Estimating Risks of Developing Second Cancers

by

Ge Yang

A research paper
presented to the University of Waterloo
in partial fulfillment of the
requirement for the degree of
Master of Mathematics
in
Computational Mathematics

Supervisor: Prof. Ali Ghodsi and Prof. Sivabal Sivaloganathan

Waterloo, Ontario, Canada, 2014

I hereby declare that I am the sole author of this report. This is a true copy of the report, including any required final revisions, as accepted by my examiners.

I understand that my report may be made electronically available to the public.

Abstract

The objectives of this paper are to discuss how state-of-the-art feature selection and classification techniques can be used to develop tools to predict risks of developing second cancer based on basic patient information and initial cancer diagnosis. We present a feature extraction method to discover and collect useful properties from the first cancer radiation therapy record based on visualization and clinical experience. We further select and verify the extracted features using a greedy column subset selection algorithm, supervised principle component analysis and greedy column subset selection via Hilbert schmidt independence Criteria(HSIC) algorithms. Finally, we adopted classification methods i.e. logistic regression and support vector machine to verify the accuracy on predicting second cancer risk.

Acknowledgements

I would like to thank my supervisor, Professor Ali Ghodsi and Professor Sivabal Sivaloganathan for their support and guidance. I would also like to thank Professor Mohammad Kohandel and Satya Kumar Manem for our valuable discussions. Thanks to my friends in CM lab as well as Victor. Last but not least, my gratitude to my parents is far beyond description.

Dedication

This is dedicated to the one I love.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Over the past decades, radiation therapy, chemotherapy and operations have significantly increased the survival rate of cancer patients. However, radiation therapy has come at the cost of some long term side effects as a result of the radiation treatment. One significant concern for cancer survivors is the risk of developing second cancers at later stages in their lives. Radiation therapy was recognized as a potential cause of second cancer many years ago[5,7]. The risk of second cancer increases with higher volumes of drug doses[1,3]. In other words, second cancer risks are much related with the dose-intensity of the radiations. According to the previous study results[1], the risk of developing a radiation-induced second lung cancer is approximately one in every 200 women previously treated with postoperative radiation therapy[1]. Radiotherapy patients typically receive high doses of radiation in the proximal organs of the tumor and low doses far away from the tumor. Researchers have also shown that among women who have survived breast cancer for at least five years, the relative risk of subsequently developing a lung cancer is increased by 8.5% per dose delivered to the lung which is proximal to the breast during the radiation process[3,11]. This means that in the future, significant effort should be put into adjusting the high doses delivered to areas outside the main field of radiation.

Surveys have also shown that basic information about the patient such as age, whether or not she has a smoking history, or has received chemotherapy all account for reasons of developing second cancer later in their lives potentially[5,11].

We focus on estimating and quantifying the risk of developing second cancer based on initial cancer treatment and basic patient information via machine learning techniques. We will first discuss some techniques on extracting useful properties from initial radiation treatment records of various lengths. Since all features are extracted based on visualization

and clinical experience, we further use various feature selection algorithms to select the most representative ones. Finally we will perform classification using logistic regression and support vector machine on whether or not the patient will develop second cancer based on the features selected. With this second cancer prediction tool being developed, hopefully the physicians could reconstruct the ways that patients had been treated for the original cancer and to adjust the amount of radiation that was delivered to the areas nearby the main cancerous organ.

The remainder of the essay is structured as follows: Chapter 2 provides a complete description of the problem and methodologies on extracting features from raw patient data by visualization and clinical hypotheses. In Chapter 3, we introduce and compare the main methods for feature selection, i.e. Greedy Column Subset Selection, Supervised Principle Component Analysis, Greedy Column Subset Selection via Hilbert-schmidt independence Criterion (HSIC) etc. Chapter 4 evaluates the feature selection results based on classification accuracy.

Chapter 2

Feature Extraction

2.1 Data Description

In this section we will introduce the data for experiments. We focus our experiments on the radiation records of three types of tissues: breast, lung and thyroid. The data was received from Princess Margaret Hospital[15].

For every single patient, the initial cancer radiation treatment is recorded as three lists of Dose Volume pairs which represents the doses of radiation applied on a specific volume of breast, lung and thyroid. Furthermore, we use binary data to indicate whether this patient has a smoking history, whether the patient has received chemotherapy previously. Also ages of the patients are represented as integers.

We have the information indicating whether the patient has received second cancer and what type of cancer the patient received. For our first experiment for breast cancer, We denote it as the label for female patient i, $i = 1 \dots 69$ as

$$y_i = \begin{cases} 1 & \text{if patient } i \text{ received second breast cancer} \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

All labels are already known.

In this problem, our objective is to first of all, extract useful features from the Dose Volume lists of various lengths for all patients, so that each patient x_i can be represented as an $m \times 1$ column vector, where m is the number of features. Then we want to find a function $f : R^n \to \{0, 1\}$ such that y_i is well-approximated by $f(x_i)$, for $i = 1 \dots n$,

n is the number of patients. The next step would be feature selection, which involves a NP-hard combinatorial optimization problem. Most feature selection methods depend on heuristics to obtain a subset of relevant features in a manageable time. Lastly, we will perform classification based on the features selected in the previous step.

2.2 Feature Extraction by Visualization and Clinical Hypotheses

Feature extraction from the Dose Volume lists is defined from a clinical aspect to capture the properties of changes of volumes over the dose range. We use the Dose Volume Histogram(DVH) as a tool to display the dose distribution in a two-dimensional graph. DVH can also be used to validate the dose constraints located in the vicinity of the tumor volume. There are two types of DVHs: cumulative and differential. Cumulative DVH has dose(Gy) as the x-axis and volume(cm^3) as y-axis. Each data point on the graph represents that the patient has received this much doses(Gy) on the corresponding volume(cm^3) of tissues, either breast, lung or thyroid. A sampled Cumulative DVM[15] is provided in Figure 2.1.

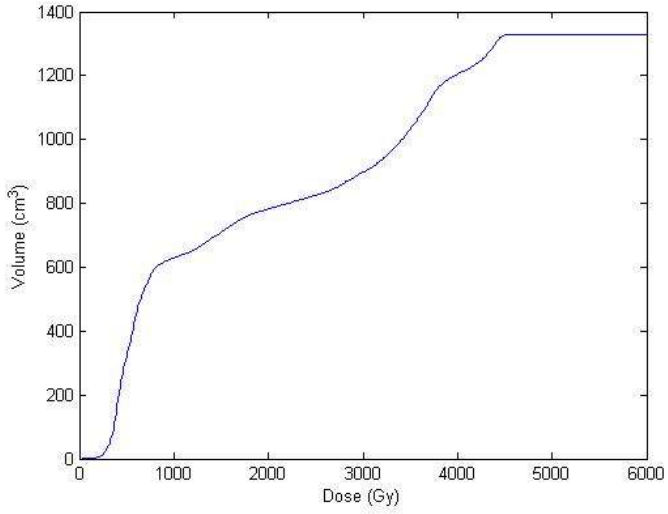


Figure 2.1: Cumulative Dose Volume Histogram

As opposed to cumulative DVH, the differential DVH is a histogram that each bin

represents the volume received per unit of dose. We provide a sampled differential DVH[15] in Figure 2.2.

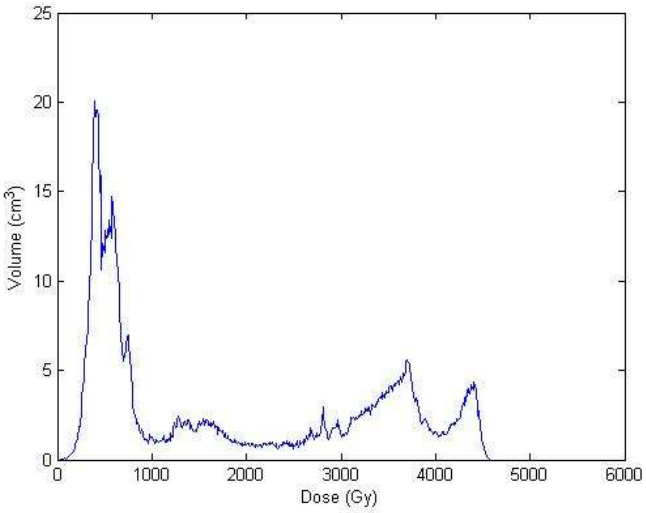


Figure 2.2: Differential Dose Volume Histogram

The Differential DVH indicates the following two peaks: one occurs at low dose and high volume; the other one occurs at high dose and low volume. From a clinical perspective, we believe that the two peaks are important features for prediction. Also, maximum dose and volume, weighted average dose-intensity (defined as total doses divided by total volume of the list) are important. For every single patient, the above two quantities are computed for all three lists of breast, lung and thyroid accordingly.

Furthermore, we are interested in discovering the differences between the DVHs of patients who got second cancer versus those who did not get second cancer. Randomly sampled female patients' DVHs[15] are provided in Figure 2.3 and 2.4.

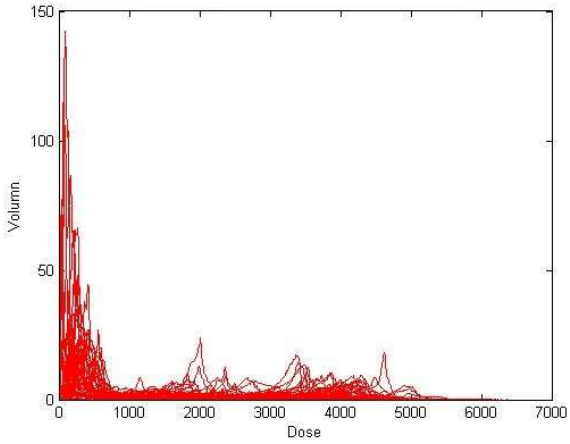


Figure 2.3: Female with Second Breast Cancer

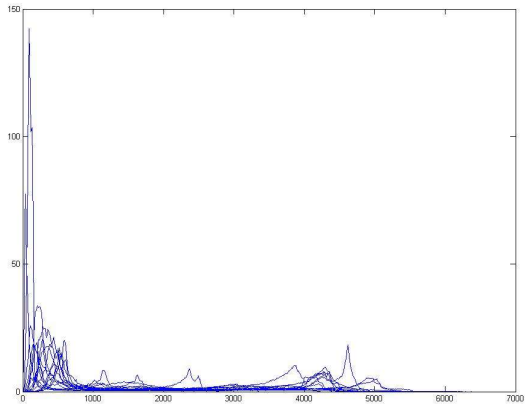


Figure 2.4: Female with No Second Breast Cancer

The second figure has less fluctuations on volumes between doses ranging from 1000 to 5000, thus we propose to extract the variance of volumes between the above dose range.

As can be shown from the two figures, there is a significant difference between total volumes over all dose ranges. Therefore, we further extracted total volume as a feature. We also conducted covariance test, which shows that total variance has a large correlation coefficient with the binary labels.

The summaries of features for female and male separately are as follows:

Table 2.1: Feature Extraction Summary for Female Breast Cancer Prediction

Index	Feature Name	Data Type
1	Weighted Average on Breast	Float
2	Maximum Dose on Breast	Float
3	Variance of Volume on Breast between Dose Range [1000,4000]	Float
4	Total Volume on Breast	Float
5	Weighted Average on Lung	Float
6	Total Volume on Lung	Float
7	Weighted Average on Thyroid	Float
8	Total Volume on Thyroid	Float
9	Age	Integer
10	25 Quartiles of Total Volume	Float
11	75 Quartiles of Total Volume	Float

Table 2.2: Feature Extraction Summary for Male Thyroid Cancer Prediction

Index	Feature Name	Data Type
1	Weighted Average on Lung	Float
2	Maximum Dose on Lung	Float
3	Variance of Volume on Thyroid between Dose Range [1000,4000]	Float
4	Total Volume on Thyroid	Float
5	Total Volume on Lung	Float
6	Age	Integer
7	25 Quartiles of Total Volume	Float
8	75 Quartiles of Total Volume	Float

Chapter 3

Feature Selection

We aim to select a subset of relevant features from all features we extracted in the previous section. Relevant features are those which provide useful information for the classification model. They also improve the model interpretability and provide helpful suggestions for future radiation therapy. In this section, we will first introduce a recent features selection method called the greedy column subset selection(GCSS) by connecting this idea with the famous principle component analysis(PCA) method. Further, we will discuss other supervised ways for features selection, i.e. supervised principle component analysis(SPCA) and Sparse sparse supervised principle component analysis(SSPCA). We propose a supervised greedy column subset selection(SGCSS) via the Hilbert-schmidt independence criterion(HSIC) method based on the idea of both greedy column subset selection and SPCA.

3.1 Notations

Throughout the paper, the following notations are used.

For matrix $A \in \mathbb{R}^{n \times d}$,

- A_{ij}: (i, j)-th entry of A
- $A_{i:}$: i-th row of A
- $A_{:j}$: j-th column of A

- $A_{S:}$: the sub-matrix of A which consists of set S of rows
- $A_{:S}$: the sub-matrix of A which consists of set S of columns
- $A_{i,1:k}$: the first k coordinates of the i-th row of A
- $||A||_F$: the Frobenius norm of A, $||A||_F = \sum_{i,j} A_{ij}^2$

3.2 Greedy Column Subset Selection

Historically, Principal Components Analysis (PCA)[6] has been the predominant dimensionality reduction technique, and it has been widely used in all scientific domains. PCA looks for an orthogonal transformation to convert the data from high dimensional space to a linearly uncorrelated low dimensional space from the most informational viewpoint . PCA is an unsupervised dimensionality reduction technique. The input parameters are the coordinates of the data points and the number of dimensions that will be retained in the projection.

We start introducing the feature selection methods with a brief description of PCA. Assume that we are given a dataset consisting of n objects, described with respect to d features or, equivalently, an $n \times d$ matrix A. Let $k \ll d$ be the dimensionality of the space that we seek to embed our data in, and assume that the columns (features) of A are mean-centered. Then, PCA returns the top k singular vectors of A corresponding to the top k largest eigenvalues (an $n \times k$ matrix U_k) and projects the data onto the k-dimensional subspace spanned by the columns of U_k. Let $P_{U_k} = U_k^T U_k$ be the projection onto the subspace spanned by the columns of U_k. It is known[4] that the reconstruction error in the F-norm is optimal i.e.

$$||A - P_{U_k} A||_F$$

is minimized over all possible k-dimensional subspaces.

We look for efficient unsupervised feature selection algorithms that identify a subset of exactly k (out of the d) features, such that if PCA is applied only on these k features, then the resulting embedding is close to the embedding that emerges when PCA is applied on all n features. let us denote S as the $n \times k$ data matrix that includes only those columns of A that correspond to the chosen features. We measure the error of a feature selection strategy for PCA by comparing the residual

$$||A - P^{(S)} A||_F$$

Here $P^{(S)}$ denotes the projection onto the k-dimensional space spanned by the columns of S. Equipped with this error minimization, our problem is equivalent to the so-called Column Subset Selection(CSS) Problem [2]:

Definition 3.2.1. *Given a matrix $A \in \mathbb{R}^{n\times d}$ and a positive integer k, pick k columns of A forming a matrix $S \in \mathbb{R}^{n\times k}$ such that the residue $||A - P^{(S)}A||_F^2$, is minimized over all possible $\binom{n}{k}$ choices for the matrix S.*

Solving the CSS problem exactly is a NP-hard combinatorial problem, and thus research has historically focused on computing approximate solutions to the CSS problem. A. Farahat et al. [2] proposed a greedy column subset selection approach that works as follows:

To solve the optimization problem

$$\mathbb{L} = \underset{S}{\operatorname{argmin}}\ F(S)$$

where the F(S) is the reconstruction error, defined as

$$F(S) = ||A - P^{(S)}A||_F^2$$

and $P^{(S)}$ has the closed form solution as

$$P^{(S)} = A_{:S}^T(A_{:S}^T A_{:S}^T)^{-1}A_{:S}^T$$

A naive implementation of this greedy algorithm can be done by initially setting the selection set as empty and then iteratively calculating the reconstruction error for each candidate feature and then greedily adding in the feature with the smallest reconstruction error, see Algorithm 1.

However, this implementation takes $O(n^2d^2)$ floating point operations, which is computationally expensive. A much more efficient approach is provided by recursively calculating the reconstruction error.

Theorem 3.2.2. *Given a set of columns S. For any $P \subset S$, $P^{(S)} = P^{(P)} + R^{(R)}$, where $R^{(R)} = E_{:R}(E_{:R}^T E_{:R})^{-1}E_{:R}^T$ is a projection matrix which projects the columns of $E = A - P^{(P)}A$ onto the span of the subset $R = S\backslash P$ of columns.*

Proof. Define $D = A_{:S}^T A_{:S}$. The projection matrix $P^{(S)}$ can be written as $P^{(S)} = A_{:S}D^{-1}A_{:S}^T$. P is a subset of S. Without loss of generality, the columns and rows of $A_{:S}$ and D can be rearranged such that the first sets of rows and columns correspond to P. Let

Algorithm 1 Naive Implementation of Greedy Column Subset Selection

Input: $A \in \mathbf{R}^{n \times d}$, $Y \in \mathbf{R}^{n \times 1}$, kernel matrix of target variable L and training data size n
$S := empty$
$R := 1 : d$
for $i = 1 : k$ **do**
  $l = \underset{i \in R}{\text{argmin}} \; ||A - P^{(S \cup \{i\})} A||_F^2$
  $S = S \cup i$
  $R = R \setminus l$
end for

$S = D_{RR} - D_{PR}^T D_{PP}^{-1} D_{PR}$ be the Schur complement[2] of D_{PP} in D, where $D_{PP} = A_{:P}^T A$: P, $D_{PR} = A_{:P}^T A_{:R}$ and $D_{RR} = A_{:R}^T A_{:R}$. Using the block-wise inversion formula[2],

$$D^{-1} = \begin{bmatrix} D_{PP}^{-1} + D_{PP}^{-1} D_{PR} S^{-1} D_{PR}^T D_{PP}^{-1} & -D_{PP}^{-1} D_{PR} S^{-1} \\ -S^{-1} D_{PR}^T D_{PP}^{-1} & S^{-1} \end{bmatrix} \tag{3.1}$$

Then the projection matrix $p^{(S)}$ can be simplified to

$$P^{(S)} = A_{:P} D_{PP}^{-1} A_{:P}^T + (A_{:R} - A_{:P} D_{PP}^{-1} D_{PR}) S^{-1} (A_{:R}^T - D_{PR}^T D_{PP}^{-1} A_{:P}^T) \tag{3.2}$$

The first term is just the projection matrix $P^{(P)}$ which projects vectors onto the span of the subset P of columns. The second term can be simplified as follows: Define E as an $d \times n$ residual matrix, i.e. $E = A - P^{(P)} A$. Then the submatrix $E_{:R}$ is

$$\begin{aligned} E_{:R} = A_{:R} - P^{(P)} A_{:R} &= A_{:R} - A_{:P} (A_{:P}^T A_{:P})^{-1} A_{:P}^T A_{:R} && (3.3) \\ &= A_{:R} - A_{:P} D_{PP}^{-1} D_{PR} && (3.4) \end{aligned}$$

Since the projection matrix is idempotent, i.e. $P^{(P)} P^{(P)} = P^{(P)}$, and

$$\begin{aligned} E_{:R}^T E_{:R} &= (A_{:R} - P^{(P)} A_{:R})^T A_{:R} - P^{(P)} A_{:R} && (3.5) \\ &= A_{:R}^T A_{:R} - A_{:R}^T P^{(P)} A_{:R} && (3.6) \\ &= D_{RR} - D_{PR}^T D_{PP}^{-1} D_{PR} = S && (3.7) \end{aligned}$$

Then Equation (3.2) can be re-written as

$$\begin{aligned} P^{(S)} &= P^{(P)} + E_{:R} (E\text{:}\, R^T E_{:R})^{-1} E_{:R}^T && (3.8) \\ &= P^{(P)} + R^{(R)} && (3.9) \end{aligned}$$

where $R^{(R)}$ is the projection matrix which projects vectors on the the span of columns of R.

□

By definition, $F(S) = ||B - P^{(S)}B||_F^2$. Using theorem 3.2.2, $P^{(S)}A = P^{(P)}A + R^{(R)}A$. The term $R^{(R)}A$ is equal to $R^R F$ as

$$E_{:R}^T F = E_{:R}^T A - E_{:R}^T P^{(P)} A \tag{3.10}$$

Thus

$$\begin{aligned} F(S) &= ||A - P^{(P)}A - R^R F||_F^2 && (3.11)\\ &= ||F - R^R F||_F^2 && (3.12)\\ &= tr((F - R^R F)^T (F - R^R)) && (3.13)\\ &= tr(F^T F - F^T R^R F) && (3.14)\\ &= ||F||_F^2 - ||R^R F||_F^2 && (3.15)\\ &= F(P) - ||R^R F||_F^2 && (3.16) \end{aligned}$$

Using Theorem 3.2.2, we can show that at iteration t, find column i such that,

$$p = \underset{i}{\text{argmin}}\ \mathbb{F}(S \cup \{i\}) = \underset{i}{\text{argmax}}\ ||R^{(i)}F||_F^2$$

where S is the set of features selected in the previous $i-1$ iterations.

Now $||R^{(\{i\})}F||_F^2$ can be further simplified to

$$\begin{aligned} ||E_{:i}(E_{:i}^T E_{:i})^{-1} E_{:i}^T F||_F^2 &= tr(F^T E_{:i}(E_{:i}^T E_{:i})^{-1} E_{:i}^T F) && (3.17)\\ &= \frac{||F^T E_{:i}||^2}{E_{:i}^T E_{:i}} && (3.18)\\ &= \frac{||H_{:i}^2||}{G_{ii}} && (3.19) \end{aligned}$$

A memory efficient way of feature selection criterion can be described in terms of H and G as

$$p = \underset{i \in R}{\text{argmax}}\ \frac{||H_{:i}^2||}{G_{ii}}$$

To further reduce the space and time complexity, we want to compute H and G recursively. At iteration t, define $\delta = G_{:l}$ and $\omega = G_{:l}/\sqrt{G_{ll}} = \delta/\sqrt{\delta_l}$. $G^{(t+1)}$ can be computed by

$$\begin{aligned} G^{(t+1)} &= (G - \frac{G_{:l}G_{:l}^T}{G_{ll}})^{(t)} && (3.20)\\ &= (G - \omega\omega^T)^{(t)} && (3.21)\\ &= A^T A - \sum_{i=1}^{t} (\omega\omega^T)^{(i)} && (3.22) \end{aligned}$$

The following theorem provides a recursive formula without computing E and G explicitly. Detailed proofs can be found in [2].

Theorem 3.2.3. *Let $f_i = ||H_{:i}||^2$ and $g_i = G_{ii}$ be the numerator and denominator of criterion (3.15) for feature i, respectively, $f = [f_i]_{i=1...n}$ and $g = [g_i]_{i=1...n}$. Then*

$$f^{(t)} = (f - 2(\omega \circ (A^T B v - \sum_{r=1} t - 2(v^{(r)T} v)\omega^{(r)})) \tag{3.23}$$

$$+ \quad ||v||^2(\omega \circ \omega))^{(t-1)} \tag{3.24}$$

$$g^{(t)} = (g - (\omega \circ \omega))^{(t-1)} \tag{3.25}$$

Note that we can also apply the Greedy Column Subset Selection(GCSS) method in a more generalized way, i.e. We select a subset of columns from a source matrix A that best approximate the span of a target matrix B. In this way, B can be the label matrix, which implies that GCSS can be used for supervised learning.

The optimization problem can be reformulated as:

$$\underset{S}{\text{argmin}}\ \mathbb{F}(S)$$

where

$$\mathbb{F}(S) = ||B - P^{(S)}B||_F^2 \tag{3.26}$$

$$P^{(S)} = A_{:S}(A_{:S}^T A_{:S})^{-1} A_{:S}^T \tag{3.27}$$

The generalized Greedy Column Subset Selection algorithm summary is provided in Algorithm 2.

3.3 SPCA for Feature Selection

Supervised Principle component analysis proposed by E. Barshan et al.[10] takes the labels into consideration for projection. The Hilbert-schmidt independence criterion(HSIC) is a measure of the dependence between two variables. The two variables are independent if and only if any continuous function of the two variables are uncorrelated. Let $\mathcal{F}$ and $\mathcal{G}$ be the separable Reproducing kernel Hilbert Space(RKHS) of real-valued functions from

Algorithm 2 Greedy Column Subset Selection

Input: source matrix $A \in \mathbb{R}^{n \times d}$, B is the label matrix
Note: if label matrix B are not given, then input both source matrix and label matrix as A
$S :=$ empty
$R := 1 : d$
Initialize $f_i^{(0)} = ||A^T A_{:i}||^2, g_i^{(0)} = A_{:i}^T A_{:i}$
for $i = 1 : k$ **do**
  $l = \underset{i \in R}{\operatorname{argmin}}\ f_i^{(t)} / g_i^{(t)}$
  $S = S \cup l$
  $R = R \setminus l$ $\delta^{(t)} = A^T A_{:l} - \sum_{r=1}^{i-1} \omega_l^{(r)} \omega^{(r)}$
  $\omega^{(t)} = \delta^{(t)} / \sqrt{\delta_l^{(t)}}$
  $f^{(t)} = (f - 2(\omega \circ (A^T B v - \sum_{r=1} t - 2(v^{(r)T} v)\omega^{(r)})) + ||v||^2 (\omega \circ \omega))^{(t-1)}$
  $g^{(t)} = (g - (\omega \circ \omega))^{(t-1)}$
end for
Ouput S

data x and label y to $\mathbb{R}$ with universal kernels $K(.,.) and L(.,.)$ respectively[10]. HSIC can be described in terms of kernel functions:

$$\begin{aligned} HSIC(P_{x,y}, \mathcal{F}, \mathcal{G}) &= E_{x,x',y,y'}[K(x,x')L(y,y')] \\ &+ E_{x,x'}[K(x,x')]E_{y,y'}[L(y,y')] \\ &- 2E_{x,y}[E_{x'}[(x,x')]E_{y'}[L(y,y')]] \end{aligned}$$

where $P_{x,y}$ is the joint probability distribution of random variables X and Y. $E_{x,x',y,y'}$ is an expression over (x, y) and (x', y') which are random variables drawn independently from $P_{x,y}$. Let us define the set $\mathcal{Z}$ which consists of n visualizations$\{(x_1, y_1), \ldots, (x_n, y_n)\} \subseteq \mathcal{XY}$ taken independently according to distribution $P_{\mathcal{XY}}$. An empirical estimation of HSIC over a finite number of samples is

$$HSIC(\mathcal{Z}, \mathcal{F}, \mathcal{G}) = (n-1)^{-2} tr(KHLH)$$

where $H, K, L \in \mathbb{R}^{n \times n}, K_{ij} = k(x_i, x_j), L_{ij} = l(y_i, y_j)$, and H is the centering matrix defined as $H_{ij} = 1 - n^{-1} ee^T$. If one of the kernels, say L is centralized, then $HLH = L$ and the fomula of HSIC estimation can be simplified as $tr(KL)$. In order to maximize the dependence between two random variables, we want to increase the value of the estimate of

HSIC, i.e. $tr(HKHL)$. Assume we have a data matrix of size $n \times d$ and label matrix of size $n \times l$. The problem can be formulated as finding the subspace $U^T X^T$ such that the HSIC value between variables $U^T X^T$ and Y is maximized. We need to maximize $tr(HKHL)$ wjere K is the kernel of $U^T X^T$, (e.g. $XUU^T X^T$) and L is the kernel of Y, (e.g. YY^T), then this objective function can be formulated as

$$\begin{aligned} tr(HKHL) &= tr(HXUU^T X^T HL) && (3.28) \\ &= tr(U^T XHLHX^T U) && (3.29) \end{aligned}$$

The optimization problem can be written as the following:

$$\begin{aligned} &\underset{U}{\text{argmax}}\ tr(U^T X^T HLHXU) \\ &\text{such that } U^T U = I \end{aligned}$$

This optimization problem has a closed form solution. Let us denote $Q = X^T HLHX$. The optimal solution is $U = [u_1, u_2, \dots u_d]$, where u_i is the eigenvector corresponding to the i-th largest eigenvalue. u_1 is the direction where data makes the maximum variance.

However, in many cases, data are not linearly separable. Non-linear transformations are required to successfully apply the learning algorithm. We can use the kernel to capture the similarity measure between any two data points. The non-linear mapping function maps the feature from x to $\phi(x)$, i.e. the non-linear kernel for feature matrix x is $K = \phi(x)^T \phi(x)$. Then this allows us to formulate supervised PCA as follows:

$$\begin{aligned} &\underset{\beta}{\text{argmax}}\ tr(U^T \phi(x)^T HLH\phi(x)U) \\ &\text{such that } U^T U = I \end{aligned}$$

The solution can be found by performing SVD on $\phi(x)^T HLH\phi(x)$. Note that for kernel functions, we can efficiently compute $\phi(x)\phi(x)^T$ without computing $\phi(x)$ explicitly. In order to fully take advantage of this, we can further represent the transformation matrix U as a linear combination of the projected data points[10], $U = \phi(x)^T \beta$. Thus we can rewrite the objective function as

$$\begin{aligned} &\underset{\beta}{\text{argmax}}\ (\beta^T KHLHK\beta) \\ &\text{such that } \beta^T K\beta = I \end{aligned}$$

The solution can be found by performing SVD on matrix $Q = KHLHK$. The eigenvector of Q corresponding to the largest eigenvalue provides a direction which is the maximum

variation of the data. For feature selection purposes, we only select k meaningful dimensions, i.e. the features corresponding with the top k coordinates ranked by taking the absolute values of the coordinates.

This Supervised PCA for feature selection algorithm is summarized in Algorithm 3.

Algorithm 3 Supervised PCA for Feature Selection

Input: $X \in \mathbf{R}^{n \times d}$, $Y \in \mathbf{R}^{n \times 1}$
Compute kernel matrix K of data matrix X
Compute kernel matrix L of target variable Y
$H := 1 - n^{-1}ee^T$
$Q := XHLHX^T$ for linear kernel
Or: $Q := KHLHK$ for non-linear kernel
Compute the first eigenvector u of Q corresponding with the largest eigenvalue.
Output indices corresponding with the top k absolute values in u

3.4 Sparse SPCA for Feature Selection

In particular, if we have a large amount of features whereas we only want to select a few, then we can perform sparse SPCA which provides sparse eigenvectors for use while still explaining most of the variance present in the data. we can perform sparse singular value decomposition on the kernel matrix. D. Witten et al.[8] proposed penalized matrix decomposition(PMD) method. Consider the kernel matrix Q of dimension $n \times n$. The singular value decomposition(SVD) of Q can be written as the following:

$$Q = UDV^T \tag{3.30}$$

$$\begin{aligned} \text{such that } U^TU &= I_n && (3.31) \\ V^TV &= I_n && (3.32) \end{aligned}$$

The classical rank-one downdating algorithm is Jordan's algorithm to compute SVD. See Algorithm 4.

The inner While loop of the Jordan SVD algorithm is power iteration. If the matrix is symmetric, then we only need to keep multiplying u by the matrix A until u converges.

Algorithm 4 Jordan SVD

Input: $X \in \mathbf{R}^{m \times n}, k \leq min(m, n)$
for $i = 1 : k$ **do**
  Select random non-zero vector u of dimension $n \times 1$
  $u = \frac{u}{||u||}$
  while u, d, v not converge **do**
    $v = \frac{A^T u}{||X^T u||}$
    $u = \frac{Av}{||Xv||}$
    $d = ||Xv||$
  end while
  $A = A - duv$
  $U(:, i) = u$
  $V(:, i) = v$
  $D(i, i) = d$
end for
Output U, D, V

The PMD method enforces sparsity constraints on U and V while applying SVD. For each inner For loop, the rank-one PMD can be formulated as the following optimization problem:

$$\underset{d,u,v}{\text{argmin}} \; ||X - duv^T||_F^2 \tag{3.33}$$

$$\begin{aligned} \text{such that} & & & (3.34) \\ ||u||_2^2 &= 1, & & (3.35) \\ ||v||_2^2 &= 1, & & (3.36) \\ P_1(u) &\leq \alpha_1, & & (3.37) \\ P_2(v) &\leq \alpha_2, & & (3.38) \\ d &\geq 0 & & (3.39) \end{aligned}$$

where u is a column of U, v is a column of V, d is a diagonal element of D. P_1 and P_2 are the penalty functions that can take a variety of forms[7,8].

$$
\begin{aligned}
||X - UDV^T||_F^2 &= tr((X - UDV^T)^T(X - UDV^T)) && (3.40)\\
&= |||X||_F^2 - 2tr(VDU^TX) + tr(VDDV^T) && (3.41)\\
&= ||X||_F^2 - 2\sum_{k=1}^{K} d_k u_k^T X v_k + \sum_{k=1}^{K} d_k^2 && (3.42)
\end{aligned}
$$

Thus for $K = 1$, u and v satisfying the optimization problem (3.14) also satisfy the following:

$$\underset{u,v}{\text{argmax}}\ u^T X v \tag{3.43}$$

$$\text{such that } ||u||_2^2 = 1, ||v||_2^2 = 1, P_1(u) \leq \alpha_1, P_2(v) \leq \alpha_2 \tag{3.44}$$

and $d = u^T X v$. The solution satisfying (3.18) also satisfies (3.14) as long as α_1 and α_2 are chosen appropriately[8]. As shown from experiments, $1 \leq \alpha_1 \leq \sqrt{m}$, $1 \leq \alpha_2 \leq \sqrt{n}$.

Specifically, consider the optimization problem

$$\min_u\ -u^T a \text{ such that } ||u||_2^2 = 1, ||u||_1 \leq \sqrt{n} \tag{3.45}$$

where a is a constant vector. This is equivalent to

$$\min_u\ -u^T a \text{ such that } ||u||_2^2 = 1, -||u||_1 \geq \sqrt{n} \tag{3.46}$$

We first rewrite the criterion using Lagrange multipliers:

$$L(u, \lambda, \delta) = -u^T a + \lambda||u||_2^2 + \delta||u||_1 \tag{3.47}$$

The Karush-Kuhn-Tucker conditions[5] consist of follows:

$$\frac{\partial L(u, \lambda, \delta)}{\partial u} = -a + 2\lambda u + \delta T \tag{3.48}$$

$$\text{where } T_i = \text{sign}(u_i) \text{ if } u_i \neq 0; \text{ otherwise, } T_i \in [-1, 1].$$

$$
\begin{aligned}
\lambda(||u||_2^2 - 1) &= 0 && (3.49)\\
\delta(||u||_1 - \sqrt{n}) &= 0 && (3.50)\\
\lambda, \delta &\geq 0 && (3.51)
\end{aligned}
$$

Denote S as the soft thresholding operator, i.e. $S(a,c) = \text{sign}(a)(|a| - c)_+$, where x_+ is defined to equal x if $x > 0$ and 0 if $x \leq 0$. If $\lambda > 0$, we have $u = \frac{S(a,\delta)}{2\lambda}$. In order to have $||u||_2 = 1$, λ is chose to satisfy $u = \frac{S(a,\delta)}{||S(a,\delta)||_2}$. $\delta = 0$ if this results in $||u||_1 \leq \sqrt{n}$; otherwise, δ is chosen so that $||u||_1 = \sqrt{n}$. In general, δ can be chosen by binary search.

The above (3.14) optimization problem is called a rank-one PMD. The iteration procedure is summarized in Algorithm 5:

Algorithm 5 PMD by Rank-one Downdate

Input: $X \in \mathbf{R}^{m \times n}, k \leq min(m,n)$
for $i = 1 : k$ **do**
  Select random non-zero vector u of dimension $n \times 1$
  $u = \frac{u}{||u||}$
  while u, d, v not converge **do**
    $v = \underset{v}{\text{argmax}}\ u^T X v$, such that $||v||_2^2 \leq 1, P_2(v) \leq \alpha_2$
    $u = \underset{u}{\text{argmax}}\ u^T X v$, such that $||u||_2^2 \leq 1, P_1(u) \leq \alpha_1$
  end while
  $A = A - duv$
  $U(:,i) = u$
  $V(:,i) = v$
  $D(i,i) = u^T X v$
end for
Output U, D, V

For sparse SPCA, given that the kernel matrix Q is symmetric positive definite, we modify the Rank-one PMD algorithm such that the first eigenvector is sparse. We take the ℓ_1-norm of u and v as the penalty function, i.e. $||u||_2^2 \leq 1, ||u||_1 \leq \sqrt{n}$. Detailed summary of Sparse SPCA is in Algorithm 6.

3.5 SPCA via Truncated Power Iteration for Feature Selection

The previous two methods only use the first eigenvector for feature selection. In fact, there are two problems with it:

Algorithm 6 Sparse SPCA for Feature Selection

Input: $X \in \mathbf{R}^{n \times d}$, $Y \in \mathbf{R}^{n \times 1}$
Compute kernel matrix K of data matrix X
Compute kernel matrix L of target variable Y
$H := 1 - n^{-1}ee^T$
$Q := XHLHX^T$ for linear kernel
Or: $Q := KHLHK$ for non-linear kernel
Select random non-zero vector u of dimension $n \times 1$
$u = \frac{u}{||u||}$
while u not converge **do**
  $u = \frac{S(Xv,\delta)}{||S(a,\delta)||_2}$. $\delta = 0$, where $\delta = 0$ if this results in $||u|| \leq \sqrt{n}$
  otherwise, δ is chosen to be a positive constant via binary search such that $||u||_1 = \sqrt{n}$
end while
Output indices with non-zero values in u

- Only one eigenvector is not sufficient to capture the total variance of data.
- For Sparse SPCA, we cannot enforce the rest eigenvectors to have the same zero coordinates as the first one.

In general, we want an algorithm that can cover all of the top eigenvectors of the kernel matrix.

Algorithm 7 Power Iteration

Input: Matrix K of size $n \times n$
v:= random vector of dimension $n \times 1$
while v not converge **do**
  $v := \frac{Kv}{||Kv||}$
end while
Output v

At the t-th iteration,

$$\begin{aligned} v^t &= Kv^{t-1} = K^2v^{t-2} = \ldots = K^tv^0 && (3.52)\\ &= c_1K^te_1 + c_2K^te_2 + \ldots c_nK^te_n && (3.53)\\ &= c_1\lambda_1^te_1 + c_2\lambda_2^te_2 \ldots c_n\lambda_n^te_n && (3.54)\\ \frac{v^t}{c_1\lambda_1^t} &= e_1 + \frac{c_2}{c_1}(\frac{\lambda_2}{\lambda_1})^te_2 + \ldots + \frac{c_n}{c_1}(\frac{\lambda_n}{\lambda_1})^te_n && (3.55) \end{aligned}$$

$$(\lambda_i/\lambda_1)^t \approx 1 \text{ for } i = 1 \dots k$$
$$(\lambda_i/\lambda_1)^t \approx 0 \text{ for } i = k+1 \dots n$$

If the power iteration can be stopped early, then the result we get is a linear combination of the k eigenvectors corresponding to the top k eigenvalues of the kernel matrix[14]. Instead of only using the first principle component, we have all information of the top eigenvectors reflected in one single vector. Truncated power iteration turns out to be very effective and efficient way of feature selection compared to SPCA. The cost either in space or time of explicitly calculating all eigenvectors is replaced by only a small number of matrix-vector multiplications, which implies that SPCA via truncated power iteration can be very efficient on large scale datasets.

The summary for SPCA via power iteration algorithm is provided in Algorithm 8.

Algorithm 8 SPCA via Power Iteration for Feature Selection

Input: $X \in \mathbf{R}^{n \times d}$, $Y \in \mathbf{R}^{n \times 1}$
Compute kernel matrix K of data matrix X
Compute kernel matrix L of target variable Y
$H := 1 - n^{-1}ee^T$
$Q := XHLHX^T$ for linear kernel
Or: $Q := KHLHK$ for non-linear kernel
Select random non-zero vector u of dimension $n \times 1$
$v = \frac{v}{||v||}$
while v not converge **do**
  $v := \frac{Qv}{||Qv||}$
end while
Output indices corresponding with the top k absolute values in v

3.6 Supervised Greedy Column Subset Selection via HSIC

In case of supervised feature selection learning, the goal is to estimate a functional dependence from training data such that f predicts well on test data. The dependence function is capable of detecting desired (linear or non-linear) functional dependence between the data and the labels.

Many popular feature selection algorithms such as Pearson correlation, mutual information, these algorithms perform well for linearly separable problems. For nonlinear problems, however, the linear correlation does not necessarily provide good features. Greedy approaches, such as forward selection and backward elimination, are often used to solve this problem directly. Similar to the mutual information, HSIC is a nonparametric dependence measure, which takes into account all modes of dependence between the variables (not just linear correlation). HSIC does not require density estimation as an intermediate step. It is based on the covariance between variables mapped to reproducing kernel Hilbert spaces (RKHS). HSIC has good uniform convergence guarantees, and an unbiased empirical estimate.

In principle, the Hilbert-Schmidt independence criterion can be employed for feature selection. As we shall see, several specific choices of kernel function will lead to well known feature selection. We propose a supervised greedy column subset selection method that works by greedily select features which maximizes the dependence between source matrix and labels. In practice, we use HSIC as the measure of dependence[10].

This greedy algorithm can be implemented in two ways: forward selection tries to increase HSIC as much as possible by the inclusion of each feature, and backward elimination tries to achieve this by backward elimination[13].

Specifically for forward selection, initially we have the column selection set S as empty, in each iteration, we greedily search for a column i, which has not been selected in S yet, such that S appended by column i has the largest HSIC value with the labels. Practically, the dependence between the source and label matrix can be measured by $(n-1)^{-2}tr(KHLH)$ where K is the kernel of source matrix and L is the kernel of label matrix. A naïve implementation of Forward Supervised Greedy Column Subset Selection(Forward SGCSS via HSIC) is shown in Algorithm 9.

We can also select features via backward elimination. That is, we initially have the full set of features. Suppose we want to select k features, in each iteration, we eliminate one feature such that the rest of features can maximize HSIC and we insert this feature to set S. We keep doing the iterations until all features are eliminated. Thus in this way all features in S are sorted in a decreasing relevance pattern. We can simply take the last k elements from S. Comparing with forward GCSS via HSIC, the backward elimination method is more computational inefficient because more features are evaluated per iteration. However, backward SGCSS tends to provide better result especially for non-linear kernels since the k features finally got selected are assessed with all other features in each iteration. The summary of the Backward Supervised Greedy Column Subset Selection(Backward GCSS via HSIC) is provided in Algorithm 10.

Algorithm 9 Forward SGCSS via HSIC

Input: $A \in \mathbf{R}^{n \times d}$, $Y \in \mathbf{R}^{n \times 1}$
Compute Kernel matrix L of target variable Y
$S :=$ empty
$R := 1 : d$
$H := 1 - ee^T$, e is a vector of of length n consisting of all 1's
for $i = 1 : k$ **do**
  $l = \underset{i \in R}{\text{argmax}}\ tr(\text{kernel}(A^{(S)} \cup A_{:,i})HLH)$
  $S = S \cup l$
  $R = R \setminus l$
end for
Output S

Algorithm 10 Backward SGCSS via HSIC

Input: $A \in \mathbf{R}^{n \times d}$, $Y \in \mathbf{R}^{n \times 1}$
Compute Kernel matrix L of target variable Y
$S := 1 : d$
$R :=$ empty
$H := 1 - ee^T$, e is a vector of of length n consisting of all 1's
while S not empty **do**
  $l = \underset{i \in R}{\text{argmin}}\ tr(\text{kernel}(A^{(S)} \cup A_{:,i})HLH)$
  $R = R \cup l$
  $S = S \setminus l$
end while
Output the last k elements of S

The type of kernel function should be selected upon prior knowledge of the features. The kernel matrix L for the labels is fixed through the whole feature selection process. It can be precomputed and stored for speed up. Hence the major computation comes from repeated calculation of the kernel matrix K for the selected source data. For linear kernels, the kernel matrix can be constructed recursively, i.e.

We propose a recursive kernel reconstruction method, i.e. we can construct the kernel matrix for d features recursively by using kernel matrix for $d-1$ features.

On the first iteration, only one column is selected. The kernel matrix is an $n \times n$ matrix where each element can be computed by applying the kernel function on two one-dimensional data points. On the k-th iteration, each data point has k dimensions. For any two data point $X_{i:}$ and $X_{j:}$ which are of k dimensions. Without loss of generality, suppose the selected k features are re-arranged to the first k columns of X. The following two properties are preserved:

$$\begin{aligned} ||X_{i,1:k} - X_{j,1:k}||^2 &= \sum_{h=1}^{k} ||X_{ih} - X_{jh}||^2 && (3.56) \\ &= \sum_{h=1}^{k-1} ||X_{ih} - X_{jh}||^2 + ||x_{ik} - x_{jk}||^2 && (3.57) \\ &= ||X_{i,1:k-1} - X_{j,1:k-1}||^2 + ||x_{ik} - x_{jk}||^2 && (3.58) \end{aligned}$$

$$\begin{aligned} X_{i,1:k} \cdot X_{j,1:k} &= \sum_{h=1}^{k} X_{ih} X_{jh} && (3.59) \\ &= \sum_{h=1}^{k-1} x_{ih} x_{jh} + X_{ik} X_{jk} && (3.60) \\ &= X_{i,1:k-1} \cdot X_{j,1:k-1} + X_{ik} X_{jk} && (3.61) \end{aligned}$$

This means that for kernel functions which includes dot products, i.e. polynomial kernels, the kernel matrix can be reconstructed using Equation (3.31). Similarly, for kernel function including vector norms, i.e. Gaussian RBF kernel, we can use Equation (3.30). We save the time complexity by always recording the norms or dot products between every two data points.

A time complexity summary for some example kernels are provided in Table 3.1. All Complexities are evaluated at the k iteration of SGCSS via HSIC.

Table 3.1: Recursive Kernel Reconstruction

Kernel Type	Original Time Complexity	Refined Time Complexity
Polynomial	$O(2kn^2)$	$O(2n^2)$
Guassian	$O(3kn^2)$	$O(3n^2)$
Spline	$O(ckn^2)$	$O(cn^2)$

In the case of linear kernels, we claim that the result of SGCSS via HSIC can be reproduced within one iteration.

Claim 3.6.1. *If the kernel function is linear, then only one iteration is needed for Supervised Greedy Selection via HSIC.*

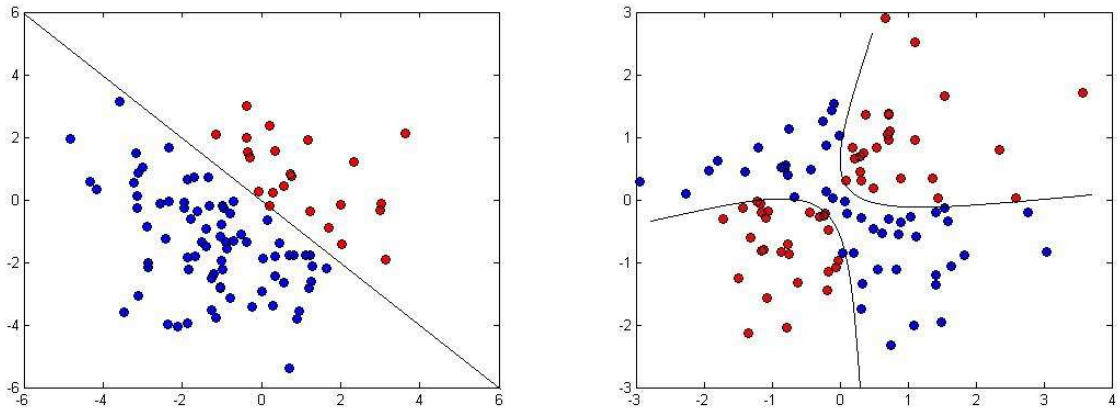
Proof. First denote $K^{(S_{:})}$ as the kernel matrix of the i^{th} column of S. initially, the first column selected by SGCSS via HSIC is the one column that has the maximum HSIC with the labels. Consider the objective function of the second iteration, if the kernel function is linear then

$$\begin{aligned} tr(HK^{(S)}HL) &= tr(H(SS^T))HL & (3.62)\\ &= tr(H(\sum_{i=1}^{k} S_{:i}S_{:i}^T)HL) & (3.63)\\ &= tr(H(K^{(S_{:1})} + K^{(S_{:2})})HL) & (3.64)\\ &= tr(HK^{(S_{:1})}HL) + tr(HK^{(S_{:2})}HL) & (3.65) \end{aligned}$$

The first term $tr(HK^{(S_{:1})}HL$ is a constant, thus we only need to find a column that has not been selected who has the largest HSIC with labels.
Therefore, maximizing $tr(HK^{(S)}HL)$ can be done by first scoring the features individually, say, feature number i, by computing $tr(HK^{(S_{:i})}HL)$. The solution of Greedy Selection via HSIC with linear kernels is the features corresponding with the top k scores. Only one iteration is needed to compute and rank the scores for all features. □

3.7 Feature Selection Results Comparison

In this section, we compare the performance of GCSS, SPCA, SGCSS via HSIC methods and other related feature selection algorithms, i.e. covariance test on synthetic data. In



(a) Linearly Separable (b) Non-Linearly Separable

Figure 3.1: Binary Classification

order to test both linear and non-linear kernels, both classification and regression problems, the following three experiments are conducted:

- Binary classification
- Multi-class classification
- Non-linear regression

We randomly generate data matrix $A \in \mathbb{R}^{n \times d}$, where n can be from $\{50, 100, 150 \dots 400\}$, $d = 15$. Each data point has 15 dimensions - only the first two dimensions are related to the prediction and the rest are just Gaussian noise. The specific settings are described in below:
Experiment 1 (Binary Classification): Generate data matrix X of size $n \times 15$ from Gaussian Normal distribution on $[-1, 1]$. Denote the label $Y = \text{sign}(X_{:1} + X_{:2})$. The first two dimensions of X with labels can be seen in Figure 3.1(a).

We also try an experiment on non-linearly separable data. Define label $Y = \text{sign}(X_{:1} \times X_{:2})$. The first two dimensions of X with labels can be seen in Figure 3.1(b).

For SGCSS via HSIC method, We use polynomial kernel $K(x, y) = (x^T y + c)^d$ for data matrix. Delta kernel is used for binary labels:

$$K(x, y) = \begin{cases} 1 & \text{x} = \text{y} \\ 0 & \text{otherwise} \end{cases} \tag{3.66}$$

Because all feature selection algorithms output the rank of the features, we plot the average rank of the first two features with respect to the number of sample size n (x-axis) for the data we generated. Note that features are ranked by decreasing importance. The average rank of the first two features is supposed to be 1.5. Figure 3.3 shows the results of binary classification in the case when data is non-linearly separable.

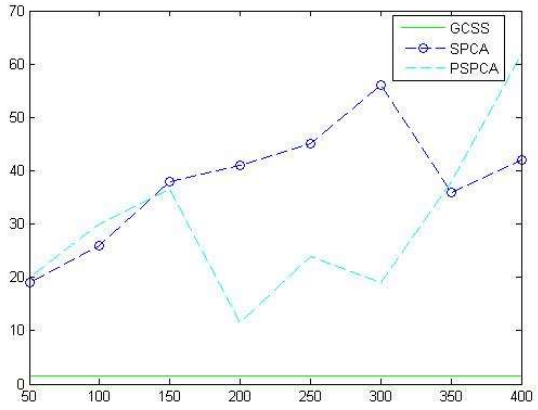


Figure 3.2: Binary Classification

The green line indicates the result of GCSS, SGCSS via HSIC and Pearson covariance test result. All GCSS, GCSS via HSIC and Pearson Correlation test successfully provides correct result. However, SPCA related methods failed. SPCA works well for data embedding but not as accurate as the other methods on feature selection.

Experiment 2: (Multi-class Classification): Generate data matrix X of size $n \times 15$ from Gaussian Normal distribution on $[-1, 1]$. Denote the label as

$$Y = \begin{cases} 0, & \text{if } \operatorname{sign}(X_{:1} + X_{:2}) = 1 \bigwedge \operatorname{sign}(X_{:1} - X_{:2}) = 1 \\ 1, & \text{if } \operatorname{sign}(X_{:1} + X_{:2}) = 1 \bigwedge \operatorname{sign}(X_{:1} - X_{:2}) = -1 \\ 2, & \text{if } \operatorname{sign}(X_{:1} + X_{:2}) = -1 \bigwedge \operatorname{sign}(X_{:1} - X_{:2}) = 1 \\ 3, & \text{if } \operatorname{sign}(X_{:1} + X_{:2}) = -1 \bigwedge \operatorname{sign}(X_{:1} - X_{:2}) = -1 \end{cases} \tag{3.67}$$

The first two dimensions of the X can be seen in Figure 3.3.

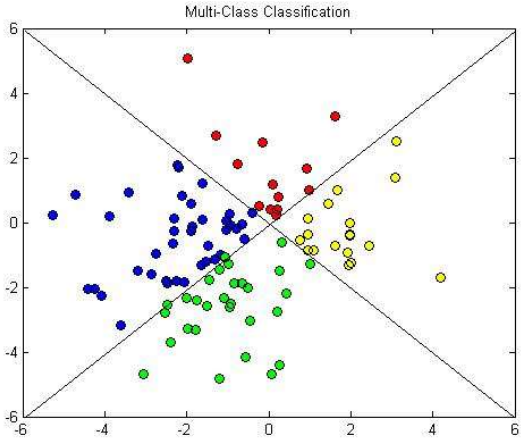


Figure 3.3: Multi-class Classification

The average rank of the first two features with respect to the number of sample size n is plotted in Figure 3.4.

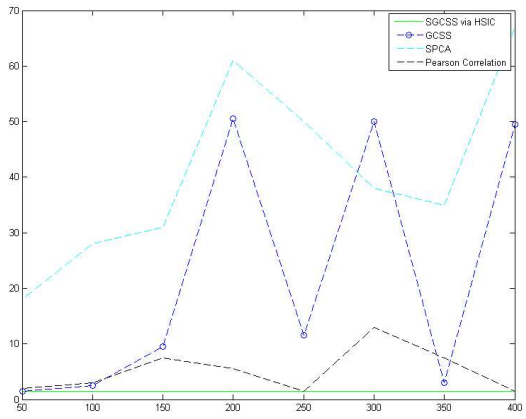


Figure 3.4: Multi-class Classification

The result shows that GCSS via HSIC provides the most accurate result.

Experiment 3: Non-linear Regression: Generate data matrix X of size $n \times 15$ from Gaussian Normal distribution on $[-1, 1]$. Denote the i-th label as

$$Y_i = X_{i1} exp(-X_{i1}^2 - X_{i,2}^2) + \epsilon \tag{3.68}$$

where ϵ is Gaussian noise for $i = 1 \ldots n$.
The first two dimensions of the X can be seen in Figure 3.5.

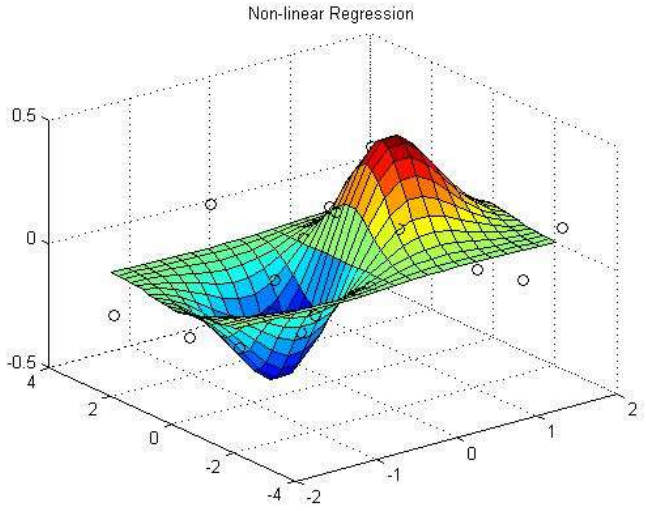


Figure 3.5: Non-linear Regression

The average rank of the first two features with respect to the number of sample size n is plotted in Figure 3.6.

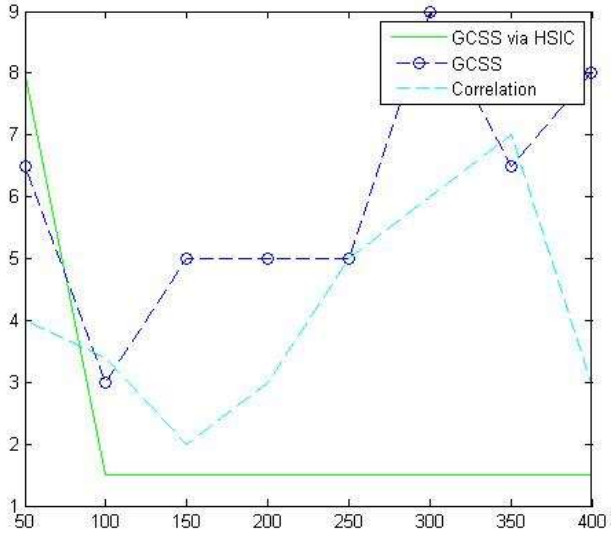


Figure 3.6: Non-linear Regression

GCSS via HSIC outputs accurate results when data size is above 100. GCSS and Pearson Correlation failed to rank the two features on top but the results are quite close.

In summary, GCSS via HSIC works well if we have previous knowledge about the kernel. GCSS instead requires no parameter tuning and no kernel selection. In general, the result GCSS from GCSS is stable and close to the true ranking. GCSS via HSIC provides good rankings consistently given the prior knowledge or tuned kernel and kernel parameters.

3.8 Feature Selection Result for Second Cancer Estimation

We use GCSS and GCSS via HSIC for feature selection on the previously extracted features from the second cancer dataset. The results for males are consistent using both methods. The feature selection results using GCSS for Female Breast Cancer dataset and Male Thyroid dataset are privided in Table 3.1 and Table 3.2:
The most important feature is ranked as 1.

Table 3.2: Feature Selection R esult for Female Breast Cancer

Index	Feature Name	Rank
1	Weighted Average on Breast	2
2	Maximum Dose on Breast	5
3	Variance of Volume on Breast between Dose Range [1000,4000]	11
4	Total Volume on Breast	7
5	Weighted Average on Lung	9
6	Total Volume on Lung	8
7	Weighted Average on Thyroid	4
8	Total Volume on Thyroid	6
9	Age	1
10	25 Quartiles of Total Volume	10
11	75 Quartiles of Total Volume	3

Table 3.3: Feature Selection Result for Male Thyroid Cancer

Index	Feature Name	Rank
1	Weighted Average on Lung	2
2	Maximum Dose on Lung	1
3	Variance of Volume on Thyroid between Dose Range [1000,4000]	4
4	Total Volume on Thyroid	3
5	Total Volume on Lung	8
6	Age	7
7	25 Quartiles of Total Volume	6
8	75 Quartiles of Total Volume	5

Chapter 4

Classification

We use two classifiers for the experiments: logistic regression and support vector machine. As a type of probabilistic statistical classification model[9], Logistic regression is used to predict a binary response (categorical class label) based on one or more predictor variables (features). That is, it is used in estimating the parameters of a qualitative response model. The probabilities describing the possible outcomes of a single trial are modeled, as a function of the explanatory (predictor) variables, using a logistic function. In our case, a binary label(response) indicating whether the patient would develop second cancer is being predicted.

Support vector machine (SVM) is a supervised learning paradigm founded on mathematical optimization. SVM can be used on analyzing data and recognizing patterns. It was shown to be successful for both classification and regression analysis[16]. Given a set of training data points, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new test data points into one category or the other, making it a deterministic binary linear classifier. Representing each training data as a point in feature space, SVM attempts to find a hyperplane in order to maximize the margin of the separate categories. The label of the test data point is decided by which side of the hyperplane it falls on.

In second cancer prediction experiment, Top eight features for females and top four features for males are selected for prediction. The classification results are shown in Table 4.1 and 4.2.

Table 4.1: Classification Result for Female Breast Cancer

	Logistic Regression	SVM
Training Error	0.2252	0.2342
Test Error	0.25	0.2523

Table 4.2: Classification Result for Male Thyroid Cancer

	Logistic Regression	SVM
Training Error	0.1	0.1977
Test Error	0.2252	0.21

Chapter 5

Conclusion

This paper provides a full set of solutions on estimating the risks of developing second cancers using modern machine learning techniques. We focus on breast cancer risks for females and thyroid cancer for males. We theoretically and algorithmically demonstrate the algorithms we used to extract and select features from the raw records of patients' information, i.e. ages and radiation therapy history.

The feature selection algorithms have been carefully tested using artificial data and provide reasonable results. We conclude that the weighted average doses, total volumes and maximum volumes on radiation are very important features which indicate the potential of developing second cancer.

The classification algorithms we used are two popular and classical methods: logistic regression and support vector machine. We obtained around 80% accuracy on prediction.

References

[1] *Radiation Therapy for Cancer.* National Cancer Institute, 2010.

[2] M.S. Kamel A. Farahat, A. Ghodsi. *An efficient greedy method for unsupervised feature selection.* IEEE 11th International Conference on. Data Mining, 2011.

[3] A.J. Roeske A.J. Mundt. *Principles of Radiation Oncology.* Oncologic Therapies. Springer Berlin Heidelberg, 2003. 9-17.

[4] A. Smola B. Schlkopf and K.R. Mller. *Kernel Principal Components Analysis.* Artificial Neural NetworksICANN'97. Springer Berlin Heidelberg,.

[5] S. Boyd and L. Vandenberghe. *Convex Optimization.*

[6] M. W. Mahoney C. Boutsidis and P. Drineas. *Unsupervised feature selection for principal components analysis.* Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2008.

[7] T. Ng C. Shiu C. Zheng, L. Zhang and D. Huang. *Molecular Pattern Discovery Based on Penalized Matrix Decomposition.* IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 8, no. 6, pp. 1592-1603, 2011.

[8] R. Tibshirani D.M.Witten and T. Hastie. *A Penalized Matrix decomposition with applications to sparse principal components and canonical correlation analysis.* Biostatistics (2009): kxp008.

[9] S. Lemeshow D.W. Hosmer and R.X. Sturdivant. *Introduction to the logistic regression model.* John Wiley & Sons, Inc., 2000.

[10] Z. Azimifar M Jahromi E. Barshan, A. Ghodsi. *Supervised Principle Component Analysis: Visualization, Classification and Regression on subspaces and submanifolds.* Pattern Recognition 44.7: 1357-1371., 2011.

[11] D.C. Hodgson et al. *Individualized estimates of second cancer risks after contemporary radiation therapy for Hodgkin lymphoma.* Cancer 110.11 (2007): 2576-2586.

[12] I. Jolliffe. *Principal Component Analysis.* John Wiley & Sons, Ltd, 2005.

[13] A. Gretton J. Bedo K. Borgwardt L. Song, A. Smola. *Feature Selection via Dependence Maximization.* Journal of Machine Learning Research 13: 1393-1434., 2012.

[14] F. Lin and W. Cohen. *Power Iteration Clustering.* Proceedings of the 27th International Conference on Machine Learning (ICML-10). 2010.

[15] Courtesy of PMH.

[16] N. Duffy D.W. Bednarski T.S. Furey, N. Cristianini. *Support vector machine classification and validation of cancer tissue samples using microarray expression data.* Bioinformatics 16.10 (2000): 906-914.