Chapter 9

Characterization and Prediction of Human Protein-Protein Interactions

Yi Xiong[1,2], Dan Syzmanski[3,1] and Daisuke Kihara[1,2,*]

[1]Department of Biological Sciences; [2]Department of Computer Science;
[3]Department of Agronomy, Purdue University, West Lafayette, IN 47907, USA
[*] *E-mail: dkihara@purdue.edu*

In this chapter, we introduce classification techniques that are used to predict group membership to which a data object belongs. In the first half of the chapter, we summarize several start-of-the-art classification algorithms. In the latter section, we show examples of applications of the classification techniques for prediction of protein-protein interactions in human proteome.

1. Introduction

Classification is a form of data analysis that can be used to construct a model to describe data classes of interest and predict a predefined class to which an input object belongs. For example, a classification model can be built to categorize a human tissue into a normal or cancer class based on its gene expression pattern. A wide variety of classification algorithms have been proposed by numerous researchers in machine learning, expert systems, and statistics. Although the algorithms are based on different theories, their applications follow the same framework. Generally, data classification is performed in two steps on a data set, which is split into a training set and a testing set [1]. In the first step, a model is built on the training set to describe a set of data classes. The model is represented in a variety of forms, for example, a decision tree or

[*] Corresponding author.

a mathematical formula. In the second step, the model is applied to the testing set to evaluate its performance. If the accuracy of the classification model on the testing set is acceptable, the model will be adopted to classify a new data set for which class labels are not known.

Classification techniques are widely applied for biological problems related to human health. For example, they have been applied to the cancer classification based on gene expression profiles by DNA microarrays. In a work by Golub *et al.* [2], classification models were based on a collection of tumor samples for which the cancer types or the eventual outcome were known. They classified tumor samples to known classes, which could reflect current states (such as different types of cancer) or future clinical outcomes (such as drug response or survival). Nayal and Honig [3] used crystallography data to analyze surface cavities of a nonredundant set of 99 drug-target complexes, and exploited 408 physicochemical, structural, and geometric features to construct a random forest model to predict drug binding sites. Additionally, classification methods were used to predict protein-protein interactions (PPIs) in human proteome, since understanding PPI networks can provide insights into mechanisms of human diseases such as cancers [4]. Furthermore, classification methods have been extensively applied to the identification of protein function and function sites in proteins, using a linear discriminant rule [5], Naïve Bayesian [6], support vector machine [7] and random forest [8].

In this chapter, we first introduce several frequently used state-of-the-art classification algorithms, namely, decision tree, neural network, Naïve Bayesian, support vector machine, and random forest. Then, we use the specific case of human PPI prediction as an example to show how classification methods are applied to this problem.

2. Classification methods

2.1. *Framework of classification*

There are two categories of learning in machine learning, unsupervised and supervised learning. In unsupervised learning, classes to which input objects belong to are not defined in advance and it is aimed at finding grouping or hidden structures in the input data. On the other hand, in supervised learning, input data with known class labels are used to construct a mapping from input data space to output class label space. Typically a classification task consists of four major components, namely, a dataset, a feature set and its representation, a learning algorithm, and model performance evaluation. The first part of this chapter focuses on introducing several commonly used learning algorithms in supervised learning.

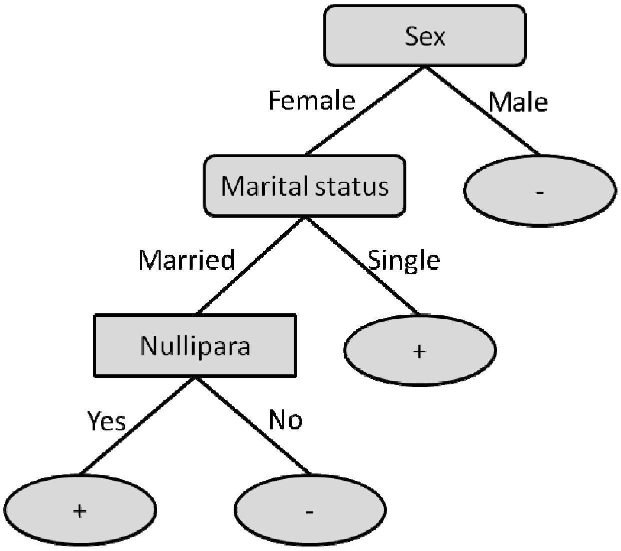

Fig. 1. A decision tree for a person that can indicate whether or not he or she has a breast cancer with high risk. The leaf node with + means a person with a high risk on breast cancer, and - is the class label for a person without high risk on breast cancer. Internal nodes are denoted by rectangles, and leaf nodes are denoted by ovals. The root as a special internal node is denoted by the rounded rectangle.

2.2. *Classification algorithms*

In this section, we introduce several state-of-art classification algorithms, which are decision tree, artificial neural network, Bayesian model, support vector machine, and random forest.

2.2.1. *Decision tree*

The advantage of decision tree-based classification is that it can be easily interpreted and intuitively understandable by human. A decision tree is a flow-chart-like tree structures (Fig. 1), where each internal node denotes a test on an attribute and leaf nodes represent classes or class distribution. The top most node in a tree is the root node, which is the starting point of the classification and where all samples belong to.

The algorithm is based on a statistical measure to select the best feature to split a node. The concept of impurity is usually used to evaluate the performance of a candidate feature in discriminating different class labels in the training samples [9]. Entropy is the most well known index to measure degree of impurity.

Let S be a set consisting of s data samples. Suppose the class label attribute has n distinct values defining n distinct classes, C_i (for i=1,2,..., n). Let s_i be the number of samples of S in class C_i. The information needed to classify a given sample (called entropy) is given by

$$I(s_1, s_2, \ldots, s_n) = -\sum_{i=1}^{n} p_i \log_2(p_i) \tag{1}$$

where p_i is the probability that an arbitrary sample belongs to class C_i, thus $p_i = s_i/s$.

The impurity reduction or information gain of an attribute A is defined as:

$$\Delta I(A) = I(s_1, s_2, \ldots, s_n) - \sum_{j=1}^{m} w_j \times I(s_{1j}, s_{2j}, \ldots, s_{nj}), \tag{2}$$

where attribute A has m distinct values, which divide S into m subsets, and s_{ij} is the number of samples of class C_i in a subset S_j. The term w_j is the weight of the jth subset. The attribute that leads to the maximal reduction of impurity is chosen to split a node among all candidate attributes.

The basic module of the decision tree construction is a greedy algorithm that builds the tree in a top-down heuristic search using the recursive manner. The complete trial and error method for all the possible partitions is intractable, therefore, the top-down heuristic search is employed on the recursive partition process. The search algorithm for the attribute is greedy and it never takes a step back to reconsider its previous choice.

The well-known algorithm ID3 [10] uses information gain (Eq. 2) to select the attribute that will best categorize the samples into individual classes. The ID3 algorithm is then executed recursively on the smaller subsets. However, attributes with continuous values are not allowed in ID3, and the information gain measure is biased on the attributes with many values. As a successor algorithm to ID3, C4.5 [11] creates a threshold to handle continuous attributes and the normalized information gain to avoid the bias.

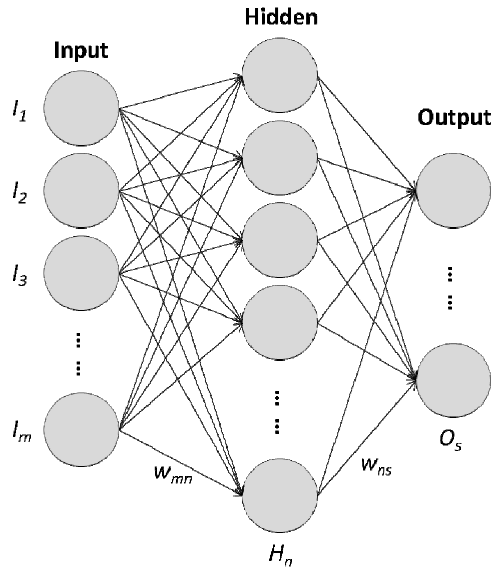

Fig. 2. The interconnected group of neurons in an artificial neural network. Weighted connections exist between consecutive layers.

2.2.2. *Artificial neural network*

Artificial neural network (ANN) is one of the most popular classification techniques, which is in the category of the artificial intelligence field that makes computers or machines simulate the thinking, psychology, evolution, and intelligent behaviors of humans. The theory of artificial neural network is inspired by the structure of human brains, which can be simplified as the network of neurons that transfer information between each other to learn data and make a decision (Fig. 2). The neurons in the input layer correspond to the attribute values of each training sample. The weighted outputs of these nodes are fed into a second layer, which is called a hidden layer. The weighted output of the hidden layer will be input to the next hidden layer. The weighted output of the last hidden layer will be input for neurons in the output layer, which makes the final prediction for given samples. One of the commonly used algorithms for training parameters for neural networks is the back-propagation method [12]. The back-propagation learning algorithm consists of two major phases: propagation and weight update. When the inputs of training samples are propagated forward, the net input and output of each neuron in the hidden and output layers are computed. Given a neuron *j* in a hidden or output layer, the net input, I_j, to neuron *j* is

$$I_j = \sum_i w_{ij} O_i + b_j \tag{3}$$

where w_{ij} is the weight of connection from neuron *i* in the previous layer to neuron *j*, and b_j is the bias (threshold value) of the neuron.

For each neuron in the hidden and output layers, the net input is then transformed by an activation function. The sigmoid function is usually used to compute output O_j of neuron *j* as:

$$O_j = \frac{1}{1+e^{-I_j}} \tag{4}$$

where I_j is input of neuron *j*.

For training the weights in the network, the algorithm compares outputs from the neural network with the actual known class labels. Observed error for each training sample is fed back to update weights so that the error of outputs will be smaller. The process of updating weights is applied to each sample in a training dataset repeatedly until the weights do not change much or until the updates is performed by a pre-determined number of times. Please see a reference [12] for more details about the back-propagation learning algorithm.

2.2.3. *Bayesian model*

Bayesian classification is based on Bayes' theorem. It can predict class membership probabilities, such as the probability that a given sample belongs to a particular class. Each training sample is represented by *n* variables (or attributes; $X_1, X_2, X_3, \ldots, X_n$). Suppose there are *m* classes, C_1, C_2, $C_3, \ldots, C_m$, for the class label variable *Y*. The possibility that a sample belongs to a class C_i is defined by the Bayes theorem as follows:

$$P(C_i \mid X_1, X_2, \ldots, X_n) = \frac{P(X_1, X_2, \ldots, X_n \mid C_i)P(C_i)}{P(X_1, X_2, \ldots, X_n)} \tag{5}$$

The Naïve Bayesian classifier assumes the independent relationship among its input attributes (Fig. 3). Therefore, the Eq. 5 can be rewritten as:

$$P(C_i \mid X_1, X_2, \ldots, X_n) = \frac{P(C_i)\prod_{i=1}^{n} p(X_i \mid C_i)}{P(X_1, X_2, \ldots, X_n)} \tag{6}$$

The class prior possibility $P(C_i)$ is estimated as the fraction of samples belonging to C_i in the whole training set.

A Bayesian model classifies a sample to class C_i if and only if

$$P(C_i \mid X_1, X_2, \ldots, X_n) > P(C_j \mid X_1, X_2, \ldots, X_n) \text{ for } 1 \le j \le m, j \ne i. \tag{7}$$

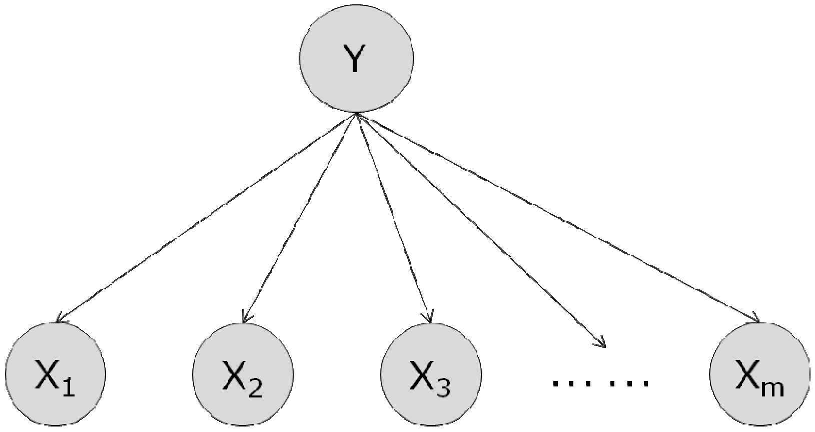

Fig. 3. A graph model of Naive Bayesian. The variable *Y* (class label) is dependent on the input variables (or observed values), which are independent between each other.

In the case that there are only two classes (C_1 and C_2, m=2) (binary classification), a likelihood ratio (*LR*) score usually is used to determine the label of the sample.

$$L(X_1, X_2, ..., X_n) = \frac{p(X_1, X_2, ..., X_n \mid C_1)}{p(X_1, X_2, ..., X_n \mid C_2)} \tag{8}$$

In the naïve Bayesian model, the *LR* can be calculated as the product of the individual likelihood ratios with respect to the features considered separately.

$$L(X_1, X_2, ..., X_n) = \prod_{i=1}^{n} \frac{p(X_i \mid C_1)}{p(X_i \mid C_2)} = \prod_{i=1}^{n} L(X_i) \tag{9}$$

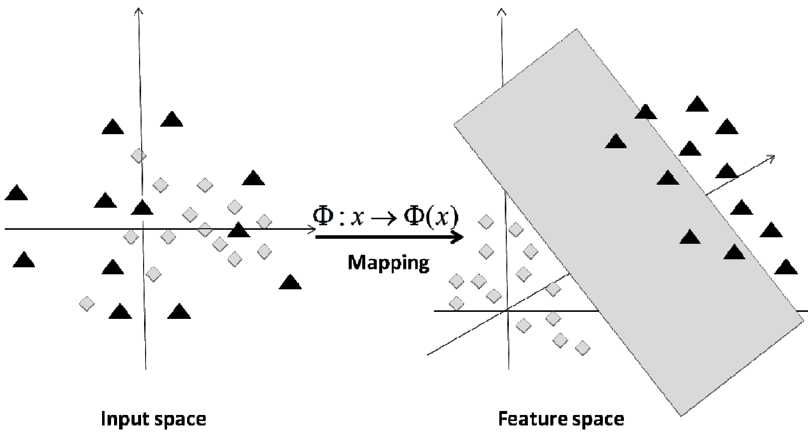

Fig. 4. Hyperplane separating two classes. In the input space, data in two classes (triangles and diamonds) cannot be linearly separated. When mapping into the feature space by a kernel function, these two classes are separated by a hyperplane in a high dimensional space.

2.2.4. *Support vector machine*

Support vector machine (SVM) classifiers map input data nonlinearly into a high dimensional feature space and separated by a hyperplane into two classes [13] (Fig. 4). Given a number of training samples belonging to two classes, a support vector machine constructs a hyperplane or set of hyperplanes in a high dimensional space. Intuitively, a good separation is achieved by selecting the hyperplane that separates the two classes but adopts the maximal distance from any one of the given samples, since the larger the margin is the lower the generalization error of the classifier for unknown samples is. The application of a kernel function allows the algorithm to fit the maximum-margin hyperplane in a transformed feature space. There are mainly three common types of kernel functions. Given the feature vectors (x_i and x_j) of two samples, kernel functions transform the distance between them.

(a) Polynomial function:

$$K\left(x_i, x_j\right) = \left[\left(x_i^T x_j\right) + 1\right]^n \tag{10}$$

(b) Gaussian radial basis function ($\gamma > 0$ is a parameter):

$$K\left(x_i, x_j\right) = e^{\left(-\gamma \left\| x_i - x_j \right\|^2\right)} \tag{11}$$

(c) Hyperbolic tangent function (v and c are parameters):

$$K\left(x_i, x_j\right) = \tanh(v(x_i^T x_j) + c) \tag{12}$$

Polynomial kernels are well suited for problems where all training data is normalized. The default recommended kernel function would be the radial basis function. The hyperbolic tangent kernel is also known as the sigmoid kernel, which comes from the neural networks field.

2.2.5. *Random forest*

Random forest is an ensemble classifier that consists of a bagging of un-pruned decision trees with a randomized selection of features at each split, and outputs the class that is the majority vote of the classes output by individual trees [14]. Generally speaking random forest can improve prediction accuracy over a single decision tree. Random forest consists of two stages of randomization. The first stage is the randomization through bagging or bootstrap aggregation, which creates new training sets by randomly sampling a given set of samples with replacement. The second randomness is to select a random subset of features for each split when building an individual tree.

To construct a random forest, the following steps are employed (Fig. 5). An individual classification tree is developed on a bootstrap sample. At each node in the tree, the split is selected on the randomly chosen subset of features. The tree is grown to full size without pruning. These two step are repeated for *n* times for *n* trees to construct a forest. The ensemble classification label is a majority vote of the prediction from all the trees.

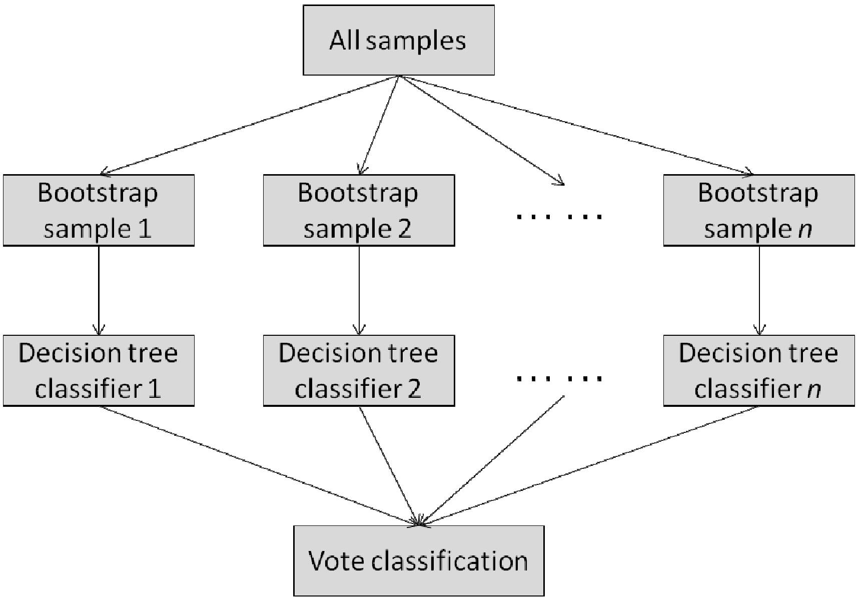

Fig. 5. A flowchart of a random forest model. Each decision tree is trained on a subset of the whole data set, which is chosen by random sampling with replacement. In each tree classifier, a random subset of the features is selected at each split. The final output class of the forest is selected by the majority vote of the classes output by all *n* trees.

2.3. *Model validation and evaluation*

When a classification model is built using one of the algorithms introduced above, it requires a golden standard dataset to validate and evaluate how well the constructed model works. The golden standard dataset consists of well-labeled samples, which are divided into a training set and testing set. For model validation, there are three methods that are commonly employed to judge the performance of classification models. They are *k*-fold cross-validation, leave-one-out cross-validation, and independent tests. In the *k*-fold cross-validation, a sample set is randomly partitioned into *k* subsets of equal size. Of the *k* subsets, one subset is retained as the validation dataset for testing the model, and the remaining *k*-1 subsets are used for training. The cross-validation process is then repeated *k* times with each of the *k* subsets used exactly once as the validation data. The *k* results from all rounds are finally averaged to generate a single estimation metric. Leave-one-out cross-validation (LOOCV) uses a single instance in the sample set for testing while the remaining instances are used as the training data. This is repeated so that each instance in the sample set is used once as the validation data. This is the same as a *k*-fold cross-validation with *k* being equal to the number of instances in the original sample set. Leave-one-out cross-validation is computationally expensive when the number of samples in the training set is too large. In order to test the model in an unseen sample set, an independent test will be adopted. Independent test is conducted on a dataset which is independent from the training set. Thus, it is mimicking the actual scenario of prediction.

In order to assess the classification performance, various performance criteria are defined:

$$\mathrm{Re}\,call=\frac{TP}{TP+FN} \tag{13}$$

$$Specificity=\frac{TN}{TN+FP} \tag{14}$$

$$False\ positive\ rate=\frac{FP}{TN+FP} \tag{15}$$

$$\mathrm{Precision}=\frac{TP}{TP+FP} \tag{16}$$

$$F_1=\frac{2\times\mathrm{Re}\,call\times\mathrm{Pr}\,ecision}{\mathrm{Re}\,call+\mathrm{Pr}\,ecision} \tag{17}$$

where *TP* is the number of correctly predicted positive samples, *TN* is the number of correctly predicted negative samples, *FP* is the number of negative samples wrongly predicted as positives and *FN* is the number of positive samples wrongly predicted as negatives.

The receiver operating characteristic (ROC) curve is a plot of the sensitivity (also called as recall) versus (1-specificity) for a binary classifier at varying thresholds. The area under the curve (*AUC*) can be used as a threshold-independent measure of classification performance. Alternatively, the precision-recall (PR) curve can be used, which plots recall relative to precision for a binary classifier at varying thresholds.

3. Application to human PPI data

3.1. *Background of human PPI and biological problem statement*

Most of cellular functions are carried out through protein interactions. PPI data identifies interactions of proteins in a cell, which can provide insights into mechanisms that underlie human diseases. PPI data may also lead to new drug development to prevent the diseases. Since the current human PPI map is estimated to be far from complete [4], there is a strong need to increase the coverage of the human interactome by classifications (also called predictions here). In the last decade, the high throughput experimental technologies such as the yeast two hybrid (Y2H) assay and targeted mass spectrometry are employed to investigate protein-protein interaction networks in a whole organism scale [15]. However, it is pointed out by some researchers that these high-throughput experimental methods have high false positive rates, and analysis of the high-throughput datasets has shown that they do not overlap much with each other [16, 17]. Accurate computational methods are therefore necessary to complete the interactome, which will compensate time-consuming and expensive experimental methods for identifying PPIs. Here, using recent works on PPI data as examples, we will see how classification techniques are used in practice. The methods capture observed features of interacting proteins and are applied to predict novel interactions between protein pairs. Note that although PPIs are dynamic and are often condition-specific, these methods classify PPIs as interact or non-interact pairs in a static manner.

Table 1. The list of public databases of protein-protein interactions (the numbers were obtained on Sep, 2012).

Database name	Number of organisms	Number of human PPI	URL
BioGRID	39	75096 (physical interaction)	http://thebiogrid.org/
DIP	504	4794	http://dip.doe-mbi.ucla.edu/dip/
HPRD	1	39194	http://www.hprd.org/
IntAct	186	4578 (direct interaction)	http://www.ebi.ac.uk/intact/
MINT	30 (main)	26700	http://mint.bio.uniroma2.it/

3.2. *Human PPIs databases*

Currently known human PPIs are collected in several databases (Table 1), which are curated from the experimental data and primary biomedical literature. The Biological General Repository for Interaction Datasets (BioGRID) [18] is a public database that collects genetic and protein interaction data from model organisms and humans. The Database of Interacting Proteins (DIP) [19] contains experimentally determined protein-protein interactions of a large number of organisms. The Human Protein Reference Database (HPRD) is a database which integrates the information of protein functions and interaction of human proteins [20]. IntAct [21] is an open-source protein interaction database, where the source code and data are freely accessible. The Molecular INTeraction database (MINT) [22] is a public repository focusing on PPI data reported in peer-reviewed literature. The interaction data contained in these databases are mainly between soluble proteins. Thus, they should be used with caution if interactions between membrane proteins are to be predicted.

3.3. *Datasets for computational study of PPIs*

Known PPIs in these databases are used for training classification methods. To build a method that can reliably classify protein pairs into interacting or non-interacting, we need two datasets, a positive dataset that contains known interacting protein pairs and a negative dataset that consists of non-interacting protein pairs.

To construct a positive dataset, known interacting protein pairs are extracted from some or all of the aforementioned databases. Duplicate interactions from different databases will be deleted. Constructing a negative dataset is not as trivial as a positive dataset, because it is often difficult to distinguish protein pairs that are actually interacting with each other but not yet detected by current experiments from pairs that are truly not interacting. The first strategy to construct a negative set is to randomly select protein pairs that are from different sub-cellular locations so that they are unlikely to physically encounter in a cell [23]. Because the probability that two randomly selected proteins physically interact is low, another approach taken is to randomly pair any two proteins from the positive data set excluding pairs that are actually known to interact [24].

There is a database named Negatome [25], which contains lists of experimentally supported non-interacting protein pairs by manual curation of literature and from analysis of protein complexes with known 3D structure. The database contains 1291 and 809 non-interacting pairs, respectively.

3.4. *Protein features used for predicting PPIs*

A prediction method for PPI considers features of known interacting proteins (from a positive dataset) and known non-interacting protein pairs (from a negative dataset) to build (train) a model. It is a binary classification problem of protein pairs, either to interacting or non-interacting. Features used in existing studies include orthologous relationship to known interacting proteins in another organism (called interolog [26]). This is an effective strategy because protein interactions are often conserved among highly diverged organisms ranging from the model plant *Arabidopsis thaliana* to humans. Gene co-expression data can be also used for predicting interacting proteins because expression profiles of interacting proteins simultaneously rise and fall in different conditions and cell types. Detecting known interacting domains in protein pairs is another strategy to predict their interaction. Functional similarity measured by semantic similarity (which indicate how similar two terms are based on their semantic properties) of Gene Ontology (GO) terms is based on the fact that interacting proteins may function in the same biological process or at the same subcellular location. Some PPI prediction methods exploit comparative genomics features, such as conserved gene orders, gene fusion and phylogenetic profile similarity (i.e. co-occurrence or co-absence of genes in multiple genomes) [27]. Below we discuss three methods using different combination of features and classifiers for integrative analysis and prediction of human PPIs.

3.5. *Case studies on human PPI prediction*

We review three examples of application of classification methods for predicting PPIs in human. The first work [28] used a naïve Bayesian model to integrate four types of features to predict PPIs in human. The second report [17] improved over the first one by using more relevant features and a semi-naïve Bayesian model, which combined dependent features together to construct a naïve Bayesian model. The third paper [16] employed an active learning strategy to guide the selection of training data, which was shown to improve prediction accuracy. The detail comparison of these three case studies is summarized in Table 2.

Table 2. Methodological differences among three case studies.

Difference	**Rhodes *et al.* [28]**	**Scott *et al.* [17]**	**Mohamed *et al.* [16]**
Positive Dataset	11,678 PPIs in HPRD	26, 896 PPIs in HPRD	14, 600 PPIs in HPRD
Negative Dataset	A localization-derived negative dataset	A randomly-generated negative dataset, and a localization-derived negative dataset	A randomly-generated negative dataset
Model	Naïve Bayesian	Semi-naïve Bayesian	Active Learning and Random Forest
Features	interolog, correlation of gene expression, the number of shared biological process GO terms, and the domain enrichment ratio	correlation of gene expression, interolog, sub-cellular localization, co-occurrence of specific InterPro and Pfam domains, co-occurrence of post-translational modifications, presence of disorder regions, and local network topology of PPI network	GO terms in cellular component, molecular function, and biological process, co-occurrence in tissue, gene expression, sequence similarity, interolog, and domain interaction
Evaluation metrics	False positive rate	AUC of ROC	Recall, precision, and F-score

3.5.1. *Application of naïve Bayesian model*

In a study by Chinnaiyan and his colleagues [28], four features were considered to predict PPI using a naïve Bayesian model. A naïve Bayesian model computes the probability that a pair of two proteins are interacting using each feature and multiplicatively combines the probabilities computed for different features. This multiplicative nature requires that the predictive data sets are conditionally independent or nonredundant. The four features they used were interolog (existence of homologous proteins that are interacting), correlation of gene expression, the number of shared biological process GO terms, and the domain enrichment ratio. The domain enrichment ratio is calculated as the probability of observing a pair of domains in a set of known interacting proteins divided by the product of the probabilities of observing each domain of the pair independently. The protein domains were taken from the InterPro database [29]. Biologically, these features have independent information, except for the last two features that are related to protein functions. To avoid bias from the two dependent features, the last two features were analyzed together.

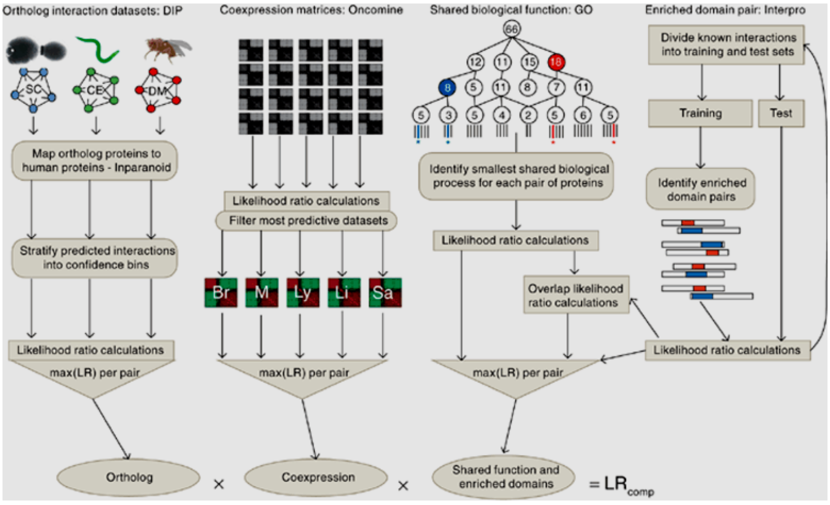
Ortholog interaction datasets: DIP
Coexpression matrices: Oncomine
Shared biological function: GO
Enriched domain pair: Interpro
SC
CE
DM
Map ortholog proteins to human proteins - Inparanoid
Stratify predicted interactions into confidence bins
Likelihood ratio calculations
max(LR) per pair
Ortholog
Likelihood ratio calculations
Filter most predictive datasets
Br
M
Ly
Li
Sa
max(LR) per pair
Coexpression
Identify smallest shared biological process for each pair of proteins
Likelihood ratio calculations
Overlap likelihood ratio calculations
max(LR) per pair
Shared function and enriched domains
Divide known interactions into training and test sets
Training
Test
Identify enriched domain pairs
Likelihood ratio calculations
×
×
= LRcomp

Fig. 6. The flowchart of data integration in a Naïve Bayes model to predict human protein-protein interactions (The figure is taken from [25] with permission from the journal).

Figure 6 shows the naïve Bayes model which combines the features mentioned above. First the authors calculated likelihood ratios (i.e. ratio of the probability that a protein pair is interacting over the probability that they are not) (Eq. 9) using each feature. For the interolog feature, human PPIs were predicted from the interaction data sets of three model organisms, *Sacchromyces cerevisiae, Caenorhabditis elegans,* and *Drosophila melanogaster* (Fig. 6, left branch). A confidence level of interolog assignments were classified by considering several parameters associated with predicted interactions, including the number of independent lines of evidence for a yeast PPI, confidence value of ortholog assignment, etc. Using these parameters, human protein pairs were classified into several confidence level bins using a decision tree (the step of stratifying predicted human PPIs in the figure). Then, Naïve Bayes model was constructed to predict whether a human protein pair interacts or not given the confidence level of the interolog assignment. If a human protein pair has interologs in two or more organisms, it will have multiple likelihood ratio from each of the PPIs. In such case, the maximal ratio was chosen from them (the last step of the PPI branch).

For the gene expression features, human protein pairs are classified into bins by their correlation value of their expression level observed in each of the five expression data sets from different tissues (Fig. 6, second branch from the left). Then, similar to the interolog-based feature, the likelihood ratio of the gene expression-based feature was computed from each of the five expression data sets that contain the given protein pairs. If a protein pair appeared in multiple expression data, the maximum ratio was chosen.

Since they found that the number of shared biological process GO terms and the domain enrichment ratio (two right branches in Fig. 6) were redundant, two features were binned together to compute the likelihood ratio. At last, the likelihood ratios were combined in a Naïve Bayesian model to generate composite likelihood ratios (LR_{comp}).

They used a training dataset of 11,678 known human PPIs from HPRD and 3,106,928 non-interacting protein pairs in human. The trained model predicted 38,986 new PPIs that were not reported in HPRD at a false positive rate of 50% and 9,651 new PPIs at a false positive rate of 20%. The authors claimed that the false positive rate of this classification method is comparable to the results of high throughput experimental approaches in model organisms.

3.5.2. *Application of semi-naïve Bayesian model*

A drawback of applying naïve Bayes for PPI prediction is that it assumes independence of each feature; however, often some features are closely related. If features considered in a model are independent, the likelihood ratio can be calculated as the product of the individual likelihood ratios (naïve Bayesian model, Eq. 9). On the other hand, if features are not independent, all possible combinations of all states of these features must be considered, which can be very computationally intensive (Eq. 8).

The semi-naïve Bayesian model is addressing the drawback of naïve Bayesian model by explicit combination of related features (Eq. 8) while handling independent features (Eq. 9) in the same way as the naïve Bayesian model. In a study by Barton and his colleagues [17], the semi-naïve Bayesian model is used for human PPI prediction using seven features: correlation of gene expression, interolog, sub-cellular localization, co-occurrence of specific InterPro [29] and Pfam [30] domains, co-occurrence of post-translational modifications, presence of disorder regions, and local network topology of PPI network. The local PPI network topology measure reflects the fraction of commonly interacting proteins for a pair of proteins. The sub-cellular localization, protein domain co-occurrence, and post-translational modification co-occurrence are integrated into one combined module because considering all their combinations (dependencies) between them achieved a higher accuracy. That is, the joint probability of all possible combinations of the four localization bins, five chi-square domain-co-occurrence bins, and four post-translational modification co-occurrence score bins, was computed. The rest of four features and the combined module were considered to be independent and integrated in the naïve Bayesian classifier (Fig. 7).

For training and testing, 62,322 human protein sequences were downloaded from the International Protein Index (IPI) database [31]. Among these proteins, 26,896 distinct human protein interactions were identified as the positive dataset from HPRD. Of the 62,322 human proteins, 22,889 human proteins were referred to as the Informative Protein Set (IPS) since they were characterized by at least one of the features mentioned above. Two types of negative datasets were constructed. The first type of negative dataset was generated by selecting protein pairs at random from IPS. The second type of negative dataset was created by selecting protein pairs from IPS for which the HPRD annotates them in two different subcellular locations. The localization-derived negative trained classifier tested on sets containing localization-derived negatives achieves a lower accuracy than that of the random negative trained classifier tested on a test set containing randomly-generated negatives. This is most likely due to the fact that the localization-derived negative trained predictor cannot sample the whole protein pair space well. Their model predicted 37,606 human PPIs, of which 32,892 were not reported in other publicly available large human interaction datasets. The newly discovered interactions thus considerably increased the coverage of the human interaction map.

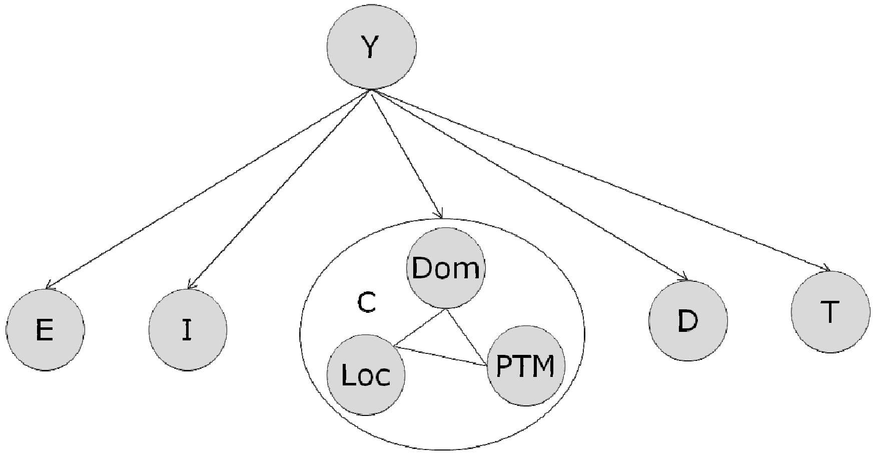

Fig. 7. Overview of semi-naive Bayesian. The variable *Y* (class label) is dependent on the observed variables, i.e. E(Expression), I(Interolog), C(Combined), D(Disorder) and T(Topology), which are treated as independent variables. In the combined module C, localization, domain co-occurrence, and post-translational modification co-occurrence are considered to be dependent on each other.

3.5.3. *Application of Active Learning and Random Forest*

Experimentally verified protein-protein interactions are expensive to obtain; therefore, it would be useful to develop strategies to minimize the number of labeled samples required in the supervised learning task. In comparison to passive learning models (labels of training sample are known prior to training) introduced in the sections above, active learning is allowed to request the label of any particular input sample in the training data. Active learning is a type of iterative supervised learning in which the system selects most informative samples each time to obtain their labels from a large pool of samples. Sampling will be repeated until the obtained samples in the training set are both plentiful and representative to construct a classification model with satisfied performance. The benefit of active learning is to substantially reduce the number of labeled samples required, making the training of a classification model more practical.

In active learning, ideas of selecting informative training data include density based sampling, where samples to be selected are distributed on clusters in proportion to the cluster size; uncertainty sampling, where samples to be selected are those which are mispredicted using current classifier; estimated-error reduction, where samples that would generate maximal error reduction to the classifier are selected. Here, we provide details of density based and uncertainty based (random seed) sampling techniques.

(1) Density based sampling

The samples are clustered by a K-means clustering algorithm. Labels are requested for a fixed number (*M*) of samples in each iteration. The selected samples are distributed across the clusters in proportion to the size of the cluster. Let n_i be the number of samples in cluster C_i, and N be the total number of all samples. Then, m_i , the number of samples to be selected from cluster C_i is given by

$$m_i = M * n_i / N \tag{18}$$

and,

$$M = \sum_{i=1}^{K} m_i \tag{19}$$

In each cluster C_i, the algorithm selects m_i unlabelled samples closest to the centroid, and assigns labels on them [16].

(2) Uncertainty based sampling

In this strategy, the samples, whose labels are asked, are randomly selected in the first iteration. For example, a random forest is employed to construct the model with the selected samples. In the following iterations, the samples selected for labeling are those which have maximum disagreement among the decision trees in the random forest. The entropy (confusion) in labeling the sample is measured as

$$e_i = -\sum_{i \in (0,1)} p_i \log(p_i) \tag{20}$$

where, p_0 is the fraction of decision trees in the random forest that label the samples as negative ones, and p_1 is the fraction of decision trees that label the samples as positive.

In each iteration, a fixed number of samples with the maximum confusion are selected and their labels are obtained. A new random forest is trained from the expanding number of samples for selecting the samples with maximal confusion in the next iteration.

In a work by Ganapathiraju and colleagues [16], 14, 600 interacting protein pairs were downloaded from HPRD and a set of 400, 000 non-interacting protein pairs were randomly generated. Features used to characterize protein pairs were GO terms in cellular component category (1), GO terms in molecular function category (1), GO terms in biological process (1), co-occurrence in tissue (1), gene expression (16), sequence similarity (1), interolog (5), and domain interaction (1). The numbers in parentheses show the number of different features in the same category. Not all types of features were available for each protein pair. A homogenous subset of data was built such that every pair had more than 80% feature coverage, which resulted in a total of 55,950 protein pairs for use in the training and testing.

In order to test the active learning model, training samples were selected using different active learning sample selection strategies as described above. Random forest was used for classification. Since some of the 27 features are obviously redundant with each other, a randomly reduced set of features were used to build each decision tree in the random forest. In their work, 20 decision trees were constructed. To split the nodes of the decision trees, a subset of seven feature elements were randomly selected from the total of 27 elements. Of the seven selected features, the feature that offered the maximal information gain (Eq. 2) was used to split each node.

It was shown that using active learning to select training data achieved higher accuracy than the model trained on randomly selected training samples without active learning. The best model achieves an F-score (harmonic mean of recall and precision) of 60% at 3000 labeled samples, with a recall of 51% and precision of 73%. The F-score dropped to 50% when active learning was not used (instead, training data were selected randomly). It was demonstrated that active learning enables better learning with less labeled training data.

4. Conclusions

Data classification is the form of data analysis for extracting models describing important data classes, and predicting a predefined class to which a given sample belongs. Five well known algorithms were introduced in this chapter. There are also a number of other methods, which are gaining increasing popularity in the data mining and bioinformatics fields. These methods include genetic programming-based algorithms [32] and fuzzy set algorithms [33]. Classification and prediction based on classification of known data is an indispensable step for understanding and using a large scale data. With applications to human PPIs discussed here, classification models have increased the coverage of the human interactome. Moreover, classification techniques have been applied for other various types of biological and medical data to discover novel knowledge from them.

Acknowledgments

This work has been supported by grants from the National Institutes of Health (R01GM075004 and R01GM097528 to D.K.), National Science Foundation (EF0850009, IIS0915801, DMS0800568 to D.K., IOS1127027 to D.S. and D.K.), and National Research Foundation of

Korea Grant funded by the Korean Government (NRF-2011-220-C00004) to D.K.

References

1. Han, J. and M. KAMBER, *Data Mining: concepts and techniques*, 2001, Morgan Kaufmann.
2. Golub, T.R., D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, *et al.*, *Molecular classification of cancer: class discovery and class prediction by gene expression monitoring.* Science, 1999. **286**(5439): p. 531-7.
3. Nayal, M. and B. Honig, On the nature of cavities on protein surfaces: application to the identification of drug-binding sites. Proteins-Structure Function and Bioinformatics, 2006. **63**(4): p. 892-906.
4. Venkatesan, K., J.F. Rual, A. Vazquez, U. Stelzl, I. Lemmens, T. Hirozane-Kishikawa, *et al.*, *An empirical framework for binary interactome mapping.* Nat Methods, 2009. **6**(1): p. 83-90.
5. Kihara, D., T. Shimizu and M. Kanehisa, Prediction of membrane proteins based on classification of transmembrane segments. Protein Eng, 1998. **11**(11): p. 961-70.
6. Messih, M.A., M. Chitale, V.B. Bajic, D. Kihara and X. Gao, *Protein domain recurrence and order can enhance prediction of protein functions.* Bioinformatics, 2012. **28**(18): p. i444-i450.
7. Xiong, Y., J. Liu and D.Q. Wei, *An accurate feature-based method for identifying DNA-binding residues on protein surfaces.* Proteins-Structure Function and Bioinformatics, 2011. **79**(2): p. 509-17.
8. Zhang, W., Y. Xiong, M. Zhao, H. Zou, X. Ye and J. Liu, Prediction of conformational B-cell epitopes from 3D structures by random forests with a distance-based feature. BMC Bioinformatics, 2011. **12**: p. 341.
9. Chen, X., M.H. Wang and H.P. Zhang, *The use of classification trees for bioinformatics.* Wiley Interdisciplinary Reviews-Data Mining and Knowledge Discovery, 2011. **1**(1): p. 55-63.
10. Quinlan, J.R., *Induction of decision trees.* Machine learning, 1986. **1**(1): p. 81-106.
11. Quinlan, J.R., *C4. 5: programs for machine learning*1993: Morgan Kaufmann.
12. Hecht-Nielsen, R. Theory of the backpropagation neural network. 1988. IEEE.
13. Cortes, C. and V. Vapnik, *Support-vector networks.* Machine learning, 1995. **20**(3): p. 273-297.
14. Breiman, L., *Random forests.* Machine learning, 2001. **45**(1): p. 5-32.
15. Rual, J.F., K. Venkatesan, T. Hao, T. Hirozane-Kishikawa, A. Dricot, N. Li, *et al.*, *Towards a proteome-scale map of the human protein-protein interaction network.* Nature, 2005. **437**(7062): p. 1173-8.

16. Mohamed, T.P., J.G. Carbonell and M.K. Ganapathiraju, *Active learning for human protein-protein interaction prediction.* BMC Bioinformatics, 2010. **11 Suppl 1**: p. S57.
17. Scott, M.S. and G.J. Barton, Probabilistic prediction and ranking of human protein-protein interactions. BMC Bioinformatics, 2007. **8**: p. 239.
18. Stark, C., B.J. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz and M. Tyers, *BioGRID: a general repository for interaction datasets.* Nucleic Acids Res, 2006. **34**(Database issue): p. D535-9.
19. Salwinski, L., C.S. Miller, A.J. Smith, F.K. Pettit, J.U. Bowie and D. Eisenberg, *The Database of Interacting Proteins: 2004 update.* Nucleic Acids Res, 2004. **32**(Database issue): p. D449-51.
20. Peri, S., J.D. Navarro, R. Amanchy, T.Z. Kristiansen, C.K. Jonnalagadda, V. Surendranath, *et al.*, *Development of human protein reference database as an initial platform for approaching systems biology in humans.* Genome Res, 2003. **13**(10): p. 2363-71.
21. Kerrien, S., B. Aranda, L. Breuza, A. Bridge, F. Broackes-Carter, C. Chen, *et al.*, *The IntAct molecular interaction database in 2012.* Nucleic Acids Res, 2012. **40**(Database issue): p. D841-6.
22. Licata, L., L. Briganti, D. Peluso, L. Perfetto, M. Iannuccelli, E. Galeota, *et al.*, *MINT, the molecular interaction database: 2012 update.* Nucleic Acids Res, 2012. **40**(Database issue): p. D857-61.
23. Guo, Y., L. Yu, Z. Wen and M. Li, Using support vector machine combined with auto covariance to predict protein-protein interactions from protein sequences. Nucleic Acids Res, 2008. **36**(9): p. 3025-30.
24. Shen, J., J. Zhang, X. Luo, W. Zhu, K. Yu, K. Chen, *et al.*, *Predicting protein-protein interactions based only on sequences information.* Proc Natl Acad Sci U S A, 2007. **104**(11): p. 4337-41.
25. Smialowski, P., P. Pagel, P. Wong, B. Brauner, I. Dunger, G. Fobo, *et al.*, *The Negatome database: a reference set of non-interacting protein pairs.* Nucleic Acids Res, 2010. **38**(Database issue): p. D540-4.
26. Walhout, A.J., R. Sordella, X. Lu, J.L. Hartley, G.F. Temple, M.A. Brasch, *et al.*, *Protein interaction mapping in C. elegans using proteins involved in vulval development.* Science, 2000. **287**(5450): p. 116-22.
27. Hawkins, T. and D. Kihara, *Function prediction of uncharacterized proteins.* J Bioinform Comput Biol, 2007. **5**(1): p. 1-30.
28. Rhodes, D.R., S.A. Tomlins, S. Varambally, V. Mahavisno, T. Barrette, S. Kalyana-Sundaram, *et al.*, *Probabilistic model of the human protein-protein interaction network.* Nat Biotechnol, 2005. **23**(8): p. 951-9.
29. Hunter, S., R. Apweiler, T.K. Attwood, A. Bairoch, A. Bateman, D. Binns, *et al.*, *InterPro: the integrative protein signature database.* Nucleic Acids Res, 2009. **37**(Database issue): p. D211-5.

30. Punta, M., P.C. Coggill, R.Y. Eberhardt, J. Mistry, J. Tate, C. Boursnell, *et al.*, *The Pfam protein families database.* Nucleic Acids Res, 2012. **40**(Database issue): p. D290-301.
31. Kersey, P.J., J. Duarte, A. Williams, Y. Karavidopoulou, E. Birney and R. Apweiler, *The International Protein Index: an integrated database for proteomics experiments.* Proteomics, 2004. **4**(7): p. 1985-8.
32. Liu, K.H. and C.G. Xu, A genetic programming-based approach to the classification of multiclass microarray datasets. Bioinformatics, 2009. **25**(3): p. 331-7.
33. Jin, Y. and L. Wang, *Fuzzy systems in bioinformatics and computational biology*. Vol. 242. 2009: Springer Verlag.