Breast Cancer Recurrence Prediction

Chang Liu Kangning Chen

College of Engineering & Tepper Business School， Carnegie Mellon University

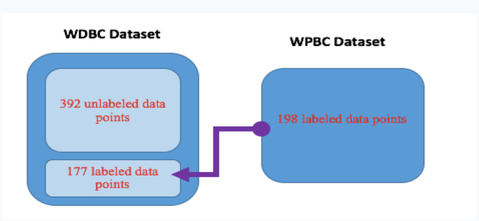
INTRODUCTION

Breast cancer is considered to be the second leading cause of cancer deaths in women today. Nowadays, as the development on machine technique, the data mining skills are widely used in medical industry to effectively predict and diagnosis diseases, including cancer-curing. In this essay, we will use supervised learning and semi-supervised machine learning method to study the probability and time for the potential recurrence.

In real world applications, labeled data are relatively hard to get while unlabeled data are cheap. The target label y requires human annotation which takes a long time, and these experiments require a lot of resources including experienced experts and special devices. Therefore, a trend to utilize the surplus unlabeled data together with scarcely labeled data is desirable. In this project, we will incorporate unlabeled data set (Diagnostic Dataset) to improve the classifier.

DATASET

We use two datasets for this project. The Wisconsin Prognostic Breast Cancer data set contains 198 instances labeled first for recurrence of breast cancer. Along with the label for recurrence, this data set records the time before recurrence or the time when a patient remains disease free. Along with the recurrence and time, this data set contains 10 relevant attributes describing the characteristics of the cell nuclei present in the digitized image of a fine needle aspirate (FNA) of the breast mass.



The second Wisconsin Prognostic Breast Cancer dataset contains 569 instances with the ID number and the diagnosis outcome (malign or benign) and the same 10 relevant attributes as the prognostic dataset. Each attribute has three values: The mean, standard error, and ”worst” or largest. In addition, the prognostic dataset has more attributes: Time, Tumor size and Lymph node status. The detailed information is displayed in Table 1.

DATA PREPROCESSING

The data set relatively clean and complete. However, we need to do two steps before implement our data. The first step is to fill the missing data. There are a few unobservable values in the attribute Lymph node status of Prognostic Dataset. We simply use the mean to generate these field to avoid noises.

Table 1: The main characteristics of the Wisconsin breast cancer dataset

Attribute	Range
*Time (recurrence time if field 2 = R, disease-free time if field 2 = N) 4-33)	(1,125)
Radius (mean of distances from the centre to all points on the perimeter)	(10.95, 27.22)
Texture (standard deviation of gray-scale values)	(10.38, 39.28)
Perimeter	(71.90,182.10)
Area	(361.60, 2250)
Smoothness (local variation in radius lengths)	(0.075, 0.145)
Compactness (perimeter^2/area − 1.0)	(0.046, 0.311)
Concavity (severity of concave portions of the con- tour)	(0.024, 0.427)
Concave points (number of concave portions of the contour)	(0.020, 0.201)
Symmetry	(0.131, 0.304)
Fractal dimension (”coastline approximation” - 1)	(0.050, 0.097)
*Tumour size - diameter of the excised tumour in centimetres	(0.400, 10.00)
*Lymph node status - number of positive auxiliary lymph nodes	(0, 27)

The attributes with * are the attributes that only Prognostic Dataset has.

METHOD 1 GUASSIAN NAÏVE BAYES

Gaussian Naive Bayes is the methodology to model the generative distribution of (X, Y) based on the training set, and predict the outcome by choosing the highest possible label. Despite its name and its naive intuition, it is still one of the most effective and efficient algorithms for data mining.

Gaussian Naive Bayes algorithm

1. Estimate μ_{ik} = mean of X_i given $Y = Y_k$, and σ_{ik}^2 = variance of X_i given $Y = Y_k$. Also, from training set, we can estimate $\mathbb{P}(Y = y_k) = \frac{\text{number of records with } Y=y_k}{\text{sample size}}$
2. By applying bayesian rule, we want to compute:

$$\mathbb{P}(Y = y_k|X = x) = \frac{\mathbb{P}(x|y_k)\mathbb{P}(y_k)}{\mathbb{P}(x)}$$

Since $\mathbb{P}(x)$ is the same for all y_k, we have:

$$\hat{Y} = argmax_y \frac{\mathbb{P}(x|y)\mathbb{P}(y)}{\mathbb{P}(x)}$$
$$= argmax_y \mathbb{P}(x|y)\mathbb{P}(y)$$

3. Assume that the attributes are independent Gaussian distributed. Then:

$$\mathbb{P}(x|y) = \prod_i \mathbb{P}(X_i = x_i|y) = \prod_i \frac{1}{\sqrt{2\pi}\sigma_{ik}} e^{-\frac{(X_i-\mu_{ik})^2}{2\sigma_{ik}^2}}$$

4. Plug the results in step 3 back to the formula in step 2. We can compute our prediction $\hat{Y}$

Improvements:

Naïve Bayes assumed that the attributes in X are pairwise independent (uncorrelated). In this problem, the assumption is clearly incorrect. For example, the correlation between Radius and Area is greater than 0.9. We have following improvements:

Multivariate Gaussian Distribution

We assume the distribution be multi-normal. Specifically, in step 3 of Naive Bayes algorithm, the formula becomes:

$$P(x|y) = \frac{1}{(2\pi)^{n/2}|\sum|^{1/2}} e^{\frac{-(X-\mu)^T\sum^{-1}(X-\mu)}{2}}$$

where $\sum$ is the covariance matrix trained by training set.

Further improvement for Naive Bayes is choosing proper feature rather than simply include all features in the dataset. In the Diana’s paper, all features related to the average are brutally included and all other features are brutally expelled. Greedy backward elimination can be applied here to eliminate redundant features.

Greedy backward elimination

1. Add all features to train the classifier.
2. Iterate each feature and train the classifier without this feature. Remove the feature with the highest test error.
3. Repeat step 2 until the optimal feature subset is founded.

Principle Component Analysis

First, we need to normalize the data, because the range in each attribute is highly diversified. Then we apply PCA transformation on the original data and choose top 5 important features before we run Naïve Bayes Algorithm. Also, we can apply PCA on both labeled and and unlabeled data to help training transformation weights.

METHOD 2 SEMI-SUPERVISED LEARNING

Semi-supervised learning a class of supervised learning tasks and techniques that also make use of unlabeled data for training - typically a small amount of labeled data with a large amount of unlabeled data. The learner has both labeled training data $\{(x_i, y_i)\}_{i=1}^{l}$and unlabeled training data $\{x_i\}_{i=l+1}^{l+u}$, and learns a prediction $f: X \to Y, f \in F$. Where F is the hypothesis space. The predictor learned by semi-supervised methods usually predicts future test data better than that learned by supervised learning which only considers the labeled training data.

1. Start from MLE $\theta = \{w, \mu, \Sigma\}_{1:2}$ on (X_l, Y_l), repeat:
2. The E-step: compute the expected label $p(y|x, \theta) = \frac{p(x,y|\theta)}{\sum_{y'} p(x,y'|\theta)}$ for all $x \in X_u$
   - label $p(y = 1|x, \theta)$-fraction of x with class 1
   - label $p(y = 2|x, \theta)$-fraction of x with class 2
3. The M-step: update MLE θ with (now labeled) X_u
   - w_c=proportion of class c
   - μ_c=sample mean of class c
   - Σ_c=sample cov of class c

Evaluation and Results

N-Fold Cross Validation: Since the we only have 198 labeled data point, we don’t have efficient data to separate training set and testing set. Specifically, the original labeled sample is randomly partitioned into 10 equal sized subsample. We use 9 of them and the whole unlabeled data set to train our model, and use the remaining data points as test set. Then we repeat this process 10 times to go over all data points as test data.

Sensitivity and Specificity: Also notice that the labeled data is unbalanced (47 Recursion and 151 Non-Recursion), we will use confusion matrix to evaluate our result, not a simple success rate.

$$sensitivity = \frac{number\ of\ True\ Positives}{number\ of True\ Positive + number\ of\ False\ Negative}$$

$$specificity = \frac{number\ of\ True\ Negatives}{number\ of True Negatives + number\ of\ False\ Positive}$$

Benchmark Results:

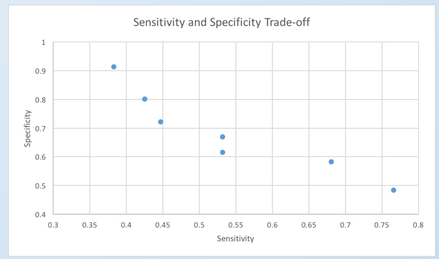
	Training (%)	Testing (%)
Sensitivity	27.59	27.78
Specificity	90.30	91.67

Our Results (Equal Sample weights):

	$\hat{y} = R$	$\hat{y} = N$	success rate(%)
$y = R$	18	29	38.30 (sensitivity)
$y = N$	13	138	91.39 (specificity)

We see that without loss of accuracy on specificity, the sensitivity is largely improved.

By altering the sample weights on loss function, we can have the following sensitivity and specificity trade-off plot.



REFERENCES

[1] Diana Dumitru. ”Prediction of recurrent events in breast cancer using the Naive Bayesian classification” Annals of University of Craiova (2009), Vol 36(2).

[2] Chapelle, Olivier; Schlkopf, Bernhard; Zien, Alexander (2006). Semi-supervised learning. Cambridge, Mass: MIT Press.ISBN978-0-262-03358-9.