```r
sum.I <- function(yy, FUN, Yi, Vi = NULL) {
  if (FUN == "<" | FUN == ">=") {
    yy <- -yy
    Yi <- -Yi
  }
  pos <- rank(c(yy, Yi), ties.method = 'f')[1:length(yy)] - rank(yy, ties.method = 'f')
  if (substring(FUN, 2, 2) == "=") pos <- length(Yi) - pos
  if (!is.null(Vi)) {
    if (substring(FUN, 2, 2) == "=") tmpind <- order(-Yi) else tmpind <- order(Yi)
    Vi <- apply(as.matrix(Vi)[tmpind, , drop = F], 2, cumsum)
    return(rbind(0, Vi)[pos + 1, ])
  } else return(pos)
}
```

```r
ccl.fit <- function(delta, ctime, predictors, n.ptb, seed = 1) {
  ccl_fit <- function(delta, ctime, predictors, weights = NULL) {
    fit <- suppressWarnings(
      glm(delta ~ log(ctime) + as.matrix(predictors),
          weights = weights, family = binomial))
    tmp <- fit$coef[-(1:2)]
    tmp / sqrt(sum(tmp^2))
  }
  n <- nrow(delta)
  K <- ncol(delta)
  p <- ncol(predictors)
  L <- vector("list", 5)
  for (k in 1:K) {
    L[[k]]$est <- ccl_fit(delta[, k], ctime[, k], predictors)
    L[[k]]$est.ptb <- matrix(NA, p, n.ptb)
    set.seed(seed)
    W <- matrix(NA, n, n.ptb)
    for (b in 1:n.ptb) {
      W[, b] <- tabulate(sample(1:n, n, replace = TRUE), nbins = n)
      L[[k]]$est.ptb[, b] <- ccl_fit(delta[, k], ctime[, k], predictors, W[, b])
    }
  }
  # optimal combination
  I <- matrix(1, K, 1)
  opt <- rep(NA, p)
  opt.ptb <- matrix(NA, p, n.ptb)
  for (i in 1:p) {
    v <- rep(NA, K)
    M <- matrix(NA, n.ptb, K)
    for (k in 1:K) {
      v[k] <- L[[k]]$est[i]
      M[, k] <- L[[k]]$est.ptb[i, ]
    }
    S <- cov(M)
    S.inv <- solve(S)
    w <- (S.inv %*% I) / c(t(I) %*% S.inv %*% I)
    w1 <- matrix(w, ncol = 1)
    opt[i] <- v %*% w1
    opt.ptb[i, ] <- M %*% w1
  }
  est <- opt / sqrt(sum(opt^2))
  est.ptb <- apply(opt.ptb, 2, function(x) x / sqrt(sum(x^2)))
  list(K = K,
       est = est,
       est.se = apply(est.ptb, 1, sd),
       est.ptb = est.ptb,
       W = W)
}
```

```
ccl.roc <- function(data, fit, t0, h) {
  ccl_roc <- function(data, est, t0, h, weights = NULL) {
    if (is.null(weights)) weights <- rep(1, nrow(data))
    k.wts <- dnorm((data$ctime - t0) / h) / h
    si.tilde <- as.matrix(data[, -(1:2)]) %*% est
    tpr.wts <- data$delta * k.wts
    fpr.wts <- (1 - data$delta) * k.wts
    tpr.tilde <- sum.I(si.tilde, "<=", si.tilde, tpr.wts * weights) /
      sum.I(-100, "<=", si.tilde, tpr.wts * weights)
    fpr.tilde <- sum.I(si.tilde, "<=", si.tilde, fpr.wts * weights) /
      sum.I(-100, "<=", si.tilde, fpr.wts * weights)
    tpr.prob <- sum.I(-100, "<=", si.tilde, tpr.wts * weights) / sum(k.wts * weights)
    fpr.prob <- sum.I(-100, "<=", si.tilde, fpr.wts * weights) / sum(k.wts * weights)
    ppv.tilde <- (tpr.tilde * tpr.prob) /
      (tpr.tilde * tpr.prob + fpr.tilde * fpr.prob)
    npv.tilde <- ((1 - fpr.tilde) * fpr.prob) /
      ((1 - tpr.tilde) * tpr.prob + (1 - fpr.tilde) * fpr.prob)
    tpr.05.tilde <- approx(x = fpr.tilde, y = tpr.tilde, xout = .05)$y
    tpr.10.tilde <- approx(x = fpr.tilde, y = tpr.tilde, xout = .10)$y
    tpr.tilde.sort <- tpr.tilde[order(fpr.tilde, tpr.tilde)]
    fpr.tilde.sort <- fpr.tilde[order(fpr.tilde, tpr.tilde)]
    fpr.tilde.diff <- diff(fpr.tilde.sort)
    auc.tilde.lower <- sum(fpr.tilde.diff * tpr.tilde.sort[1:(nrow(data) - 1)])
    auc.tilde.upper <- sum(fpr.tilde.diff * tpr.tilde.sort[2:nrow(data)])
    c("auc.tilde.lower" = auc.tilde.lower,
      "auc.tilde.upper" = auc.tilde.upper,
      "tpr.05.tilde" = tpr.05.tilde,
      "tpr.10.tilde" = tpr.10.tilde)
  }
  n.ptb <- ncol(fit$W)
  roc <- ccl_roc(data, fit$est, t0, h)
  roc.ptb <- matrix(NA, length(roc), n.ptb)
  for (b in 1:n.ptb) {
    weights <- rep(fit$W[, b], fit$K)
    roc.ptb[, b] <- ccl_roc(data, fit$est.ptb[, b], t0, h, weights)
  }
  list(roc = roc,
       roc.se = apply(roc.ptb, 1, sd),
       roc.ptb = roc.ptb)
}
```

```
simdata <- readRDS("simdata.rds")
delta <- as.matrix(simdata[, 1:5])
ctime <- as.matrix(simdata[, 5 + 1:5])
predictors <- as.matrix(simdata[, 10 + 1:3])
```

```r
fit <- ccl.fit(delta, ctime, predictors, n.ptb = 500, seed = 1)
data.frame(est = fit[[2]], est.se = fit[[3]])
```

```
##         est     est.se
## 1 0.6418898 0.04868211
## 2 0.5514986 0.05546564
## 3 0.5327539 0.05103419
```

```r
data <- NULL
for (k in 1:ncol(delta)) {
  DF <- data.frame(delta = delta[, k], ctime = ctime[, k], predictors)
  data <- rbind(data, DF)
}
n <- nrow(simdata)
t0 <- median(data$ctime)
h <- sd(data$ctime) / n^0.3
ans <- ccl.roc(data, fit, t0, h)
data.frame(roc = ans[[1]], roc.se = ans[[2]])
```

```
##                       roc     roc.se
## auc.tilde.lower 0.6049827 0.02274278
## auc.tilde.upper 0.6049842 0.02274271
## tpr.05.tilde    0.1307085 0.02560484
## tpr.10.tilde    0.1799483 0.02859876
```

```r
proc.time()
```

```
##    user  system elapsed
##  15.633   1.489  17.153
```