Inequality

Let $p_1, \ldots, p_r$ be nonnegative real numbers that sum to one. Ditto for $q_1, \ldots, q_r$.

Let $\lambda_1, \ldots, \lambda_r$ be strictly positive real numbers.

Show that

$$\sum_{i=1}^{r} p_i \lambda_i \leq \sum_{i=1}^{r} q_i \lambda_i$$

if & only if

$$\sum_{i=1}^{r} q_i / \lambda_i \leq \sum_{i=1}^{r} p_i / \lambda_i.$$

Equivalently, show that

$$\left(\sum_{i=1}^{r} (p_i - q_i) \lambda_i \right) \left(\sum_{i=1}^{r} (p_i - q_i) / \lambda_i \right) \leq 0$$

Proof for $r = 2$

We can see that since $\frac{1}{1/\lambda_i} = \lambda_i$, we only need to show the equivalence in one direction.

Since $\sum_i^2 p_i = \sum_i^2 q_i = 1$, if we fix p_1 and q_1, we get that $p_2 = 1 - p_1$ and $q_2 = 1 - q_2$. Let

$$\sum_{i=1}^{2} p_i \lambda_i \leq \sum_{i=1}^{2} q_i \lambda_i$$

or equivalently

$$p_1 \lambda_1 + (1 - p_1) \lambda_2 \leq q_1 \lambda_1 + (1 - q_1) \lambda_2$$

$$\implies p_1 \lambda_1 - p_1 \lambda_2 \leq q_1 \lambda_1 - q_1 \lambda_2$$
$$\implies (p_1 - q_1)(\lambda_1 - \lambda_2) \leq 0$$

Then we have four cases:

1. $p_1 = q_1$
2. $\lambda_1 = \lambda_2$
3. $p_1 < q_1$ and $\lambda_1 > \lambda_2$
4. $p_1 > q_1$ and $\lambda_1 < \lambda_2$

The first two are fairly trivial cases.

If the third is true, then $\lambda_1 > \lambda_2 \implies 1/\lambda_1 < 1/\lambda_2 \implies 1/\lambda_1 - 1/\lambda_2 > 0 \implies (p_1 - q_1)(1/\lambda_1 - 1/\lambda_2) > 0$.

If the fourth is true, then $\lambda_1 < \lambda_2 \implies 1/\lambda_1 > 1/\lambda_2 \implies 1/\lambda_1 - 1/\lambda_2 < 0 \implies (p_1 - q_1)(1/\lambda_1 - 1/\lambda_2) > 0$.

Counterexample for $r = 3$

Let

$$p = \begin{bmatrix} 1/5 \\ 2/5 \\ 2/5 \end{bmatrix}$$

$$q = \begin{bmatrix} 1/3 \\ 1/2 \\ 1/6 \end{bmatrix}$$

$$\lambda = \begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix}$$

Then $p \cdot \lambda = 3/5 + 2/5 + 4/5 = 9/5$ and $q \cdot \lambda = 1 + 1/2 + 1/3 = 11/6$. So $p \cdot \lambda < q \cdot \lambda$.

Define $\lambda^{-1} = \begin{bmatrix} 1/3 \\ 1 \\ 1/2 \end{bmatrix}$. Then $p \cdot \lambda^{-1} = 1/15 + 2/5 + 1/5 = 2/3$ and $q \cdot \lambda^{-1} = 1/9 + 1/2 + 1/12 = 25/36$. So $p \cdot \lambda^{-1} < q \cdot \lambda^{-1}$.

Exploration of conditions that make the relation hold

Lemma 1: If $\lambda_{(i)}$ is an increasing sequence of positive real numbers, then $\frac{1}{\lambda_{(i)}}$ is a decreasing sequence.

Proof: $\lambda_{(i)} \leq \lambda_{(i+1)}$. Therefore, $1/\lambda_{(i)} \geq 1/\lambda_{(i+1)}$.

Proposition 1: Given $\overrightarrow{p}$, $\overrightarrow{q}$, and $\overrightarrow{\lambda}$ defined as above, if the indicies (i) represent the increasing order of $\lambda_{(i)}$'s and $p_{(i)}$ is decreasing and $q_{(i)}$ is increasing, then $\sum_i^r p_i \lambda_i \leq \sum_i^r q_i \lambda_i$. Furthermore, since $\lambda_{(i)}$ is increasing, $1/\lambda_{(i)}$ is decreasing, so $\sum_i^r p_i/\lambda_i \geq \sum_i^r q_i/\lambda_i$.

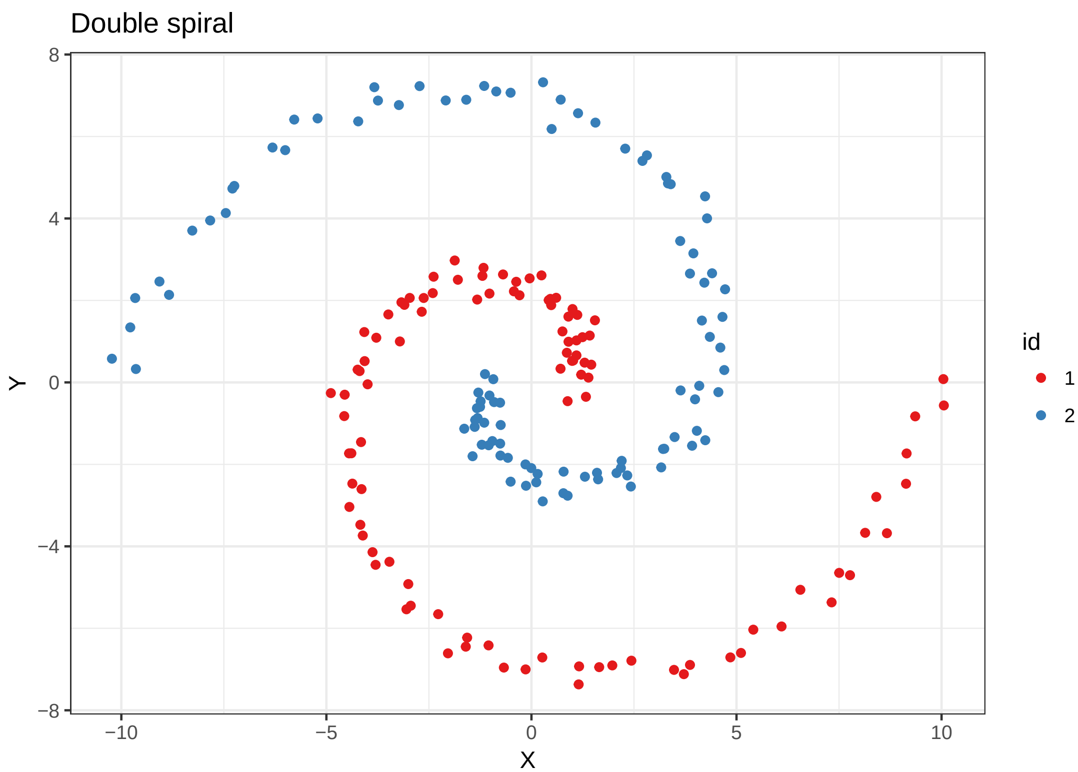
Proposition 2: Consider the OLS slope between $\overrightarrow{\lambda}$ and $\overrightarrow{p}$, $\hat{\beta}_p$. Also consider the OLS slope between $\overrightarrow{\lambda}$ and $\overrightarrow{q}$, $\hat{\beta}_q$. If $\hat{\beta}_p \leq \hat{\beta}_q$, then $\sum_i^r p_i \lambda_i \leq \sum_i^r q_i \lambda_i$.

Unfortunately, $\hat{\beta}_p \leq \hat{\beta}_q$ does not guarantee that the OLS slope between $1/\overrightarrow{\lambda}$ and $\overrightarrow{p}$ is greater than or equal to the OLS slope between $1/\overrightarrow{\lambda}$ and $\overrightarrow{q}$.

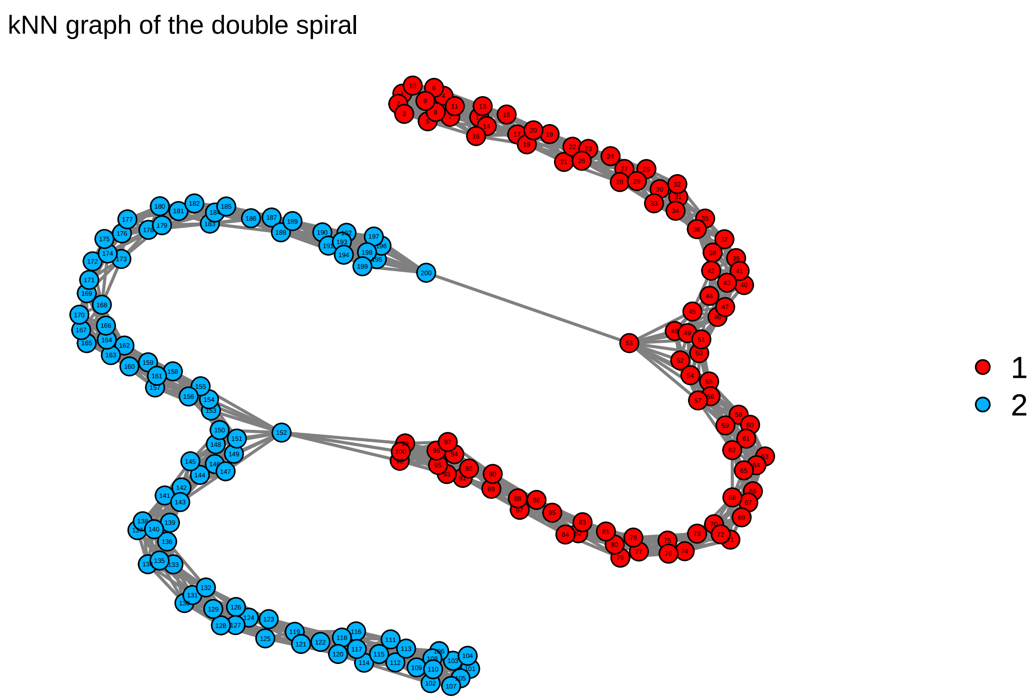
Example: Double Spiral

```r
# packages
library(ggplot2)
import::from(magrittr, `%>%`, `%<>%`)
theme_set(theme_bw())
import::from(psych, tr)
library(qgraph)

# borrow some functions from S675
source('http://pages.iu.edu/~mtrosset/Courses/675/manifold.r')

# parameters
set.seed(112358)
```

```r
s <- 2 ** 5
eps <- 2 ** -2
k <- 10  # for constructing the knn graph
K <- 2  # number of clusters
cols2 <- colorRampPalette(c('blue', 'white', 'red'))(256)
rad.max <- 10
ang.max <- 2 * pi
angles <- seq(0, ang.max, length.out = 100)
radii <- seq(1, sqrt(rad.max), length.out = 100) ** 2
iter <- 1000

# data
spiral.df <- dplyr::data_frame(X = radii * cos(angles),
                               Y = radii * sin(angles))
spiral.df <- dplyr::data_frame(X = radii * cos(angles),
                               Y = radii * sin(angles))
neg.spiral.df <- dplyr::mutate(spiral.df,
                               X = -X, Y = -Y,
                               id = '2')
spiral.df %<>%
  dplyr::mutate(id = '1') %>%
  dplyr::bind_rows(neg.spiral.df) %>%
  dplyr::mutate(X = X + rnorm(n = n(), sd = eps),
                Y = Y + rnorm(n = n(), sd = eps))

n <- nrow(spiral.df)  # number of vertices

ggplot(spiral.df) +
  geom_point(aes(x = X, y = Y, colour = id)) +
  coord_fixed() +
  scale_colour_brewer(palette = 'Set1') +
  labs(title = 'Double spiral')
```




```r
# construct W
W <- spiral.df %>%
  dplyr::select(X, Y) %>%
  as.matrix() %>%
  mds.edm1() %>%
  graph.knn(k) %>%
  graph.adj()

# viz
qgraph(W, groups = spiral.df$id,
       title = 'kNN graph of the double spiral',
       layout = 'spring')
```




```
# construct L
L <- graph.laplacian(W)

# pseudoinverse
L.svd <- svd(L)
L.dagger <- L.svd$v %*% diag(c(1 / L.svd$d[seq(n - 1)], 0)) %*% t(L.svd$u)
```

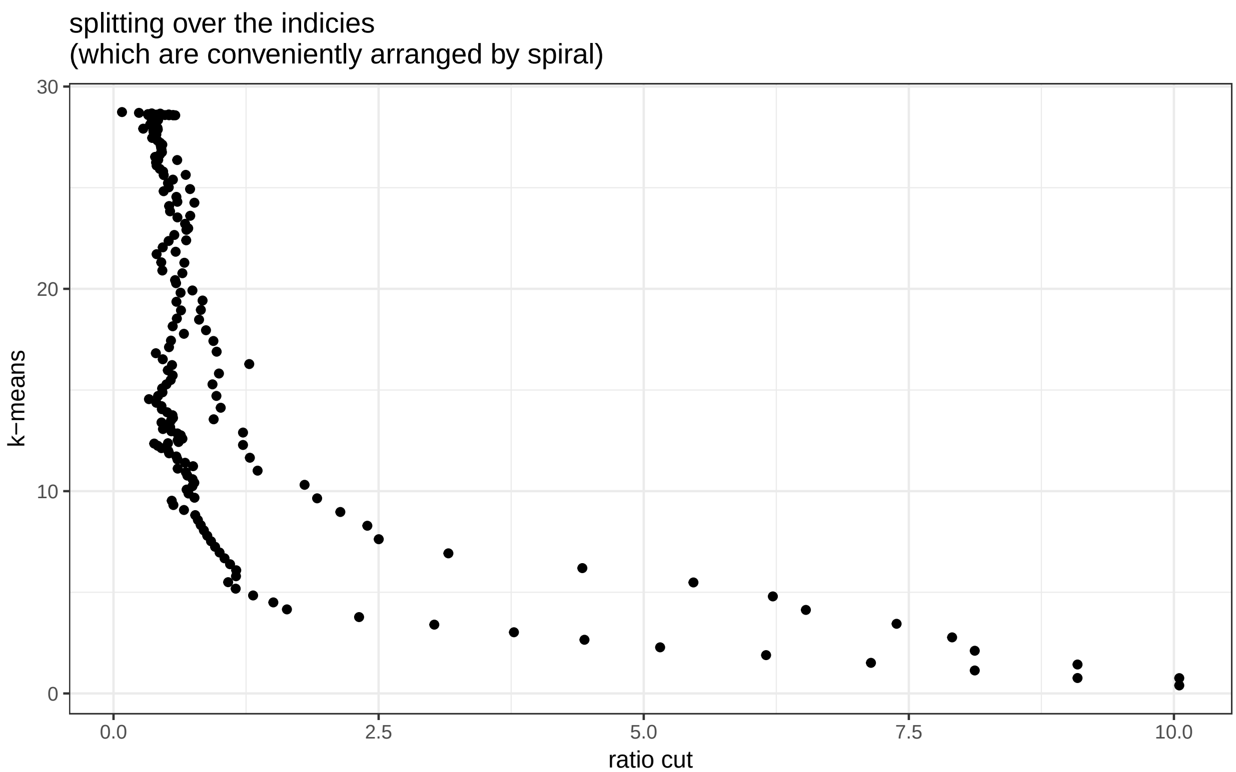
Ratio cut: minimize $\mathrm{Tr}(N^{-1/2} H L H^T N^{-1/2})$
k-means: maximize $\mathrm{Tr}(N^{-1/2} H L^{\dagger} H^T N^{-1/2})$

… where H is a $k \times n$ matrix of cluster assignments and N is a $k \times k$ diagonal matrix.

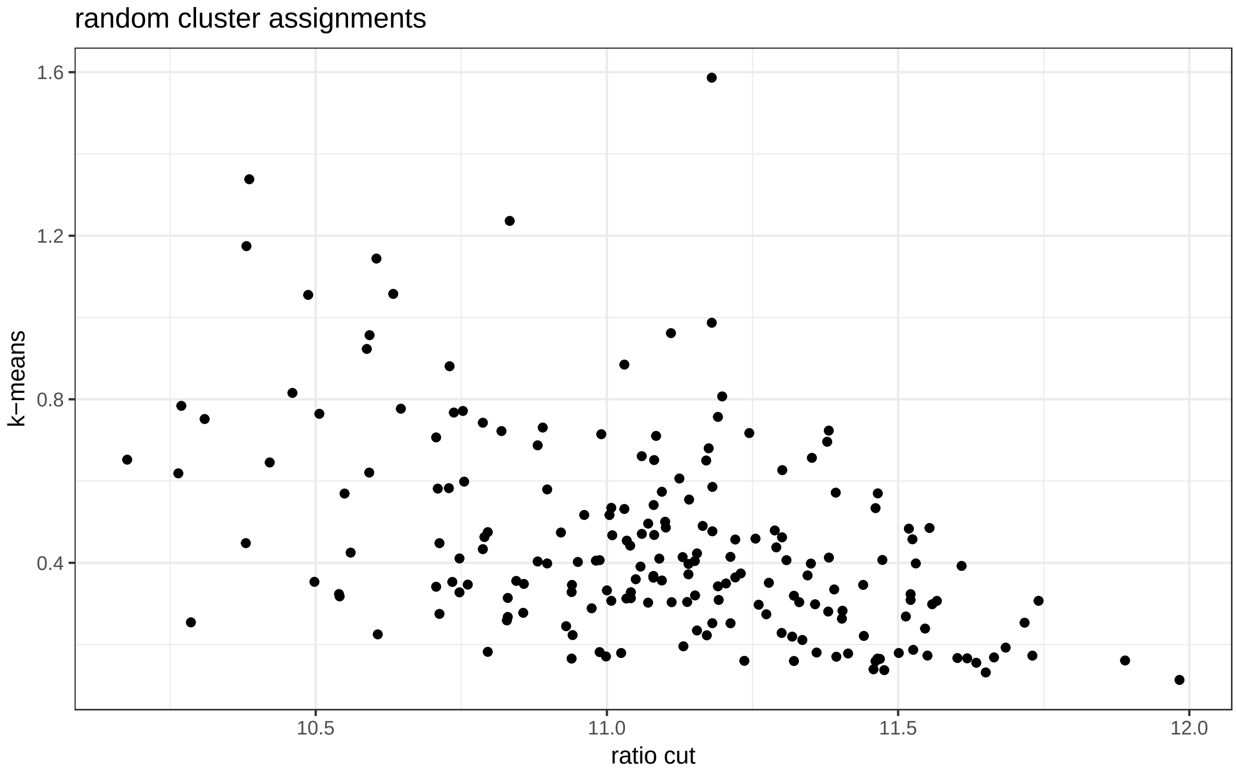
First, we can try random cluster assignments:

```
#' @title construct H function
#' @description H is constructed based on a vector that designates the clusters
#' @param clustering (numeric) A vector of cluster assignments
#' @return (matrix) H based on the cluster assignments
construct.H <- function(clustering) {
  # this function is limited to nonempty clusters
  # e.g., if there are 3 clusters, they must be assigned as 1, 2, 3
  clusters <- unique(clustering)
  if (length(clusters) != max(clustering)) {
    stop(simpleError('there are empty clusters'))
  }
  if (min(clustering) < 1) {
    stop(simpleError('cluster indexing starts at 1'))
  }

  # construct H
  H <- sapply(clustering, function(i) {
```

```r
    h <- rep(0, length(clusters))
    h[i] <- 1
    return(h)
  })

  return(H)
}

#' @title construct N
#' @param clustering (numeric) A vector of cluster assignments
#' @return (matrix) N based on cluster sizes
construct.N.sqrt <- function(clustering) {
  # this function is limited to nonempty clusters
  # e.g., if there are 3 clusters, they must be assigned as 1, 2, 3
  clusters <- unique(clustering)
  if (length(clusters) != max(clustering)) {
    stop(simpleError('there are empty clusters'))
  }
  if (min(clustering) < 1) {
    stop(simpleError('cluster indexing starts at 1'))
  }

  sapply(unique(clustering), function(i) {
    1 / sqrt(sum(clustering == i))
  }) %>%
    diag()
}

compare.ordered.df <- lapply(seq(n - 1), function(i) {
  # clust <- c(1, 2, sample(seq(2), n - 2, replace = TRUE))
  clust <- c(rep(1, i), rep(2, n - i))

  H <- construct.H(clust)
  N.sqrt <- construct.N.sqrt(clust)

  ratio.cut <- tr(N.sqrt %*% H %*% L %*% t(H) %*% N.sqrt)
  k.means <- tr(N.sqrt %*% H %*% L.dagger %*% t(H) %*% N.sqrt)

  dplyr::data_frame(ratio.cut = ratio.cut,
                    k.means = k.means)
}) %>%
  dplyr::bind_rows()

ggplot(compare.ordered.df) +
  geom_point(aes(x = ratio.cut, y = k.means)) +
  labs(x = 'ratio cut', y = 'k-means',
       title = paste('splitting over the indicies',
                     '(which are conveniently arranged by spiral)',
                     sep = '\n'))
```




```r
compare.random.df <- lapply(seq(n - 1), function(i) {
  clust <- c(1, 2, sample(seq(2), n - 2, replace = TRUE))

  H <- construct.H(clust)
  N.sqrt <- construct.N.sqrt(clust)

  ratio.cut <- tr(N.sqrt %*% H %*% L %*% t(H) %*% N.sqrt)
  k.means <- tr(N.sqrt %*% H %*% L.dagger %*% t(H) %*% N.sqrt)

  dplyr::data_frame(ratio.cut = ratio.cut,
                    k.means = k.means)
}) %>%
  dplyr::bind_rows()

ggplot(compare.random.df) +
  geom_point(aes(x = ratio.cut, y = k.means)) +
  labs(x = 'ratio cut', y = 'k-means',
       title = 'random cluster assignments')
```


random cluster assignments
1.6
1.2
0.8
0.4
k−means
10.5
11.0
11.5
12.0
ratio cut