# Estimating the Total Variation Distance using Two Samples

Tom Dietterich, October 30, 2014, version 3

**Objective:** Let and be two probability distributions over Given a set of iid samples and a set of iid samples generated from unknown probability distributions and respectively, estimate the distance between the distributions and ,

**Solution:** For a finite alphabet, eq. (1) is equivalent to

which is half the L1 distance between the corresponding discrete probability mass functions.

Suppose we define a partition of into sets . Let denote the number of points from that fall in cell . Then we can approximate by

The below algorithm does exactly this by growing a kd-tree on the given data samples X and Y. At each node, the kd-tree splits on the median and stops when further splitting create leaves that are too small.

BuildTree(Data, minLeaf, NA, NB, NDim)

% the data. Data.A data set A; Data.B is data set B.

% NA = number of points in data set A

% NB = number of points in data set B

% NDim = number of dimensions

if (Data.A.numPoints() < 2\*minLeaf || Data.B.numPoints() < 2\*minLeaf) {

% not enough points to split, so compute the variation distance at this leaf

return 0.5\*(abs(Data.A.numPoints() / NA – Data.B.numPoints()/NB))

}

else {

% select a dimension at random

d = floor(random(NDim))

% Compute the median of the union of the two data sets along this dimension

med = median(SetUnion(Data.A[d], Data.B[d])) % [] returns only d’th dimension

% split the data according to the median

LeftData.A = all elements of Data.A <= med

LeftData.B = all elements of Data.B <= med

RightData.A = all elements of Data.A > med

RightData.B = all elements of Data.B > med

% combine the results of the left and right recursive calls

return BuildTree(LeftData, minLeaf, NA, NB, NDim) +

BuildTree(RightData, minLeaf, NA, NB, NDim)

}

We can call this function times and take the average as an approximate measure of total variation