TOPICS IN TWO-SAMPLE TESTING

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF STATISTICS
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Nelson C. Ray
2013

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Susan P. Holmes) Principal Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Persi W. Diaconis)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Bradley Efron)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Jerome H. Friedman)

Approved for the University Committee on Graduate Studies.

Contents

Chapter 1

Friedman's Test

In this chapter we describe Friedman's approach to the two-sample problem, give examples using a kernel support vector machine (KSVM), and explain the connection between the KSVMs and the theory developed in chapter **??**.

1.1 Motivation

The two-sample problem addresses the issue of comparing samples from two possibly different probability distributions. They range from simple parametric, location alternative tests on univariate data such as the t-test to more general non-parametric, asymptotically consistent tests, which have power against all alternatives. Many options exist for vectorial data, and kernels provide an enticing avenue for extensions to more general data types.

The two-sample problem is also widely prevalent: ensuring cross-platform compability of microarray data allows for the merging samples to achieve larger sample sizes. Biologists would like to know whether gene expression levels on a set of genes differ between cancer and control groups. Further uses for two-sample testing include authorship validation: Given two sets of documents, is the hypothesis of a single author consistent with the data?

	parametric	non-parametric
univariate	t-test	permutation t-test; Kolmogorov-Smirnov test; Wilcoxon rank-sum test
multivariate	Hotelling's T^2 test	Friedman-Rafsky test
non-vectorial	Maximum Mean Discrepancy (asymptotic)	Friedman's test (KSVM); MMD (distribution-free)
heterogeneous		Friedman's test (MKL)

Table 1.1: Two-sample tests.

1.2 Two-Sample Tests

The two-sample problem is generally posed in the following fashion: $\{\mathbf{x}_i\}_1^n$ are drawn from $p(\mathbf{x})$ and $\{\mathbf{y}_i\}_1^m$ are drawn from $q(\mathbf{y})$, where $\mathbf{x}_i, \mathbf{y}_i \in \mathbb{R}^p$. The goal is to test $H_0 : p(\mathbf{x}) = q(\mathbf{y})$ against $H_A : p(\mathbf{x}) \neq q(\mathbf{y})$. An ideal test should have power against all alternatives. That is, as $n, m \to \infty$, the test will always reject when $p \neq q$ for any non-zero significance level α.

There are many two-sample tests in the literature, as Table 1.1 illustrates.

1.3 The Friedman Two-Sample Test

Friedman proposed the following approach to the two-sample problem [5]:

For $\{\mathbf{x}_i\}_1^N$ drawn from $p(\mathbf{x})$ and $\{\mathbf{z}_i\}_1^M$ drawn from $q(\mathbf{x})$, we would like to test $\mathcal{H}_A$: $p \neq q$ against $\mathcal{H}_0$: $p = q$.

1. Pool the two samples $\{\mathbf{u}_i\}_1^{N+M} = \{\mathbf{x}_i\}_1^N \cup \{\mathbf{z}_i\}_1^M$ to create a predictor variable training set.

2. Assign a response value $y_i = 1$ to the observations from the first sample ($1 \leq i \leq N$) and $y_i = -1$ to the observations from the second sample ($N{+}1 \leq i \leq N{+}M$).

3. Apply a binary classification learning machine to the training data to produce a scoring function $f(\mathbf{u})$ to score each of the observations $\{s_i = f(\mathbf{u}_i)\}_1^{N+M}$.

4. Calculate a univariate two-sample test statistic $\hat{t} = T(\{s_i\}_1^N, \{s_i\}_{N+1}^{N+M})$.

5. Determine the permutation null distribution of the above statistic to yield a p-value.

6. The test rejects $\mathcal{H}_0$ at significance level α if $p < \alpha$.

Note that in step 3, for a given learning machine, there can still be some choice in the scoring function $f(\mathbf{u})$.

The Friedman Test (FT) is a simple, elegant idea that leverages the many advancements made over the past several decades in the fields of prediction and classification and applies them to the problem of two-sample testing. In short, as long as there exists a learning machine for the problem at hand, the Friedman Test provides a recipe for turning that learning machine into a two-sample test. This immediately yields two-sample tests for many kinds of data, including all types for which kernels have been defined. But there still remains some choice in the scoring function $F(\mathbf{u})$. It must be flexible enough to discriminate between the potential distributional differences of the problem at hand. The operating characteristics of the new two-sample test is *solely* a function of the paired learning algorithm.

By virtue of its permutation construction, the test has level α—the probability that we reject the null hypothesis given that the null hypothesis is true, also known as type I error. Given a threshold α, we wish to minimize the type II error, accepting the null hypothesis given that the alternative hypothesis is true. Equivalently, we wish to maximize the power, one minus the type II error [14]. The downside of the permutation design is, of course, that any computational cost is naïvely multiplied by the number of permutations. However, there are many situations for which the cost is sublinear in the number of permutations. For instance, caching the computation of the kernel matrix yields substantial savings when re-using it for permutation based inference. This is especially true when computation of the kernel matrix is expensive relative to finding the SVM parameters via quadratic programming.

The exact randomization distribution will be a complicated, discrete distribution parametrized by the observed data. If, however, we can approximate this distribution

with a simpler one and derive error bounds on the difference between the two distributions in some probability metric, then we can use the target distribution as a basis for inference. We will gain in computational efficiency by only having to compute the test statistic once.

1.4 Kernel Methods

There exist many two-sample tests for vectorial data $\mathbf{x}_i \in \mathbb{R}^p$. Increasingly, data collected for many applications is heterogeneous in nature and include non-vectorial components such as text, audio, or graph structures for which the mathematical and geometric operations required of many learning algorithms are not defined. Kernel methods allow us to identify a mapping of the data from a general set into a Hilbert space in which we can apply certain classes of algorithms. There is much literature on kernel methods, but one particularly comprehensive treatment is the monograph by Schölkopf and Smola [22].

Given n observed datapoints in some general set, $x_1, \dots, x_n \in \mathcal{X}$, kernelized learning algorithms depend only on the pairwise "similarities" between any two observations by way of the kernel function. Thus, kernel methods effectively decouple the algorithm (e.g. a support vector machine) from the representation of the data (e.g. a particular kernel).

Definition 1.1 (Positive Semidefinite Kernel)**.** *A function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a positive semidefinite kernel iff it is symmetric ($K(x, x') = K(x', x)$ for any $x, x' \in \mathcal{X}$) and positive semidefinite:*

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j K(x_i, x_j) \geq 0$$

for any $n > 0$, any choice of n objects $x_1, \dots, x_n \in \mathcal{X}$, and any $c_1, \dots, c_n \in \mathbb{R}$.

Because inner products are symmetric, positive semidefinite functions, they satisfy Definition 1.1 and are valid kernels. When $\mathcal{X} = \mathbb{R}^p$, the linear kernel is defined as

$$K(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}' \rangle$$

for $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$.

For objects in general sets $\mathcal{X}$, we can define a mapping ϕ into a Hilbert space $\mathcal{H}$ for which an inner product exists. As there are many such mappings, one challenge is to choose a one that is maximally useful in exploiting the structure of the data for the task at hand. In Chapter 2, we shall explore a technique to identify the most useful mapping given some parametrized space of mappings.

In fact, for every kernel K we can identify a feature mapping into a Hilbert space, where the kernel can be expressed as the inner product of the mapped features [**?**]:

Theorem 1.2. *For any kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, there exists a Hilbert space $\mathcal{H}$ and a feature mapping $\phi : \mathcal{X} \to \mathcal{H}$ such that*

$$K(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}$$

for any $x, x' \in \mathcal{X}$, where $\langle u, v \rangle_{\mathcal{H}}$ represents the inner product in $\mathcal{H}$.

In the coming sections and in Chapter 2, we shall see examples of nonvectorial spaces $\mathcal{X}$, kernels K, feature mappings ϕ, and Hilbert spaces $\mathcal{H}$.

1.5 Support Vector Machines

Support Vector Machines (SVM) [**?**] are a supervised learning technique that seeks to find a hyperplane that maximizes the margin between points of different classes. In the case that there exists no separating hyperplane, a regularization term can be added that controls the effect of misclassified points.

Although SVMs find linear decision boundaries, the algorithm depends only on inner products between its datapoints. The "kernel trick" [**?**] allows us to replace the inner products with kernel function evaluations, thus effectively identifying a linear decision boundary in the Hilbert space $\mathcal{H}$ identified by the feature mapping $\phi(x) : \mathcal{X} \to \mathcal{H}$. Although linear in the typically-higher-dimensional space $\mathcal{H}$, the decision boundary can be nonlinear in $\mathcal{X}$.

Since kernel methods divorce the representation of the data with the learning algorithm, development on both fronts can proceed independently. For instance, faster

optimization algorithms for solving the general SVM problem can proceed in parallel with the problem-specific designing of new kernels to more efficiently or effectively exploit the structure of the data.

Consider the ℓ_1-regularized (soft margin) support vector classification problem [22] in its primal form:

$$\begin{aligned} \underset{\mathbf{w}\in\mathcal{H},b\in\mathbb{R},\xi\in\mathbb{R}^m}{\text{minimize}} \quad & \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{m}\xi_i \\ \text{subject to} \quad & y_i(\mathbf{w}^t\mathbf{x}_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \qquad\qquad \text{for all } i = 1, \dots, m. \end{aligned} \tag{1.1}$$

There are three obvious possibilities for the Friedman scoring function f:

1. The predicted class label $f_1(\mathbf{x}_i) = \text{sign}(\mathbf{w}^t\mathbf{x}_i + b)$.
2. An estimate of the posterior class probability, such as by a sigmoid (Platt [18, 16]) or a logistic link function (Wahba [24, 25]) $f_2(\mathbf{x}_i) = \frac{1}{1+\exp(-(\mathbf{w}^t\mathbf{x}_i+b))}$.
3. The margin $f_3(\mathbf{x}_i) = \mathbf{w}^t\mathbf{x}_i + b$.

Because the predicted class label is simply the sign of the margin, we lose information about how likely it is for a given observation to belong to a particular class. Moreover, given constant within-class predicted class labels, the sample standard deviation is 0 and hence the t-statistic is unbounded.

Using, for instance, a logistic link function, f_2 has the interpretability of being a posterior class probability (if you believe in the probability model) and yields no information loss (since it is simply an invertible function of the margin). However, it is typically not a linear function of the margin.

The margin f_3 has the advantage of being an affine function of the data. In one dimension, $f_3(x_i) = wx_i + b$, and since the t-statistic is invariant (up to sign) to affine transformations of the data, we can see that using the margin generalizes the permutation t-test in some sense.

In the univariate setting, whether or not the t-statistic computed on the scores $\{f_3(x_i)\}$ agrees with that on the raw data $\{x_i\}$ depends on $\text{sign}(w)$. Due to symmetry,

in the permutation null distribution, w is negative with probability .5 and positive with probability .5. Thus, the permutation null in both settings appears to be t (and hence asymptotically normal).

1.5.1 Kernelized Form

It is advantageous to treat the dual [**?**] of Problem (1.1). Because of strong duality, the primal and dual solutions are equivalent. Although both optimization problems are quadratic programs, there exist fast algorithms such as the sequential minimal optimization algorithm [**?**] that exploit the special structure of the dual problem.

In addition, the dual problem is expressed only in terms of inner products, $\langle \mathbf{x_i}, \mathbf{x_j} \rangle$. The kernel trick [**?**] amounts to replacing these inner products with kernel function evaluations, $K(\mathbf{x_i}, \mathbf{x_j})$. The dual optimization problem is given by

$$\begin{aligned} \underset{\alpha \in \mathbb{R}^m}{\text{minimize}} \quad & \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j K(\mathbf{x_i}, \mathbf{x_j}) \\ \text{subject to} \quad & 0 \leq \alpha_i \leq C \quad \text{for all } i = 1, \dots, m \\ \text{and} \quad & \sum_{i=1}^{m} \alpha_i y_i = 0. \end{aligned} \tag{1.2}$$

The Karush–Kuhn–Tucker (KKT) conditions for optimality imply that

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i.$$

Therefore,

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b = \sum_{i=1}^{m} \alpha_i k(\mathbf{x}, \mathbf{x}_i) + b.$$

In fact, there is another view of the SVM problem in the framework of regularized empirical risk minimization that will be useful for Chapter 2. Define the hinge loss function

$$L(y, f(x)) := (1 - yf(x))_+ := \max(0, 1 - yf(x)).$$

Then the following optimization problem is equivalent to Problem (1.1):

$$\min_{b\in\mathbb{R},\mathbf{w}\in\mathcal{H}} \sum_{i=1}^{m}(1 - y_i(\mathbf{w}^T\mathbf{x}_i + b))_+ + \frac{1}{2C}\|\mathbf{w}\|_2^2. \tag{1.3}$$

It turns out that regularized risk minimization problems admit particularly elegant solutions, a result owing to Kimeldorf and Wahba [**?**]. We present a slightly generalized representer theorem from [22]:

Theorem 1.3 (Representer Theorem)**.** *Let $\Omega : [0,\infty] \to \mathbb{R}$ be a strictly monotonic increasing function, $\mathcal{X}$ be a set, and $c : (\mathcal{X} \times \mathbb{R}^2)^m \to \mathbb{R} \cup \{\infty\}$ be an arbitrary loss function. Then each minimizer $f \in \mathcal{H}$ of the regularized risk*

$$c((\mathbf{x_1}, y_1, f(\mathbf{x_1})), \dots, (\mathbf{x_m}, y_m, f(\mathbf{x_m}))) + \Omega(\|f\|_{\mathcal{H}})$$

admits a representation of the form

$$f'(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i' k(\mathbf{x}_i, \mathbf{x}).$$

We apply Theorem 1.3 to Problem (1.3), noting that it holds for all fixed b, to conclude that

$$f'(\mathbf{x}) = \mathbf{w}^T\mathbf{x} = \sum_{i=1}^{m} \alpha_i' k(\mathbf{x}_i, \mathbf{x}).$$

Thus, setting $\alpha_i' = y_i\alpha_i$, we again find

$$f(\mathbf{x}) = f'(\mathbf{x}) + b = \sum_{i=1}^{m} y_i\alpha_i k(\mathbf{x}, \mathbf{x}_i) + b.$$

We list the kernelized representations of possible Friedman scoring functions:

1. The predicted class label $f_1(\mathbf{x}_i) = \text{sign}(\sum_{i=1}^{m} y_i\alpha_i k(\mathbf{x}, \mathbf{x}_i) + b)$.
2. An estimate of the posterior class probability, such as by a sigmoid (Platt [18, 16]) or a logistic link function (Wahba [24, 25]) $f_2(\mathbf{x}_i) = \frac{1}{1+\exp(-(\sum_{i=1}^{m} y_i\alpha_i k(\mathbf{x},\mathbf{x}_i)+b)}$.
3. The margin $f_3(\mathbf{x}_i) = \sum_{i=1}^{m} y_i\alpha_i k(\mathbf{x}, \mathbf{x}_i) + b$.

1.5.2 Equivalence to Permutation t-Test

We know that in the univariate setting, we have an equivalence to the permutation t-test as long as the margin is an affine function of the data. So, for what kernels k do we have

$$\sum_{i=1}^{m} y_i \alpha_i k(x, x_i) + b = cx + d? \tag{1.4}$$

A sufficient condition is for $k(x, x_i) = \langle \Phi(x), \Phi(x_i) \rangle = f(x_i)x$. The linear kernel satisfies this condition with $f(x_i) = x_i$. The RBF kernel $k(x, x_i) = \exp(-\sigma(x - x_i)^2)$ does not yield an affine function of the data:

$$\sum_{i=1}^{m} y_i \alpha_i \exp(-\sigma(x - x_i)^2) + b \tag{1.5}$$

cannot be written as $cx + d$.

We use Support Vector Machine (SVM) classification as implemented in the **ksvm** function of the **R** [21] package **kernlab** [11].

SVM regression scores also generalize the permutation t-test. Recall that SVM regression solves the following problem [22]:

$$\begin{aligned} \underset{\mathbf{w} \in \mathcal{H}, \boldsymbol{\xi}^{(*)} \in \mathbb{R}^m, b \in \mathbb{R}}{\text{minimize}} \quad & \tau(\mathbf{w}, \boldsymbol{\xi}^{(*)}) = \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{M} (\xi_i + \xi_i^*) \\ \text{subject to} \quad & f(\mathbf{x}_i) - y_i \leq \epsilon + \xi_i \\ & y_i - f(\mathbf{x}_i) \leq \epsilon + \xi_i^* \\ & \xi_i, \xi_i^* \geq 0 \qquad \text{for all } i = 1, \ldots, m. \end{aligned}$$

with solution is given by

$$f(x) = \sum_{i=1}^{m} (\alpha_i^* - \alpha_i) k(x_i, x) + b.$$

1.5.3 Tuning Parameters

The cost parameter C controls the complexity of the prediction function, and ϵ controls the leniency of the loss function. These parameters are typically chosen via

cross-validation over a grid of choices. In subsection 1.8.2, we describe a sample of string data from Twitter. In figure **??** we demonstrate the statistical power of the test for the Twitter data over a grid of SVM parameters. It is clear that these parameters play a *crucial* role in the operating characteristics of the resultant test.

We emphasize that the proper strategy is to conduct the search anew for each statistic calculation in each permutation. That is, use cross-validation to find the best performing pair (C_0, ϵ_0) in terms of the Friedman Statistic. For each permutation i, use cross-validation over the same grid to find the ith pair (C_i, ϵ_i). This ensures symmetry of protocol and enforces that the test have level α. The grid search likely maximizes the power over the set of tuning parameters: it is hoped that the search benefits the actual labeling of values by at least as much as it does permuted labels.

1.5.4 Equivalence to the Permutation t-test

Theorem 1.4. *The Friedman Test paired with support vector regression or support vector classification (using the margin as a score) with the appropriate kernel generalizes the two-sample permutation t-test. Namely, the two procedures are equivalent with univariate data and a linear kernel.*

Proof.

$$f(x) = \sum_{i=1}^{m} (\alpha_i^* - \alpha_i) k(x_i, x) + b = \sum_{i=1}^{m} (\alpha_i^* - \alpha_i) x_i x + b = wx + b$$

since we have univariate data and a linear kernel. Therefore, the SVM score is simply a linear transformation of the data. Welch's t-statistic is given by

$$T(\{x_i\}_1^N, \{z_i\}_1^M) = \frac{\bar{x} - \bar{z}}{\sqrt{\frac{s_X^2}{N} + \frac{s_z^2}{M}}}$$

where

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i \text{ and } s_X^2 = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \bar{x})^2.$$

Let $z = f(x) = wx + b$ and note that

$$\bar{z} = \frac{1}{N}\sum_{i=1}^{N} z_i = \frac{w}{N}\sum_{i=1}^{N} x_i + b = w\bar{x} + b$$

and

$$s_Z^2 = \frac{1}{N-1}\sum_{i=1}^{N}(z_i - \bar{z})^2 = \frac{1}{N-1}\sum_{i=1}^{N}(wx_i + b - w\bar{x} + b)^2 = w^2 s_X^2.$$

Therefore,

$$T(\{f(x_i)\}_1^N, \{f(z_i)\}_1^M) = \frac{w\bar{x} + b - w\bar{z} + b}{|w|\sqrt{\frac{s_X^2}{N} + \frac{s_z^2}{M}}} = \text{sign}(w)T(\{x_i\}_1^N, \{z_i\}_1^M).$$

Since we are interested in two-sided testing, we consider

$$|T(\{f(x_i)\}_1^N, \{f(z_i)\}_1^M)| = |T(\{x_i\}_1^N, \{z_i\}_1^M)|.$$

Thus, the t-statistics are identical, and since the permutation procedure is the same, the tests are equivalent. □

Despite the slight dependence, the randomization distribution of the t-statistic converges weakly to the normal distribution [13]. Anonymous and Anonymous [1] use Stein's Method of exchangeable pairs [3, 23] to prove a conservative $\mathcal{O}(N^{-1/4})$ rate of convergence in Kolmogorov-Smirnov distance between the two distributions. The problem is not as straightforward as in the i.i.d. case because the permutation structure induces a global—though mild and diminishing in sample size—negative dependence in the data. This dependence thwarts traditional Fourier-analytic techniques yet can be managed via Stein's eponymous method of proof.

1.6 Maximum Mean Discrepancy

Gretton et al. [8, 10, 9, 2] introduce a kernel based approach for the two-sample problem based on the Maximum Mean Discrepancy (MMD) statistic, an integral probability metric. MMD provides good performance in practice, strong theoretical

guarantees, and is the first two-sample test for comparing distributions over graphs.

Definition 1.5. *With $\mathfrak{F}$ a class of functions $f : \mathcal{X} \to \mathbb{R}$, p and q probability distributions, and $X \sim p$ and $Z \sim q$ random variables, the maximum mean discrepancy (MMD) and its empirical estimate are defined as*

$$MMD[\mathfrak{F}, p, q] := \sup_{f \in \mathfrak{F}} (\mathbb{E}_{x \sim p}[f(x)] - \mathbb{E}_{z \sim q}[f(z)]),$$

$$MMD[\mathfrak{F}, X, Z] := \sup_{f \in \mathfrak{F}} \left(\frac{1}{N} \sum_{i=1}^{N} f(x_i) - \frac{1}{M} \sum_{i=1}^{M} f(z_i) \right).$$

The function class $\mathfrak{F}$ is typically taken to be the unit ball in a reproducing kernel Hilbert space (RKHS), however, well-known metrics can be obtained over other function classes. Although Gretton et al. provide several distribution-free tests based on MMD theory, we instead compare the Friedman Test (FT) against the permutation-based MMD so as to compare statistic with statistic. In this way, the theory is dissociated from the comparison. We feel that this is the most fair comparison of the two tests because many of the theoretical results are inexact. We also do not have big enough sample sizes in our real datasets to ensure low error in theoretical approximations. Even if we did, the power for the tests would be very nearly one, making comparisons on non-simulated data difficult.

1.7 Null Distributions

The null distribution plays a fundamental role in frequentist statistical inference. Hotelling's T^2-statistic has null distribution that corresponds to a scaled central $F_{(p,n+m-1-p)}$ distribution, where p is the dimensionality of the data and n, m are the sample sizes of the two groups. As its name suggests, the T^2-test is a generalization of Student's t-test, and for $T \sim t(n+m-2)$, we have that $T^2 \sim F_{(1,n+m-2)}$. As a consequence of Theorem 1.4, the Friedman Statistic in the univariate data, linear kernel setting is equal to the $|T|$. In figure 1.1 we simulate 200 standard multivariate normal draws from each class with dimension $D \in \{1, 5, 10\}$. We compare the null

distributions of the T^2-statistic, KMMD, and Friedman statistics with a linear kernel and RBF kernel with width parameter 1. We draw 5,000 samples from each permutation null distribution and apply a kernel density smoother to the results. It appears that many of the null distributions are very close to Normal (or, $t(398)$, rather).

Observe the Anderson-Darling test for normality p-values for each statistic:

	D	variable	p-value
1	1	T2	NaN
2	1	sqrtT2	1.152257e-157
3	1	KMMD.l	2.659966e-156
4	1	FS.l	9.210211e-01
5	1	KMMD.rbf	2.220671e-84
6	1	FS.rbf	7.945208e-01
7	5	T2	1.750434e-156
8	5	sqrtT2	1.320034e-14
9	5	KMMD.l	4.955331e-39
10	5	FS.l	2.908307e-03
11	5	KMMD.rbf	1.475365e-15
12	5	FS.rbf	5.837144e-01
13	10	T2	5.660731e-105
14	10	sqrtT2	4.132584e-11
15	10	KMMD.l	4.087932e-38
16	10	FS.l	7.618190e-01
17	10	KMMD.rbf	3.778002e-01
18	10	FS.rbf	8.071885e-01

Now only p-values $> .001$:

	D	variable	p-value
4	1	FS.l	0.921021148
6	1	FS.rbf	0.794520826
10	5	FS.l	0.002908307
12	5	FS.rbf	0.583714415

```
16 10     FS.l 0.761819040
17 10 KMMD.rbf 0.377800181
18 10   FS.rbf 0.807188534
```

The Friedman Statistic null distributions appear to be consistent with a standard Normal distribution.

Note that in one dimension, the t-statistic is independent of the regularization parameter C. This relationship does not hold in higher dimensions:

```
laply(10^seq(-3, 3, 1), function(C) computeFS(u, km, l, C)) ## 1 dimension
> [1] 0.6438212 0.6438212 0.6438212 0.6438212 0.6438212 0.6438212 0.6438212
laply(10^seq(-3, 3, 1), function(C) computeFS(u, km, l, C)) ## 2 dimensions
> [1] 3.172028 3.172028 3.174125 3.168278 3.172856 3.169756 3.169530
laply(10^seq(-3, 3, 1), function(C) computeFS(u, km, l, C)) ## 10 dimensions
> [1] 6.830184 6.850385 6.034328 6.619036 6.619036 6.619036 6.619036
```

This suggests that C actually has an influence in higher dimensions.

The T^2 densities correspond to a parametrized family of F-distributions. It is not surprising that the MMD linear kernel null distributions shift rightward as a function of dimension: the higher dimensionality affords the function in the RKHS to better find discrepancies between the two empirical distributions. The same rationale holds true for the FS when thinking of separating hyperplanes. Interestingly, there are marked differences between the MMD and FS for the RBF kernel.

1.8 Experiments

1.8.1 Vectorial Data

We consider $\{x_i\}_{i=1}^{20} \sim \text{MVN}_d(\mathbf{0}, \mathbf{I})$ and $\{y_i\}_{i=1}^{20} \sim \text{MVN}_d(\Delta\mathbf{1}, \mathbf{I})$ where our dimensionality $d \in \{1, 5, 10, 20\}$ and mean difference $\Delta \in \{0, .5, \dots, 1.5\}$ in figure 1.2. The width parameter in the RBF kernel is fixed at 1.

For FS and MMD, we used the the RBF kernel with a width of 1. The methods perform similarly with the exception of the kernel methods using the RBF kernel.

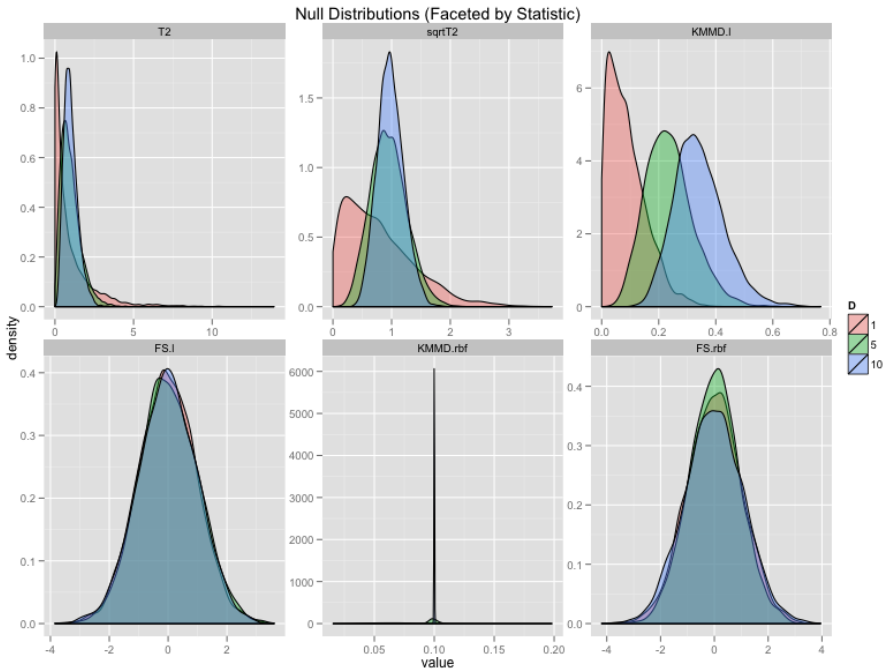


Figure 1.1: T2: Hotelling's T^2-statistic; sqrtT2: $|T|$; KMMD.l: kernel MMD with a linear kernel; FS.l: FS with a linear kernel; KMMD.rbf: kernel MMD with a radial basis function (RBF) kernel; FS.rbf: FS with RBF kernel

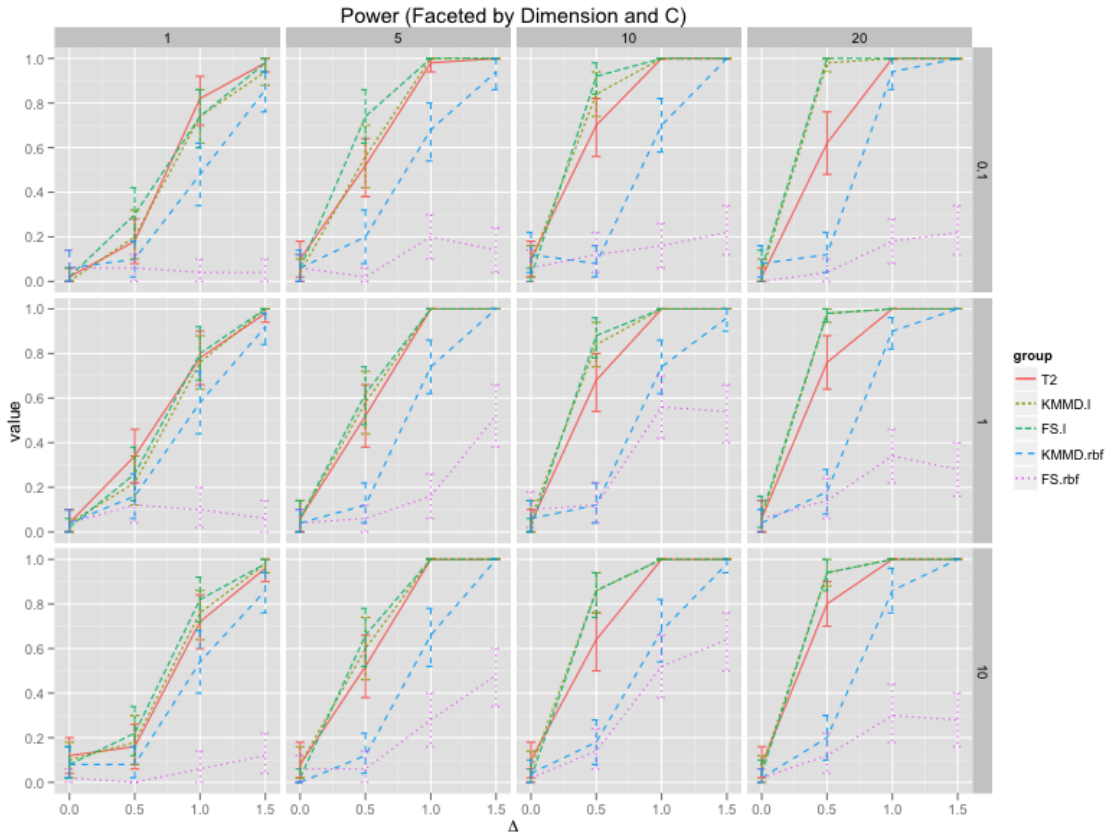


Figure 1.2: FS: Friedman statistic; KMMD: kernel Maximum Mean Discrepancy; T2: Hotelling's T^2-statistic; Error bars indicate 95% bootstrap confidence intervals. The tests perform similarly, and the kernel-based tests use a linear kernel.

This suggests that either a width of 1 is ineffective or the RBF kernel is unsuitable for these data (probably the former).

1.8.2 String Data

For a string data comparison, we consider Twitter data and look at the latest 1,000 tweets from Barack Obama (@BarackObama) and Sarah Palin (@SarahPalinUSA) obtained from the **R package twitteR [6]. We pre-process each tweet by removing all hyperlinks and anything that is neither a letter nor a space. Finally, we convert all letters to lowercase. For simplicity, we choose the k-spectrum kernel [15] with $k \in \{1, 2, 3\}$ as our kernels for both the FT and MMD. Thus, each string is mapped to a 27^k dimensional feature vector of counts of the number of k letter and space combinations. We draw samples of various sizes from both the Barack Obama tweets and Sarah Palin tweets in order to empirically determine the power, with results detailed in figure 1.3.**

The MMD test outperforms the Friedman test on this task for $k < 3$. Power increases as a function of k for both tests, and it is somewhat surprising to see the strong performance from considering only frequencies of unigrams.

1.8.3 Image Data

We consider the task of discriminating between images of roosters and pigeons from the Caltech 101 Object Categories dataset [4]. Samples of the birds are in figure 1.4. We resize images to a common resolution of 300×297 and convert to a vector of 8 bit grayscale values. To correct for global differences in illumination and ensure that only local patterns would be used for discrimination, we center and scale each vector. Power comparisons can be seen in figure 1.5.

Again, MMD performs better. However, it appears that the linear kernel performs significantly worse for the MMD than for the FS. This

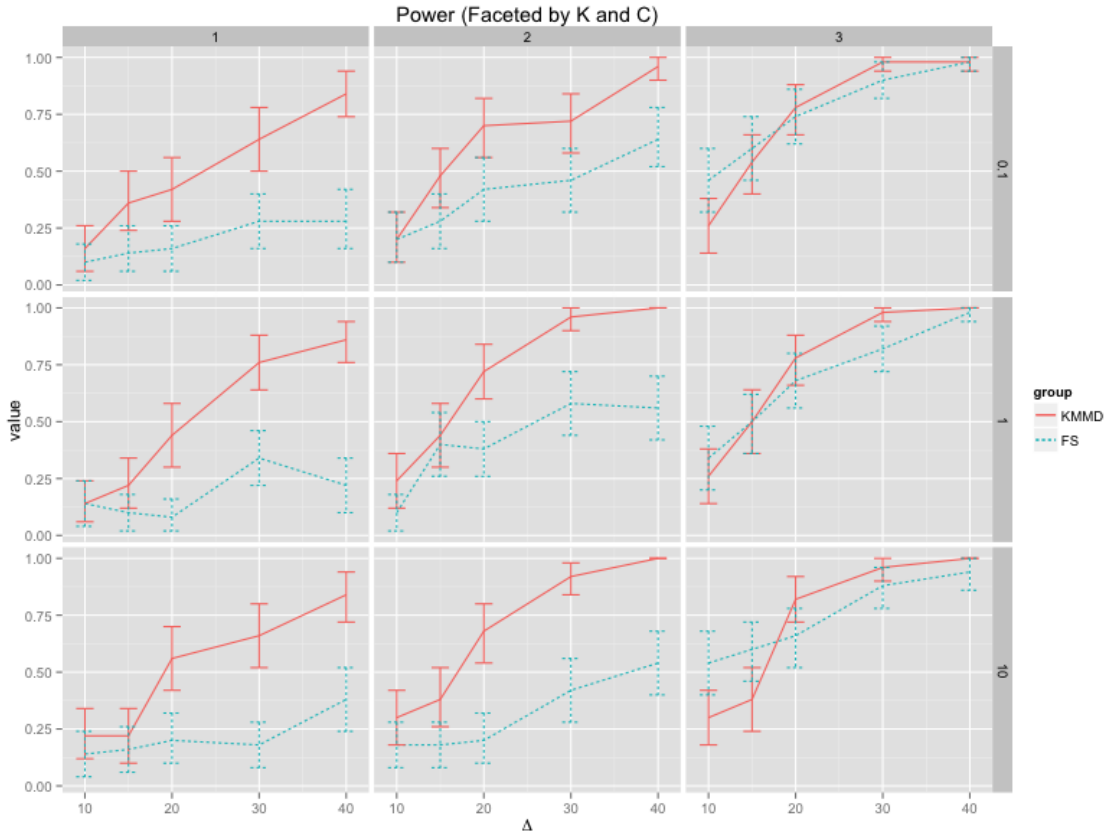


Figure 1.3: FS: Friedman statistic; KMMD: kernel Maximum Mean Discrepancy; Error bars indicate 95% bootstrap confidence intervals.

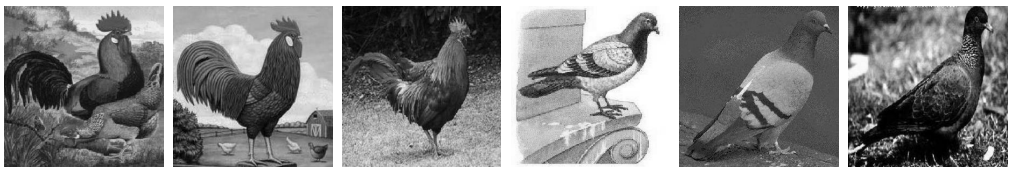


Figure 1.4: Images of roosters and pigeons for use in discrimination test.

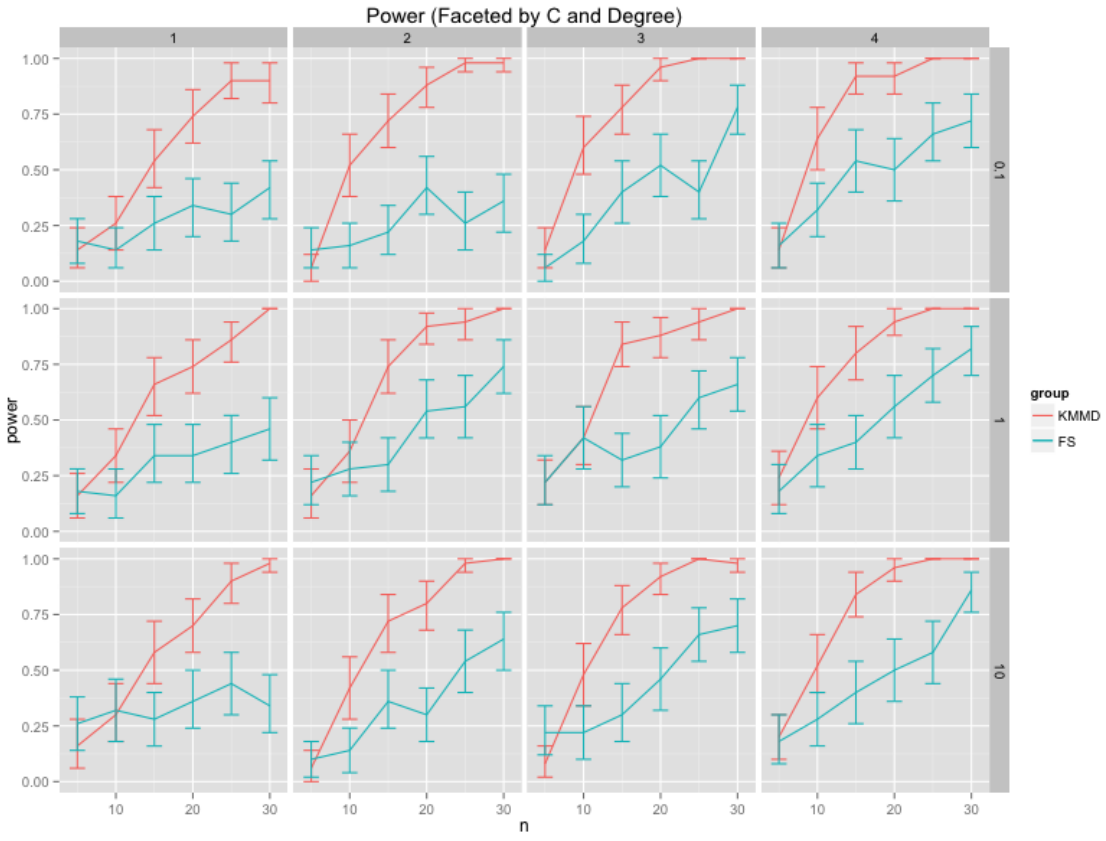


Figure 1.5: p1: linear kernel; p2: inhomogeneous degree 2 polynomial kernel; rbf: radial basis function kernel; Error bars indicate 95% bootstrap confidence intervals.

could reflect a difference in the function classes over which each technique operates.

1.9 Extensions

1.9.1 Heterogeneous Data

This procedure extends naturally to the heterogeneous data setting via multiple kernel learning (MKL) [12, 7]. Qiu et al. [20] develop MKL for support vector regression. Given j different data modalities, it suffices to match a kernel K_i to each—or perhaps more than one kernel for each data source, so as to better target specific features. The semidefinite programming approach (SDP) to MKL finds the best linear combination $K = \sum_{i=1}^{j} \mu_i K_i$ for some relevant objective function. For computational reasons, the best non-negative linear combination is frequently sought, as this yields a simpler quadratically constrained quadratic program (QCQP).

1.9.2 Missing Data

If we further consider entire missing modalities (e.g. one sample is missing some biometric reading), Poh et al. [19] develop the *neutral point substitution* technique to allow substitution of the missing modality with a new kernel that is *unbiased* with regard to the classification at hand. This allows for full use of both modalities that are present for all samples as well as those that are present only for a subset of the samples and effective utilization of all the data in the training set. Panov et al. [17] modify the NPS method to allow for missing modalities in the test set.

1.9.3 Theoretical Guarantees

Having proved a bound in the univariate data, linear kernel case by constructing an exchangeable pair, Anonymous and Anonymous [1] use simulations to suggest that the same pair is likely to yield success in more general settings: the key ***approximate regression condition*** holds more universally for multivariate data, a non-linear kernel, and a combination of the two settings. Further simulations demonstrate that the $\mathcal{O}(N^{-1/4})$ rate of convergence does not appear to be tight and a more typical $\mathcal{O}(N^{-1/2})$ is within reach.

A rate of convergence result with known constant allows for a single calculation of the Friedman statistic—rather than the N_{perm} required for randomization-based inference. Theoretical inference could be done on the limiting distribution, with error characterized by the proven bound. This large savings in computation comes only at the known cost of the limiting distribution approximation, which falls rapidly in sample size.

1.10 Discussion

We have tested a two-sample testing method of Friedman's [5] with a particular choice of learning algorithm—support vector regression. This Friedman Test can be seen as a generalization of the celebrated permutation t-test, or randomization test. Without tuning, performance is competitive in some settings with the MMD test. Simulations suggest that more powerful tests may be achieved with the added complexity of tuning—at some computational cost. Further work is required to determine a good set of heuristic choices for the SVM tuning parameters.

Modern data sources often consist of different modalities. Wireless sensor networks (including cellular phones) are deployed to collect large quantities of *diverse* data. These networks may be heterogeneous, with newer and upgraded hardware logging novel sources of data. Because

Friedman's idea leverages *any* learning algorithm, we can at present easily incorporate extensions such as both the treatment of heterogeneous data *and* an allowance for missing data modalities. Future developments in regression and classification can be incorporated to advance the state-of-the-art in two-sample testing.

Chapter 2

Multiple Kernels

In this chapter we introduce a framework for two sample testing based on heterogeneous data based on multiple kernel learning (MKL).

2.1 Introduction

Introduction to MKL, theory of MKL, and optimization problems/tuning parameters.

2.2 Simulations

2.2.1 Vectorial Data Mixture Distribution

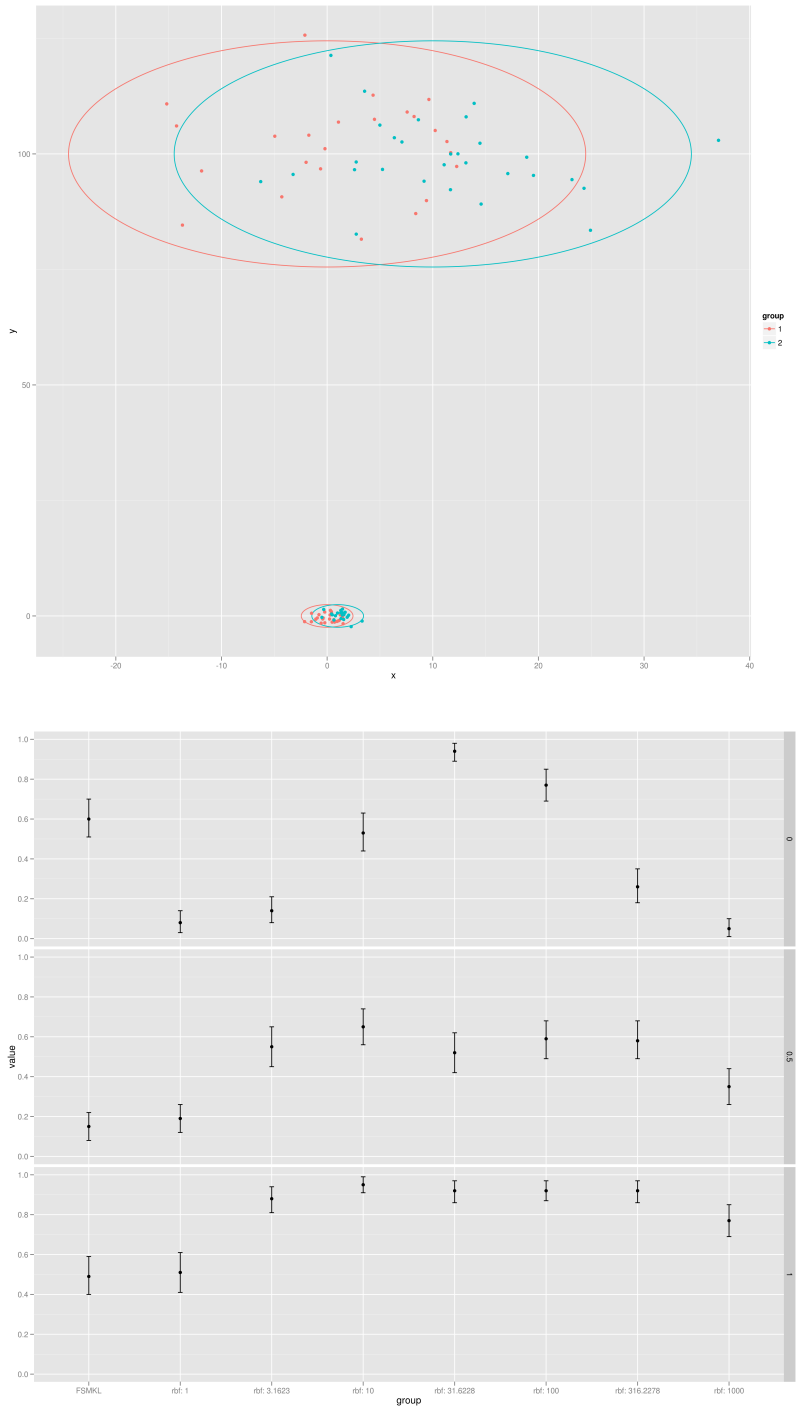
Let's look at mixtures of MVN. Let

$$\Sigma_1 = \begin{bmatrix} 1^2 & 0 \\ 0 & 1^2 \end{bmatrix}$$
$$\Sigma_2 = \begin{bmatrix} 10^2 & 0 \\ 0 & 10^2 \end{bmatrix}$$
$$\mu_1(\delta_1) = [\delta_1, 0]^T$$
$$\mu_2(\delta_2) = [\delta_2, 100]^T.$$

Let d_1 be a mixture distribution of $\mathcal{N}_2([0,0]^T, \Sigma_1)$ with probability p and $\mathcal{N}_2([0,100]^T, \Sigma_2)$ with probability $1-p$. Let d_2 be a mixture distribution of $\mathcal{N}_2([1,0]^T, \Sigma_1)$ with probability p and $\mathcal{N}_2([10,100]^T, \Sigma_2)$ with probability $1-p$. Note that $\delta_1 = 1$ and $\delta_2 = 10$ were chosen to be one standard deviation away (on the x-axis, see $(\Sigma_1)_{1,1}$ and $(\Sigma_2)_{1,1}$). In all the simulations, we draw $n = 50$ samples from each mixture distribution, d_1 and d_2. We take the mixture probability $p = .5$ unless otherwise specified.

Here is a plot of the 95% confidence ellipses of the mixture distributions:

Here we plot the average power and bootstrap 95% confidence intervals (100 simulations) for each RBF kernel individually and MKL on all of them, faceted on the mixture probability. So for $p = 0$, we have all the weight on $\mathcal{N}_2(\mu_2, \Sigma_2)$, and for $p = 1$, we have all the weight on $\mathcal{N}_2(\mu_1, \Sigma_1)$. Since the latter is on a smaller scale, we expect the smaller width RBF kernels to do better. We take $C = 1$, and the widths to be from the middle run of the last section: The smaller distribution ($p = 1$) has higher power for smaller kernels, but the MKL power is about the same as compared



with the $p = 0$ case. Both outperform the mixed setting.

Here are the null distributions of the weights and the observed weights:

2.2.2 Heterogeneous Data

I created a test heterogeneous dataset that has two components to it: DNA string and univariate. I created the DNA data via a Markov chain because I wanted the joint distribution of 2-grams to be different from the product of two 1-grams. I wanted this dependence so that I could later pick out differences between two groups with a 2-spectrum kernel instead of a 1-spectrum kernel. For the first group, I randomly picked a starting string according to the stationary distribution and then proceeded via the transition probabilities. For the second group, I had independent draws from the stationary distribution.

The univariate data is simply $\mathcal{N}(\{-\mu, \mu\}, 1)$, and there are 20 samples in each group. I fixed the kernel training parameter values and trained a convex combination of 5 kernels on the data via MKL: Gaussian RBF

kernels with parameter .1, .2, .5, and 1, and a 2-spectrum kernel (later I do want to test that the 2-spectrum is required in this case over the 1-spectrum because of the Markov chain construction of the data, but I had to add pre-processors and I'm still not too familiar with shogun yet).

I looked at the kernel weights (so no Friedman test yet) over $\mu = .5, .6, \ldots, 2.9, 3$. The weights on, say, the 2-spectrum kernel aren't monotonically decreasing because of sampling variance: I only ran each of these once. I have attached MKL1.png to show that the signal in the DNA data outweighs (in the sense of yielding a higher MKL weight) the signal in the univariate part of the data until $\mu = 1.5$ or so. At that point, the RBF1 kernel takes over.

The pictures from our meeting weren't that compelling, so I've been looking for better examples. I decided a reasonable one was the Christmas star example on page 1548 of http://eprints.pascal-network.org/archive/00002269/01/s nenburg06a.pdf

Imagine an outer star (radius 4, 5, 6, 7, 8) with an inner star (radius 4) inside. The different stars correspond to different labelings in the classification problem. I looked at 5 kernels individually (RBF with width .01, .1, 1, 10, 1000) and the MKL combination of all of them).

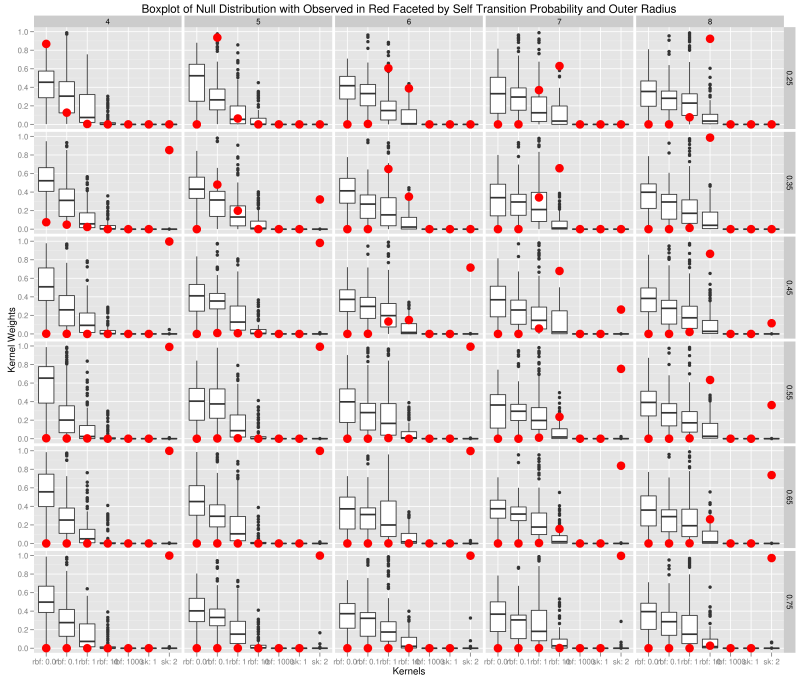
Here's a heterogeneous data example. I added string (DNA) data to the Christmas star example from last week. So each point is $(l_i, x_i, y_i, s_i) =$ (label, Christmas star x-coordinate, Christmas star y-coordinate, DNA sequence). I generated the DNA by first picking a random (Poisson) length, sampling the starting point from the stationary distribution (all 1/4) of the Markov chain, and then picking according to the transition matrix M, where $M_{i,i} = s$ (for self transition probability) and $M_{i,j} = \frac{(1-s)}{3}$ for $i \neq j$.

Here are the MKL weights:

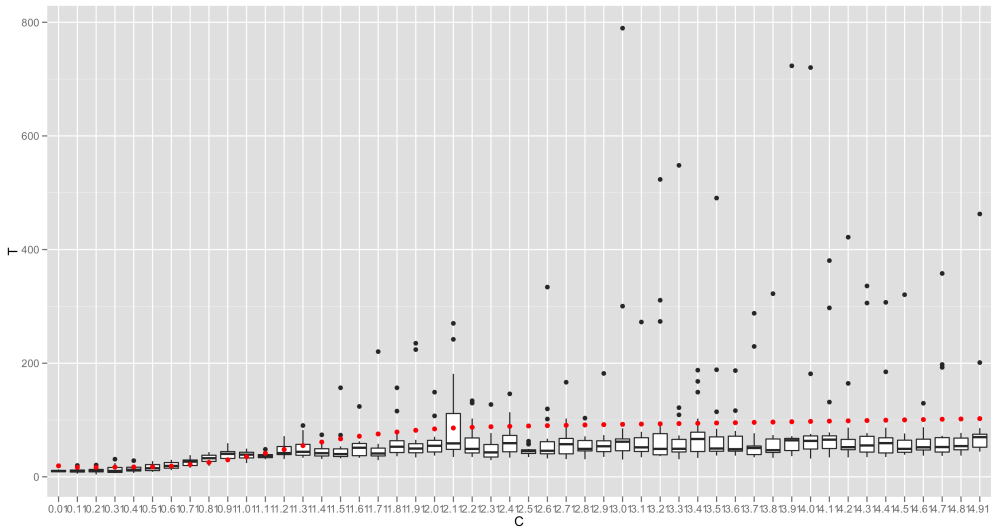
And the power:

The effect of C:

C clearly has an effect, especially if you get it very wrong (>.8). I'm a little disappointed in the performance of MKL but pleased that it does

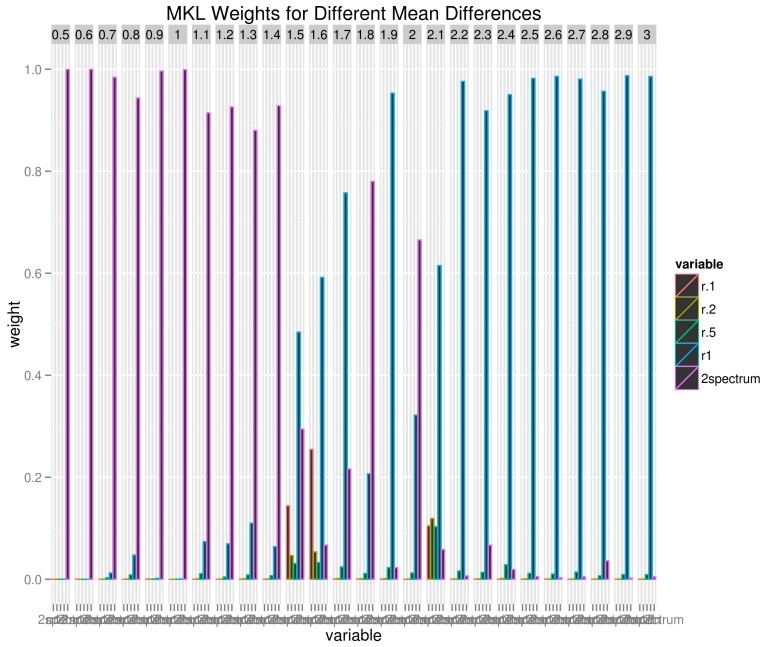


pick out the right structure in the data. I'd like to say that if you know the structure of the data a priori and use that in the kernel, you will obviously



get the best performance. It seems like you give up a lot of performance using MKL (maybe too much to justify its convenience), only doing better than the worst choices of kernels given.

MKL can pick out the structure of the data:



2.3 Wine Example

TODO

References

[1] A. Anonymous and B. Anonymous. A rate of convergence bound for the randomization t-distribution using stein's method. ***Unpublished Technical Report.***

[2] K.M. Borgwardt, A. Gretton, M.J. Rasch, H.P. Kriegel, B. Schölkopf, and A.J. Smola. Integrating structured biological data by kernel maximum mean discrepancy. ***Bioinformatics***, 22(14):e49–e57, 2006.

[3] L.H.Y. Chen, L. Goldstein, and Q.M. Shao. ***Normal Approximation by Stein's Method.*** Springer Verlag, 2010.

[4] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. ***Computer Vision and Image Understanding***, 106(1):59–70, 2007.

[5] J.H. Friedman. On Multivariate Goodness–of–Fit and Two–Sample Testing. ***Proceedings of Phystat2003, http://www.slac.stanford.edu/econf/C***, 30908, 2003.

[6] Jeff Gentry. ***twitteR: R based Twitter client***, 2011. R package version 0.99.6.

[7] M. Gönen and E. Alpaydın. Multiple kernel learning algorithms. ***Journal of Machine Learning Research***, 12:2211–2268, 2011.

[8] A. Gretton, KM Borgwardt, M. Rasch, B. Schölkopf, and AJ Smola. A kernel method for the two-sample-problem. *Advances in Neural Information Processing Systems*, 19:513–520, 2007.

[9] A. Gretton, K.M. Borgwardt, M.J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13:723–773, 2012.

[10] A. Gretton, K. Fukumizu, Z. Harchaoui, and B.K. Sriperumbudur. A fast, consistent kernel two-sample test. *Advances in Neural Information Processing Systems*, 22:673–681, 2010.

[11] Alexandros Karatzoglou, Alex Smola, Kurt Hornik, and Achim Zeileis. kernlab – an S4 package for kernel methods in R. *Journal of Statistical Software*, 11(9):1–20, 2004.

[12] G.R.G. Lanckriet, N. Cristianini, P. Bartlett, L.E. Ghaoui, and M.I. Jordan. Learning the kernel matrix with semidefinite programming. *The Journal of Machine Learning Research*, 5:27–72, 2004.

[13] E.L. Lehmann. *Elements of large-sample theory*. Springer Verlag, 1999.

[14] E.L. Lehmann and J.P. Romano. *Testing Statistical Hypotheses*. Springer Verlag, 2005.

[15] C. Leslie, E. Eskin, and W.S. Noble. The spectrum kernel: A string kernel for SVM protein classification. In *Proceedings of the Pacific Symposium on Biocomputing*, volume 7, pages 566–575. Hawaii, USA., 2002.

[16] H.T. Lin, C.J. Lin, and R.C. Weng. A note on platt's probabilistic outputs for support vector machines. *Machine learning*, 68(3):267–276, 2007.

[17] M. Panov, A. Tatarchuk, V. Mottl, and D. Windridge. A modified neutral point method for kernel-based fusion of pattern-recognition modalities with incomplete data sets. ***Multiple Classifier Systems***, pages 126–136, 2011.

[18] J. Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. ***Advances in large margin classifiers***, 10(3):61–74, 1999.

[19] N. Poh, D. Windridge, V. Mottl, A. Tatarchuk, and A. Eliseyev. Addressing missing values in kernel-based multimodal biometric fusion using neutral point substitution. ***Information Forensics and Security, IEEE Transactions on***, 5(3):461–469, 2010.

[20] S. Qiu and T. Lane. Multiple kernel learning for support vector regression. ***Computer Science Department, The University of New Mexico, Albuquerque, NM, USA, Tech. Rep***, 2005.

[21] R Development Core Team. ***R: A Language and Environment for Statistical Computing***. R Foundation for Statistical Computing, Vienna, Austria, 2011. ISBN 3-900051-07-0.

[22] B. Schölkopf and A.J. Smola. ***Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond***. the MIT Press, 2002.

[23] C. Stein. Approximate computation of expectations. ***Lecture Notes-Monograph Series***, 7, 1986.

[24] G. Wahba. Multivariate function and operator estimation, based on smoothing splines and reproducing kernels. In ***SANTA FE INSTITUTE STUDIES IN THE SCIENCES OF COMPLEXITY-PROCEEDINGS VOLUME-***, volume 12, pages 95–95. ADDISON-WESLEY PUBLISHING CO, 1992.

[25] G. Wahba et al. Support vector machines, reproducing kernel hilbert spaces and the randomized gacv. *Advances in Kernel Methods-Support Vector Learning*, 6:69–87, 1999.