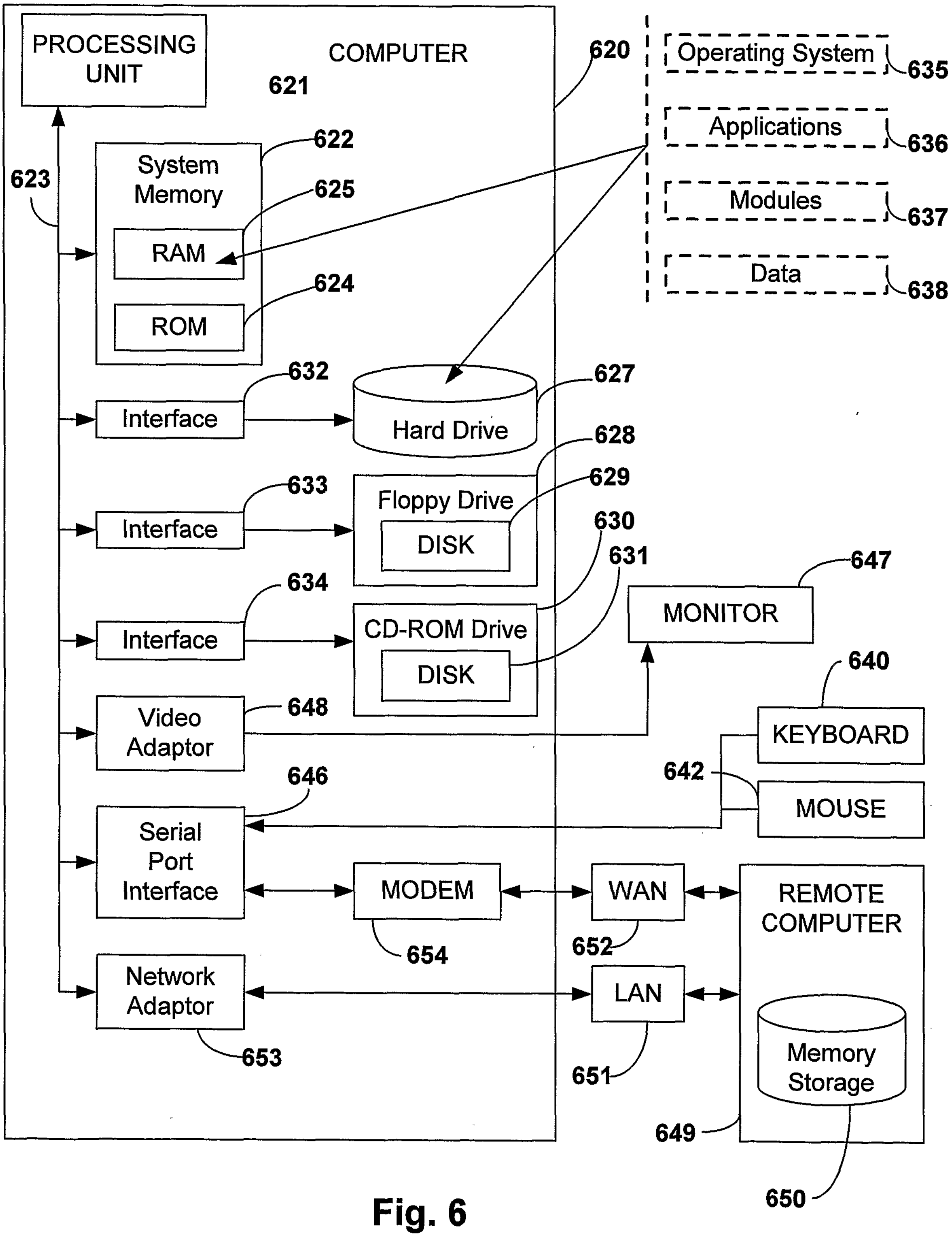

PROCESSING UNIT
COMPUTER
621
620
Operating System
635
Applications
636
Modules
637
Data
638
System Memory
622
625
RAM
624
ROM
623
632
Interface
Hard Drive
627
633
Interface
Floppy Drive
628
629
DISK
630
631
634
Interface
CD-ROM Drive
DISK
MONITOR
647
Video Adaptor
648
640
KEYBOARD
642
MOUSE
646
Serial Port Interface
MODEM
654
WAN
652
REMOTE COMPUTER
Network Adaptor
653
LAN
651
Memory Storage
649
650

Fig. 6