The Falling Factorial Basis and Its Statistical Applications

Yu-Xiang Wang YUXIANGW@CS.CMU.EDU
Alex Smola ALEX@SMOLA.ORG
Ryan J. Tibshirani RYANTIBS@STAT.CMU.EDU
Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213

Abstract

We study a novel spline-like basis, which we name the *falling factorial basis*, bearing many similarities to the classic truncated power basis. The advantage of the falling factorial basis is that it enables rapid, linear-time computations in basis matrix multiplication and basis matrix inversion. The falling factorial functions are not actually splines, but are close enough to splines that they provably retain some of the favorable properties of the latter functions. We examine their application in two problems: trend filtering over arbitrary input points, and a higher-order variant of the two-sample Kolmogorov-Smirnov test.

1. Introduction

Splines are an old concept, and they play important roles in various subfields of mathematics and statistics; see e.g., de Boor (1978), Wahba (1990) for two classic references. In words, a spline of order k is a piecewise polynomial of degree k that is continuous and has continuous derivatives of orders $1, 2, \dots k-1$ at its knot points. In this paper, we look at a new twist on an old problem: we examine a novel set of spline-like basis functions with sound computational and statistical properties. This basis, which we call the *falling factorial basis*, is particularly attractive when assessing higher order of smoothness via the total variation operator, due to the capability for sparse decompositions. A summary of our main findings is as follows.

- The falling factorial basis and its inverse both admit a linear-time transformation, i.e., much faster decompositions than the spline basis, and even faster than, e.g., the fast Fourier transform.
- For all practical purposes, the falling factorial basis shares the statistical properties of the spline basis. We derive a sharp characterization of the discrepancy between the two bases in terms of the polynomial degree and the distance between sampling points.
- We simplify and extend known convergence results on trend filtering, a nonparametric regression technique that implicitly employs the falling factorial basis.
- We also extend the Kolmogorov-Smirnov two-sample test to account for higher order differences, and utilize the falling factorial basis for rapid computations. We provide no theory but demonstrate excellent empirical results, improving on, e.g., the maximum mean discrepancy (Gretton et al., 2012) and Anderson-Darling (Anderson & Darling, 1954) tests.

In short, the falling factorial function class offers an exciting prospect for univariate function regularization.

Now let us review some basics. Recall that the set of kth order splines with knots over a fixed set of n points forms an $(n+k+1)$-dimensional subspace of functions. Here and throughout, we assume that we are given ordered input points $x_1 < x_2 < \ldots < x_n$ and a polynomial order $k \geq 0$, and we define a set of knots $T = \{t_1, \dots t_{n-k-1}\}$ by excluding some of the input points at the left and right boundaries, in particular,

$$T = \begin{cases} \{x_{k/2+2}, \dots x_{n-k/2}\} & \text{if } k \text{ is even,} \\ \{x_{(k+1)/2+1}, \dots x_{n-(k+1)/2}\} & \text{if } k \text{ is odd.} \end{cases} \tag{1}$$

The set of kth order splines with knots in T hence forms an n-dimensional subspace of functions. The canonical parametrization for this subspace is given by the truncated power basis, $g_1, \dots g_n$, defined as

$$\begin{aligned} &g_1(x) = 1,\ g_2(x) = x,\ \ldots\ g_{k+1}(x) = x^k, \\ &g_{k+1+j}(x) = (x-t_j)^k \cdot 1\{x \geq t_j\}, \\ &\qquad\qquad\qquad j = 1, \dots n-k-1. \end{aligned} \tag{2}$$

These functions can also be used to define the truncated power basis matrix, $G \in \mathbb{R}^{n \times n}$, by

$$G_{ij} = g_j(x_i), \quad i, j = 1, \dots n, \tag{3}$$

Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32. Copyright 2014 by the author(s).

i.e., the columns of G give the evaluations of the basis functions $g_1, \ldots g_n$ over the inputs $x_1, \ldots x_n$. As $g_1, \ldots g_n$ are linearly independent functions, G has linearly independent columns, and hence G is invertible.

As noted, our focus is a related but different set of basis functions, named the falling factorial basis functions. We define these functions, for a given order $k \geq 0$, as

$$\begin{aligned} h_j(x) &= \prod_{\ell=1}^{j-1} (x - x_\ell), \quad j = 1, \ldots k+1, \\ h_{k+1+j}(x) &= \prod_{\ell=1}^{k} (x - x_{j+\ell}) \cdot 1\{x \geq x_{j+k}\}, \\ &\qquad\qquad j = 1, \ldots n-k-1. \end{aligned} \tag{4}$$

(Our convention is to take the empty product to be 1, so that $h_1(x) = 1$.) The falling factorial basis functions are piecewise polynomial, and have an analogous form to the truncated power basis functions in (2). Loosely speaking, they are given by replacing an rth order power function in the truncated power basis with an appropriate r-term product, e.g., replacing x^2 with $(x-x_2)(x-x_1)$, and $(x-t_j)^k$ with $(x - x_{j+k})(x - x_{j+k-1}) \cdot \ldots (x - x_{j+1})$. Similar to the above, we can define the falling factorial basis matrix, $H \in \mathbb{R}^{n \times n}$, by

$$H_{ij} = h_j(x_i), \quad i, j = 1, \ldots n, \tag{5}$$

and the linear independence of $h_1, \ldots h_n$ implies that H too is invertible.

Note that the first $k+1$ functions of either basis, the truncated power or falling factorial basis, span the same space (the space of kth order polynomials). But this is not true of the last $n-k-1$ functions. Direct calculation shows that, while continuous, the function h_{j+k+1} has discontinuous derivatives of all orders $1, \ldots k$ at the point x_{j+k}, for $j = 1, \ldots n-k-1$. This means that the falling factorial functions $h_{k+2}, \ldots h_n$ are not actually kth order splines, but are instead continuous kth order piecewise polynomials that are "close to" splines. Why would we ever use such a seemingly strange basis as that defined in (4)? To repeat what was summarized above, the falling factorial functions allow for linear-time (and closed-form) computations with the basis matrix H and its inverse. Meanwhile, the falling factorial functions are close enough to the truncated power functions that using them in several spline-based problems (i.e., using H in place of G) can be statistically legitimized. We make this statement precise in the sections that follow.

As we see it, there is really nothing about their form in (4) that suggests a particularly special computational structure of the falling factorial basis functions. Our interest in these functions arose from a study of trend filtering, a nonparametric regression estimator, where the inverse of H plays a natural role. The inverse of H is a kind of discrete derivative operator of order $k+1$, properly adjusted for the spacings between the input points $x_1, \ldots x_n$. It is really the special, banded structure of this derivative operator that underlies the computational efficiency surrounding the falling factorial basis; all of the computational routines proposed in this paper leverage this structure.

Here is an outline for rest of this article. In Section 2, we describe a number of basic properties of the falling factorial basis functions, culminating in fast linear-time algorithms for multiplication H and H^{-1}, and tight error bounds between H and the truncated power basis matrix G. Section 3 discusses B-splines, which provide another highly efficient basis for spline manipulations; we explain why the falling factorial basis offers a preferred parametrization in some specific statistical applications, e.g., the ones we present in Sections 4 and 5. Section 4 covers trend filtering, and extends a known convergence result for trend filtering over evenly spaced input points (Tibshirani, 2014) to the case of arbitrary input points. The conclusion is that trend filtering estimates converge at the minimax rate (over a large class of true functions) assuming only mild conditions on the inputs. In Section 5, we consider a higher order extension of the classic two-sample Kolmogorov-Smirnov test. We find this test to have better power in detecting higher order (tail) differences between distributions when compared to the usual Kolmogorov-Smirnov test; furthermore, by employing the falling factorial functions, it can computed in linear time. In Section 6, we end with some discussion.

2. Basic properties

Consider the falling factorial basis matrix $H \in \mathbb{R}^{n \times n}$, as defined in (5), over input points $x_1 < \ldots < x_n$. The following subsections describe a recursive decomposition for H and its inverse, which lead to fast computational methods for multiplication by H and H^{-1} (as well as H^T and $(H^T)^{-1}$). The last subsection bounds the maximum absolute difference bewteen the elements of H and G, the truncated power basis matrix (also defined over $x_1, \ldots x_n$). Lemmas 1, 2, 4 below were derived in Tibshirani (2014) for the special case of evenly spaced inputs, $x_i = i/n$ for $i = 1, \ldots n$. We reiterate that here we consider generic input points $x_1, \ldots x_n$. In the interest of space, we defer all proofs to a supplementary document.

2.1. Recursive decomposition

Our first result shows that H decomposes into a product of simpler matrices. It helpful to define, for $k \geq 1$,

$$\Delta^{(k)} = \mathrm{diag}\big(x_{k+1} - x_1,\, x_{k+2} - x_2,\, \ldots\, x_n - x_{n-k}\big),$$

the $(n-k) \times (n-k)$ diagonal matrix whose diagonal elements contain the k-hop gaps between input points.

Lemma 1. *Let I_m denote the $m \times m$ identity matrix, and L_m the $m \times m$ lower triangular matrix of 1s. If we write $H^{(k)}$ for the falling factorial basis matrix of order k, then in this notation, we have $H^{(0)} = L_n$, and for $k \geq 1$,*

$$H^{(k)} = H^{(k-1)} \cdot \begin{bmatrix} I_k & 0 \\ 0 & \Delta^{(k)} L_{n-k} \end{bmatrix}. \quad (6)$$

Lemma 1 is really a key workhorse behind many properties of the falling factorial basis functions. E.g., it acts as a building block for results to come: immediately, the representation (6) suggests both an analogous inverse representation for $H^{(k)}$, and a computational strategy for matrix multiplication by $H^{(k)}$. These are discussed in the next two subsections. We remark that the result in the lemma may seem surprising, as there is not an apparent connection between the falling factorial functions in (4) and the recursion in (6), which is based on taking cumulative sums at varying offsets (the rightmost matrix in (6)). We were led to this result by studying the evenly spaced case; its proof for the present case is considerably longer and more technical, but the statement of the lemma is still quite simple.

2.2. The inverse basis

The result in Lemma 1 clearly also implies a result on the inverse operators, namely, that $(H^{(0)})^{-1} = L_n^{-1}$, and

$$(H^{(k)})^{-1} = \begin{bmatrix} I_k & 0 \\ 0 & L_{n-k}^{-1} (\Delta^{(k)})^{-1} \end{bmatrix} \cdot (H^{(k-1)})^{-1} \quad (7)$$

for all $k \geq 1$. We note that

$$L_m^{-1} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 1 & 1 & \dots & 0 \\ \vdots & & & \\ 1 & 1 & \dots & 1 \end{bmatrix}^{-1} = \begin{bmatrix} e_1^T \\ D^{(1)} \end{bmatrix}, \quad (8)$$

with $e_1 = (1, 0, \dots 0) \in \mathbb{R}^m$ being the first standard basis vector, and $D^{(1)} \in \mathbb{R}^{(m-1) \times m}$ the first discrete difference operator

$$D^{(1)} = \begin{bmatrix} -1 & 1 & 0 & \dots & 0 & 0 \\ 0 & -1 & 1 & \dots & 0 & 0 \\ \vdots & & & & & \\ 0 & 0 & 0 & \dots & -1 & 1 \end{bmatrix}, \quad (9)$$

With this in mind, the recursion in (7) now looks like the construction of the higher order discrete difference operators, over the input $x_1, \dots x_n$. To define these operators, we start with the first order discrete difference operator $D^{(1)} \in \mathbb{R}^{(n-1) \times n}$ as in (9), and define the higher order difference discrete operators according to

$$D^{(k+1)} = D^{(1)} \cdot k \cdot (\Delta^{(k)})^{-1} \cdot D^{(k)}, \quad (10)$$

for $k \geq 1$. As $D^{(k+1)} \in \mathbb{R}^{(n-k-1) \times n}$, leading matrix $D^{(1)}$ above denotes the $(n-k-1) \times (n-k)$ version of the first order difference operator in (9).

To gather intuition, we can think of $D^{(k)}$ as a type of discrete kth order derivative operator across the underlying points $x_1, \dots x_n$; i.e., given an arbitrary sequence $u = (u_1, \dots u_n) \in \mathbb{R}^n$ over the positions $x_1, \dots x_n$, respectively, we can think of $(D^{(k)} u)_i$ as the discrete kth derivative of the sequence u evaluated at the point x_i. It is not difficult to see, from its definition, that $D^{(k)}$ is a banded matrix with bandwidth $k+1$. The middle (diagonal) term in (10) accounts for the fact that the underlying positions $x_1, \dots x_n$ are not necessarily evenly spaced. When the input points are evenly spaced, this term contributes only a constant factor, and the difference operators $D^{(k)}$, $k = 1, 2, 3, \dots$ take a very simple form, where each row is a shifted version of the previous, and the nonzero elements are given by the kth order binomial coefficients (with alternating signs); see Tibshirani (2014).

By staring at (7) and (10), one can see that the falling factorial basis matrices and discrete difference operators are essentially inverses of each other. The story is only slightly more complicated because the difference matrices are not square.

Lemma 2. *If $H^{(k)}$ is the kth order falling factorial basis matrix defined over the inputs $x_1, \dots x_n$, and $D^{(k+1)}$ is the $(k+1)$st order discrete difference operator defined over the same inputs $x_1 \dots x_n$, then*

$$(H^{(k)})^{-1} = \begin{bmatrix} C \\ \frac{1}{k!} \cdot D^{(k+1)} \end{bmatrix}, \quad (11)$$

for an explicit matrix $C \in \mathbb{R}^{(k+1) \times n}$. If we let A_i denote the ith row of a matrix A, then C has first row $C_1 = e_1^T$, and subsequent rows

$$C_{i+1} = \left[\frac{1}{(i-1)!} \cdot (\Delta^{(i)})^{-1} \cdot D^{(i)} \right]_1, \quad i = 1, \dots k.$$

Lemma 2 shows that the last $n-k-1$ rows of $(H^{(k)})^{-1}$ are given exactly by $D^{(k+1)}/k!$. This serves as the crucial link between the falling factorial basis functions and trend filtering, discussed in Section 4. The route to proving this result revealed the recursive expressions (6) and (7), and in fact these are of great computational interest in their own right, as we discuss next.

2.3. Fast matrix multiplication

The recursions in (6) and (7) allow us to apply $H^{(k)}$ and $(H^{(k)})^{-1}$ with specialized linear-time algorithms. Further, these algorithms are completely in-place: we do not need to form the matrices $H^{(k)}$ or $(H^{(k)})^{-1}$, and the algorithms

Algorithm 1 Multiplication by $H^{(k)}$

```
Input: Vector to be multiplied $y \in \mathbb{R}^n$, order $k \geq 0$, sorted inputs vector $x \in \mathbb{R}^n$.
Output: $y$ is overwritten by $H^{(k)}y$.
for $i = k$ to 0 do
    $y_{(i+1):n} = \mathrm{cumsum}(y_{(i+1):n})$,
    where $y_{a:b}$ denotes the subvector $(y_a, y_{a+1}, ..., y_b)$ and cumsum is the cumulative sum operator.
    if $i \neq 0$ then
        $y_{(i+1):n} = (x_{(i+1):n} - x_{1:(n-i)}) \;.\!*\; y_{(i+1):n}$,
        where $.*$ denotes entrywise multiplication.
    end if
end for
Return $y$.
```

Algorithm 2 Multiplication by $(H^{(k)})^{-1}$

```
Input: Vector to be multiplied $y \in \mathbb{R}^n$, order $k \geq 0$, sorted inputs vector $x \in \mathbb{R}^n$.
Output: $y$ is overwritten by $(H^{(k)})^{-1}y$.
for $i = 0$ to $k$ do
    if $i \neq 0$ then
        $y_{(i+1):n} = y_{i+1:n} \;./\; (x_{(i+1):n} - x_{1:(n-i]})$,
        where $./$ is entrywise division.
    end if
    $y_{(i+2):n} = \mathrm{diff}(y_{(i+1):n})$,
    where diff is the pairwise difference operator.
end for
Return $y$.
```

operate entirely by manipulating the input vector (the vector to be multiplied).

Lemma 3. *For the kth order falling factorial basis matrix $H^{(k)} \in \mathbb{R}^{n \times n}$, over arbitrary sorted inputs $x_1, \ldots x_n$, multiplication by $H^{(k)}$ and $(H^{(k)})^{-1}$ can each be computed in $O((k+1)n)$ in-place operations with zero memory requirements (aside from storing the inputs and the vector to be multiplied), i.e., we do not need to form $H^{(k)}$ or $(H^{(k)})^{-1}$. Algorithms 1 and 2 give the details. The same is true for matrix multiplication by $(H^{(k)})^T$ and $[(H^{(k)})^T]^{-1}$; Algorithms 3 and 4, found in the supplement, give the details.*

Note that the lemma assumes presorted inputs $x_1, \ldots x_n$ (sorting requires an extra $O(n \log n)$ operations). The routines for multiplication by $H^{(k)}$ and $(H^{(k)})^{-1}$, in Algorithms 1 and 2, are really just given by inverting each term one at a time in the product representations (6) and (7). They are composed of elementary in-place operations, like cumulative sums and pairwise differences. This brings to mind a comparison to wavelets, as both the wavelet and inverse wavelets operators can be viewed as highly specialized linear-time matrix multplications.

Borrowing from the wavelet perspective, given a sampled signal $y_i = f(x_i)$, $i = 1, \ldots n$, the action $(H^{(k)})^{-1}y$ can be thought of as the forward transform under the piecewise polynomial falling factorial basis, and $H^{(k)}y$ as the backward or inverse transform under this basis. It might be interesting to consider the applicability of such transforms to signal processing tasks, but this is beyond the scope of the current paper, and we leave it to potential future work.

We do however include a computational comparison between the forward and backward falling factorial transforms, in Algorithms 2 and 1, and the well-studied Fourier and wavelet transforms. Figure 1(a) shows the runtimes of one complete cycle of falling factorial transforms (i.e., one forward and one backward transform), with $k = 3$, versus one cycle of fast Fourier transforms and one cycle of wavelet transforms (using symmlets). The comparison was run in Matlab, and we used Matlab's "fft" and "ifft" functions for the fast Fourier transforms, and the Stanford WaveLab's "FWT_PO" and "IWT_PO" functions (with symmlet filters) for the wavelet transforms (Buckheit & Donoho, 1995). These functions all call on C implementations that have been ported to Matlab using MEX-functions, and so we did the same with our falling factorial transforms to even the comparison. For each problem size n, we chose evenly spaced inputs (this is required for the Fourier and wavelet transforms, but recall, not for the falling factorial transform), and averaged the results over 10 repetitions. The figure clearly demonstrates a linear scaling for the runtimes of the falling factorial transform, which matches their theoretical $O(n)$ complexity; the wavelet and fast fourier transforms also behave as expected, with the former having $O(n)$ complexity, and the latter $O(n \log n)$. In fact, a raw comparison of times shows that our implementation of the falling factorial transforms runs slightly faster than the highly-optimized wavelet transforms from the Stanford WaveLab.

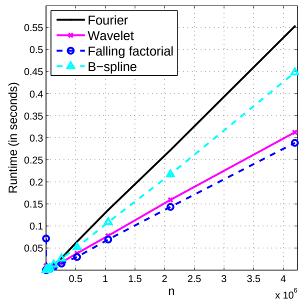
For completeness, Figure 1(b) displays a comparison between the falling factorial transforms and the corresponding transforms using the truncated power basis (also with $k = 3$). We see that the latter scale quadratically with n, which is again to be expected, as the truncated power basis matrix is essentially lower triangular.

2.4. Proximity to truncated power basis

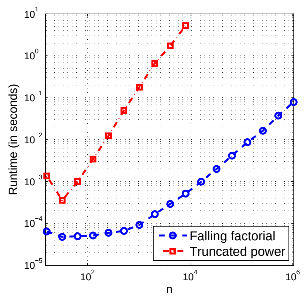
With computational efficiency having been assured by the last lemma, our next lemma lays the footing for the statistical credibility of the falling factorial basis.

Lemma 4. *Let $G^{(k)}$ and $H^{(k)}$ be the kth order truncated power and falling factorial matrices, defined over inputs $0 \leq x_1 < \ldots < x_n \leq 1$. Let $\delta = \max_{i=1,\ldots n}(x_i - x_{i-1})$, where we write $x_0 = 0$. Then*

$$\max_{i,j=1,\ldots n} |G^{(k)}_{ij} - H^{(k)}_{ij}| \leq k^2\delta.$$



(a) Falling factorial vs. Fourier, wavelet, and B-spline transforms (linear scale)



(b) Falling factorial (H) vs. truncated power (G) transforms (log-log scale)

Figure 1. Comparison of runtimes for different transforms. The experiments were performed on a laptop computer.

This tight elementwise bound between the two basis matrices will be used in Section 4 to prove a result on the convergence of trend filtering estimates. We will also discuss its importance in the context of a fast nonparametric two-sample test in Section 5. To give a preview: in many problem instances, the maximum gap δ between adjacent sorted inputs $x_1, \dots x_n$ is of the order $\log n/n$ (for a more precise statement see Lemma 5), and this means that the maximum absolute discrepancy between the elements of $G^{(k)}$ and $H^{(k)}$ decays very quickly.

3. Why not just use B-splines?

B-splines already provide a computationally efficient parametrization for the set of kth order splines; i.e., since they produce banded basis matrices, we can already perform linear-time basis matrix multiplication and inversion with B-splines. To confirm this point empirically, we included B-splines in the timing comparison of Section 2.3, refer to Figure 1(a) for the results. So, why not always use B-splines in place of the falling factorial basis, which only approximately spans the space of splines?

A major reason is that the falling factorial functions (like the truncated power functions) admit a sparse representation under the total variation operator, whereas the B-spline functions do not. To be more specific, suppose that $f_1, \dots f_m$ are kth order piecewise polynomial functions with knots at the points $0 \leq z_1 < \dots < z_r \leq 1$, where $m = r + k + 1$. Then, for $f = \sum_{j=1}^m \alpha_j f_j$, we have

$$\mathrm{TV}(f^{(k)}) = \sum_{i=1}^{r} \left| \sum_{j=1}^{m} \Big(f_j^{(k)}(z_i) - f_j^{(k)}(z_{i-1}) \Big) \cdot \alpha_j \right|,$$

denoting $z_0 = 0$ for ease of notation. If $f_1, \dots f_m$ are the falling factorial functions defined over the points $z_1, \dots z_r$, then the term $f_j^{(k)}(z_i) - f_j^{(k)}(z_{i-1})$ is equal to 0 for all i, j, except when $i = j - k - 1$ and $j \geq k + 2$, in which case it equals 1. Therefore, $\mathrm{TV}(f^{(k)}) = \sum_{j=k+2}^m |\alpha_j|$, a simple sum of absolute coefficients in the falling factorial expansion. The same result holds for the truncated power basis functions. But if $f_1, \dots f_m$ are B-splines, then this is not true; one can show that in this case $\mathrm{TV}(f^{(k)}) = \|C\alpha\|_1$, where C is a (generically) dense matrix. The fact that C is dense makes it cumbersome, both mathematically and computationally, to use the B-spline parametrization in spline problems involving total variation, such as those discussed in Sections 4 and 5.

4. Trend filtering for arbitrary inputs

Trend filtering is a relatively new method for nonparametric regression. Suppose that we observe

$$y_i = f_0(x_i) + \epsilon_i, \quad i = 1, \dots n, \tag{12}$$

for a true (unknown) regression function f_0, inputs $x_1 < \dots < x_n \in \mathbb{R}$, and errors $\epsilon_1, \dots \epsilon_n$. The trend filtering estimator was first proposed by Kim et al. (2009), and further studied by Tibshirani (2014). In fact, the latter work motivated the current paper, as it derived properties of the falling factorial basis over evenly spaced inputs $x_i = i/n$, $i = 1, \dots n$, and use these to prove convergence rates for trend filtering estimators. In the present section, we allow $x_1, \dots x_n$ to be arbitrary, and extend the convergence guarantees for trend filtering, utilizing the properties of the falling factorial basis derived in Section 2.

The trend filtering estimate $\hat{\beta}$ of order $k \geq 0$ is defined by

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^n}{\mathrm{argmin}} \; \frac{1}{2}\|y - \beta\|_2^2 + \lambda \cdot \frac{1}{k!}\|D^{(k+1)}\beta\|_1, \tag{13}$$

where $y = (y_1, \dots y_n) \in \mathbb{R}^n$, $D^{(k+1)} \in \mathbb{R}^{(n-k-1)\times n}$ is the $(k+1)$st order discrete difference operator defined in (10) over the input points $x_1, \dots x_n$, and $\lambda \geq 0$ is a tuning parameter. We can think of the components of $\hat{\beta}$ as defining an estimated function $\hat{f}$ over the input points. To give an example, in Figure 2, we drew noisy observations from a smooth underlying function, where the input points $x_1, \dots x_n$ were sampled uniformly at random over $[0, 1]$, and we computed the trend filtering estimate $\hat{\beta}$ with $k = 3$ and a particular choice of λ. From the plot (where we interpolated between $(x_1, \hat{\beta}_1), \dots (x_n, \hat{\beta}_n)$ for visualization purposes), we can see that the implicitly defined trend filtering function $\hat{f}$ displays a piecewise cubic structure, with adaptively chosen knot points. Lemma 2 makes this connection precise by showing that such a function $\hat{f}$ is indeed a linear combination of falling factorial functions. Letting $\beta = H^{(k)}\alpha$, where $H^{(k)} \in \mathbb{R}^{n\times n}$ is the kth order falling factorial basis matrix defined over the inputs $x_1, \dots x_n$, the trend filtering problem in (13) becomes

$$\hat{\alpha} = \underset{\alpha \in \mathbb{R}^n}{\mathrm{argmin}} \; \frac{1}{2}\|y - H^{(k)}\alpha\|_2^2 + \lambda \cdot \sum_{j=k+2}^{n} |\alpha_j|, \tag{14}$$

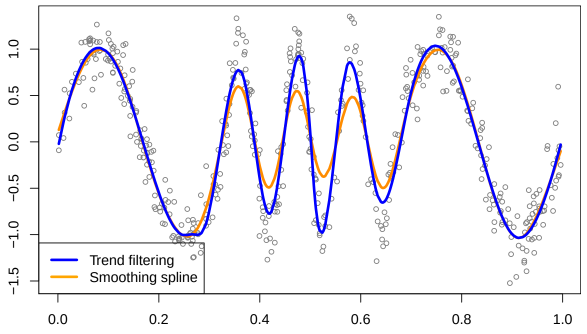


Figure 2. Example trend filtering and smoothing spline estimates.

equivalent to the functional minimization problem

$$\hat{f} = \underset{f \in \mathcal{H}_k}{\operatorname{argmin}} \ \frac{1}{2} \sum_{i=1}^{n} \big(y_i - f(x_i)\big)^2 + \lambda \cdot \mathrm{TV}\big(f^{(k)}\big), \quad (15)$$

where $\mathcal{H}_k = \mathrm{span}\{h_1, \ldots h_n\}$ is the span of the kth order falling factorial functions in (4), $\mathrm{TV}(\cdot)$ denotes the total variation operator, and $f^{(k)}$ denotes the kth weak derivative of f. In other words, the solutions of problems (13) and (15) are related by $\hat{\beta}_i = \hat{f}(x_i)$, $i = 1, \ldots n$. The trend filtering estimate hence verifiably exhibits the structure of a kth order piecewise polynomial function, with knots at a subset of $x_1, \ldots x_n$, and this function is not necessarily a spline, but is close to one (since it lies in the span of the falling factorial functions $h_1, \ldots h_n$).

In Figure 2, we also fit a smoothing spline estimate to the same example data. A striking difference: the trend filtering estimate is far more locally adaptive towards the middle of plot, where the underlying function is less smooth (the two estimates were tuned to have the same degrees of freedom, to even the comparison). This phenomenon is investigated in Tibshirani (2014), where it is shown that trend filtering estimates attain the minimax convergence rate over a large class of underlying functions, a class for which it is known that smoothing splines (along with any other estimator linear in y) are suboptimal. This latter work focused on evenly spaced inputs, $x_i = i/n$, $i = 1, \ldots n$, and the next two subsections extend the trend filtering convergence theory to cover arbitrary inputs $x_1, \ldots x_n \in [0, 1]$. We first consider the input points as fixed, and then random. All proofs are deferred until the supplement.

4.1. Fixed input points

The following is our main result on trend filtering.

Theorem 1. *Let $y \in \mathbb{R}^n$ be drawn from (12), with fixed inputs $0 \le x_1 < \ldots < x_n \le 1$, having a maximum gap*

$$\max_{i=1,\ldots n} (x_i - x_{i-1}) = O(\log n / n), \quad (16)$$

and i.i.d., mean zero sub-Gaussian errors. Assume that, for an integer $k \ge 0$ and constant $C > 0$, the true function f_0 is k times weakly differentiable, with $\mathrm{TV}(f_0^{(k)}) \le C$. Then the kth order trend filtering estimate $\hat{\beta}$ in (13), with tuning parameter value $\lambda = \Theta(n^{1/(2k+3)})$, satisfies

$$\frac{1}{n} \sum_{i=1}^{n} \big(\hat{\beta}_i - f_0(x_i)\big)^2 = O_{\mathbb{P}}(n^{-(2k+2)/(2k+3)}). \quad (17)$$

Remark 1. The rate $n^{-(2k+2)/(2k+3)}$ is the minimax rate of convergence with respect to the class of k times weakly differentiable functions f such that $\mathrm{TV}(f^{(k)}) \le C$ (see, e.g., Nussbaum (1985), Tibshirani (2014)). Hence Theorem 1 shows that trend filtering estimates converge at the minimax rate over a broad class of true functions f_0, assuming that the fixed input points are not too irregular, in that the maximum adjacent gap between points must satisfy (16). This condition is not stringent and is naturally satisfied by continuously distributed random inputs, as we show in the next subsection. We note that Tibshirani (2014) proved the same conclusion (as in Theorem 1) for unevenly spaced inputs $x_1, \ldots x_n$, but placed very complicated and basically uninterpretable conditions on the inputs. Our tighter analysis of the falling factorial functions yields the simple sufficient condition (16).

Remark 2. The conclusion in the theorem can be strengthened, beyond the the convergence of $\hat{\beta}$ to f_0 in (17); under the same assumptions, the trend filtering estimate $\hat{\beta}$ also converges to $\hat{f}^{\text{spline}}$ at the same rate $n^{-(2k+2)/(2k+3)}$, where we write $\hat{f}^{\text{spline}}$ to denote the solution in (15) with $\mathcal{H}_k$ replaced by $\mathcal{G}_k = \mathrm{span}\{g_1, \ldots g_n\}$, the span of the truncated power basis functions in (2). This asserts that the trend filtering estimate is indeed "close to" a spline, and here the bound in Lemma 4, between the truncated power and falling factorial basis matrices, is key. Moreover, we actually rely on the convergence of $\hat{\beta}$ to $\hat{f}^{\text{spline}}$ to establish (17), as the total variation regularized spline estimator $\hat{f}^{\text{spline}}$ is already known to converge to f_0 at the minimax rate (Mammen & van de Geer, 1997).

4.2. Random input points

To analyze trend filtering for random inputs, $x_1, \ldots x_n$, we need to bound the maximum gap between adjacent points with high probability. Fortunately, this is possible for a large class of distributions, as shown in the next lemma.

Lemma 5. *If $0 \le x_1 < \ldots < x_n \le 1$ are sorted i.i.d. draws from an arbitrary continuous distribution supported on $[0, 1]$, whose density is bounded below by $p_0 > 0$, then with probability at least $1 - 2p_0 n^{-10}$,*

$$\max_{i=1,\ldots n} (x_i - x_{i-1}) \le \frac{c_0 \log n}{p_0 n},$$

for a universal constant c_0.

The proof of this result is readily assembled from classical results on order statistics; we give a simple alternate proof in the supplement. Lemma 5 implies the next corollary.

Corollary 1. *Let $y \in \mathbb{R}^n$ be distributed according to the model* (12)*, where the inputs $0 \le x_1 < \ldots < x_n \le 1$ are sorted i.i.d. draws from an arbitrary continuous distribution on* $[0,1]$*, whose density is bounded below. Assume again that the errors are i.i.d., mean zero sub-Gaussian variates, independent of the inputs, and that the true function f_0 has k weak derivatives and satisfies $\mathrm{TV}(f_0^{(k)}) \le C$. Then, for $\lambda = \Theta(n^{1/(2k+3)})$, the kth order trend filtering estimate $\hat{\beta}$ converges at the same rate as in Theorem 1.*

5. A higher order Kolmogorov-Smirnov test

The two-sample Kolmogorov-Smirnov (KS) test is a standard nonparametric hypothesis test of equality between two distributions, say $\mathbb{P}_X$ and $\mathbb{P}_Y$, from independent samples $x_1, \ldots x_m \sim \mathbb{P}_X$ and $y_1, \ldots y_n \sim \mathbb{P}_Y$. Writing $X_{(m)} = (x_1, \ldots x_m)$, $Y_{(n)} = (y_1, \ldots y_n)$, and $Z_{(m+n)} = (z_1, \ldots z_{m+n}) = X_{(m)} \cup Y_{(n)}$ for the joined samples, the KS statistic can be expressed as

$$\mathrm{KS}(X_{(m)}, Y_{(n)}) = \max_{z_j \in Z_{(m+n)}} \left| \frac{1}{m} \sum_{i=1}^m 1\{x_i \le z_j\} - \frac{1}{n} \sum_{i=1}^n 1\{y_i \le z_j\} \right|. \tag{18}$$

This examines the maximum absolute difference between the empirical cumulative distribution functions from $X_{(m)}$ and $Y_{(n)}$, across all points in the joint set $Z_{(m+n)}$, and so the test rejects for large values of (18). A well-known alternative (variational) form for the KS statistic is

$$\mathrm{KS}(X_{(m)}, Y_{(n)}) = \max_{f \,:\, \mathrm{TV}(f) \le 1} \left| \hat{\mathbb{E}}_{X_{(m)}}[f(X)] - \hat{\mathbb{E}}_{Y_{(n)}}[f(Y)] \right|, \tag{19}$$

where $\hat{\mathbb{E}}_{X_{(m)}}$ denotes the empirical expectation under $X_{(m)}$, so that $\hat{\mathbb{E}}_{X_{(m)}}[f(X)] = 1/m \sum_{i=1}^m f(x_i)$, and similarly for $\hat{\mathbb{E}}_{Y_{(n)}}$. The equivalence between (19) and (18) comes from the fact that maximum in (19) is achieved by taking f to be a step function, with its knot (breakpoint) at one of the joined samples $z_1, \ldots z_{m+n}$.

The KS test is perhaps one of the most widely used nonparametric tests of distributions, but it does have its shortcomings. Loosely speaking, it is known to be sensitive in detecting differences between the centers of distributions $\mathbb{P}_X$ and $\mathbb{P}_Y$, but much less sensitive in detecting differences in the tails. In this section, we generalize the KS test to "higher order" variants that are more powerful than the original KS test in detecting tail differences (when, of course, such differences are present). We first define the higher order KS test, and describe how it can be computed in linear time with the falling factorial basis. We then empirically compare these higher order versions to the original KS test, and several other commonly used nonparametric two-sample tests of distributions.

5.1. Definition of the higher order KS tests

For a given order $k \ge 0$, we define the kth order KS test statistic between $X_{(m)}$ and $Y_{(n)}$ as

$$\mathrm{KS}_G^{(k)}(X_{(m)}, Y_{(n)}) = \left\| (G_2^{(k)})^T \left(\frac{\mathbb{1}_{X_{(m)}}}{m} - \frac{\mathbb{1}_{Y_{(n)}}}{n} \right) \right\|_\infty. \tag{20}$$

Here $G^{(k)} \in \mathbb{R}^{(m+n)\times(m+n)}$ is the kth order truncated power basis matrix over the joined samples $z_1 < \ldots < z_{m+n}$, assumed sorted without a loss of generality, and $G_2^{(k)}$ is the submatrix formed by excluding its first $k+1$ columns. Also, $\mathbb{1}_{X_{(m)}} \in \mathbb{R}^{(m+n)}$ is a vector whose components indicate the locations of $x_1 < \ldots < x_m$ among $z_1 < \ldots < z_{m+n}$, and similarly for $\mathbb{1}_{Y_{(n)}}$. Finally, $\|\cdot\|_\infty$ denotes the ℓ_∞ norm, $\|u\|_\infty = \max_{i=i,\ldots r} |u_i|$ for $u \in \mathbb{R}^r$.

As per the spirit of our paper, an alternate definition for the kth order KS statistic uses the falling factorial basis,

$$\mathrm{KS}_H^{(k)}(X_{(m)}, Y_{(n)}) = \left\| (H_2^{(k)})^T \left(\frac{\mathbb{1}_{X_{(m)}}}{m} - \frac{\mathbb{1}_{Y_{(n)}}}{n} \right) \right\|_\infty, \tag{21}$$

where now $H^{(k)} \in \mathbb{R}^{(m+n)\times(m+n)}$ is the kth order falling factorial basis matrix over the joined samples $z_1 < \ldots < z_{m+n}$. Not surprisingly, the two definitions are very close, and Hölder's inequality shows that

$$\begin{aligned} &|\mathrm{KS}_G^{(k)}(X_{(m)}, Y_{(n)}) - \mathrm{KS}_H^{(k)}(X_{(m)}, Y_{(n)})| \\ &\qquad \le \max_{i,j=1,\ldots m+n} 2|G_{ij}^{(k)} - H_{ij}^{(k)}| \le 2k^2\delta, \end{aligned}$$

the last inequality due to Lemma 4, with δ the maximum gap between $z_1, \ldots z_{m+n}$. Recall that Lemma 5 shows δ to be of the order $\log(m+n)/(m+n)$ for continuous distributions $\mathbb{P}_X, \mathbb{P}_Y$ supported nontrivially on $[0,1]$, which means that with high probability, the two definitions differ by at most $2k^2 \log(m+n)/(m+n)$, in such a setup.

The advantage to using the falling factorial definition is that the test statistic in (21) can be computed in $O((k+1)(m+n))$ time, without even having to form the matrix $H_2^{(k)}$ (this is assuming sorted points $z_1, \ldots z_{m+n}$). See Lemma 3, and Algorithm 3 in the supplement. By comparison, the statistic in (20) requires $O((m+n)^2)$ operations. In addition to the theoretical bound described above, we also find empirically that the two definitions perform quite similarly, as shown in the next subsection, and hence we advocate the use of $\mathrm{KS}_H^{(k)}$ for computational reasons.

A motivation for our proposed tests is as follows: it can be shown that (20), and therefore (21), approximately take a

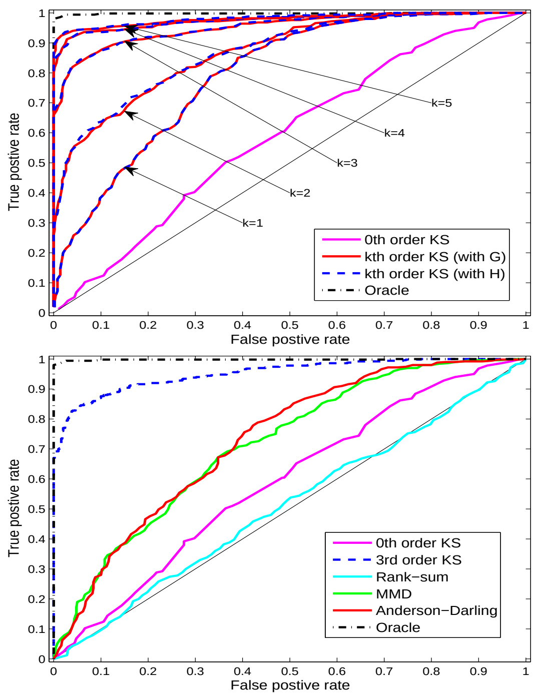


Figure 3. ROC curves for experiment 1, normal vs. t.

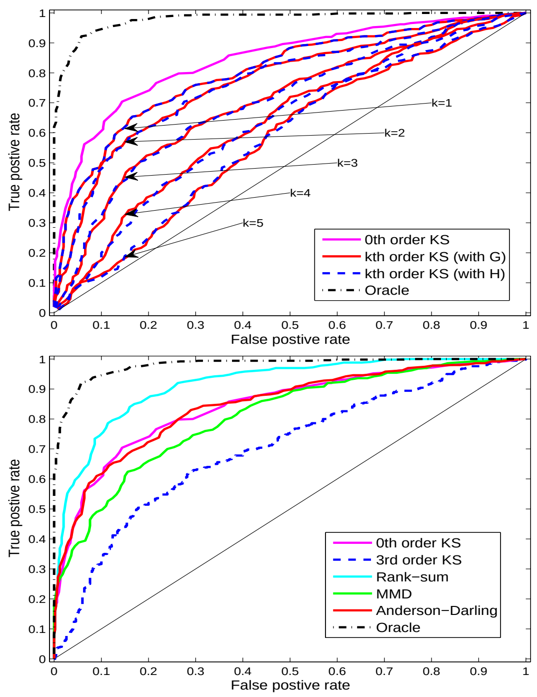


Figure 4. ROC curves for experiment 2, Laplace vs. Laplace.

variational form similar to (19), but where the constraint is over functions whose kth (weak) derivative has total variation at most 1. See the supplement.

5.2. Numerical experiments

We examine the higher order KS tests by simulation. The setup: we fix two distributions P, Q. We draw n i.i.d. samples $X_{(n)}, Y_{(n)} \sim P$, calculate a test statistic, and repeat this $R/2$ times; we also draw n i.i.d. samples $X_{(n)} \sim P$, $Y_{(n)} \sim Q$, calculate a test statistic, and repeat $R/2$ times. We then construct an ROC curve, i.e., the true positive rate versus the false positive rate of the test, as we vary its rejection threshold. For the test itself, we consider our kth order KS test, in both its G and H forms, as well as the usual KS test, and a number of other popular two-sample tests: the Anderson-Darling test (Anderson & Darling, 1954; Scholz & Stephens, 1987), the Wilcoxon rank-sum test (Wilcoxon, 1945), and the maximum mean discrepancy (MMD) test, with RBF kernel (Gretton et al., 2012).

Figures 3 and 4 show the results of two experiments in which $n = 100$ and $R = 1000$. (See the supplement for more experiments.) In the first we used $P = N(0, 1)$ and $Q = t_3$ (t-distribution with 3 degrees of freedom), and in the second $P = \text{Laplace}(0)$ and $Q = \text{Laplace}(0.3)$ (Laplace distributions of different means). We see that our proposed kth order KS test performs favorably in the first experiment, with its power increasing with k. When $k = 3$, it handily beats all competitors in detecting the difference between the standard normal distribution and the heavier-tailed t-distribution. But there is no free lunch: in the second experiment, where the differences between P, Q are mostly near the centers of the distributions and not in the tails, we can see that increasing k only decreases the power of the kth order KS test. In short, one can view our proposal as introducing a family of tests parametrized by k, which offer a tradeoff in center versus tail sensitivity. A more thorough study will be left to future work.

6. Discussion

We formally proposed and analyzed the spline-like falling factorial basis functions. These basis functions admit attractive computational and statistical properties, and we demonstrated their applicability in two problems: trend filtering, and a novel higher order variant of the KS test. These examples, we feel, are just the beginning. As typical operations associated with the falling factorial basis scale merely linearly with the input size (after sorting), we feel that this basis may be particularly well-suited to a rich number of large-scale applications in the modern data era, a direction that we are excited to pursue in the future.

Acknowledgements This work was partially supported by NSF Grant DMS-1309174, a Google Research Grant, and the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office.

Supplement This paper and its accompanying supplement (with proofs and extra experiments) can be found on arXiv.

References

Anderson, Theodore and Darling, Donald. A test of goodness of fit. *Journal of the American Statistical Association*, 49(268):765–769, 1954.

Buckheit, Jonathan and Donoho, David. Wavelab and reproducible research. *Lecture Notes in Statistics*, 103:55–81, 1995.

de Boor, Carl. *A Practical Guide to Splines*. Springer, New York, 1978.

Gretton, Arthur, Borgwardt, Karsten, Rasch, Malte, Schölkopf, Bernhard, and Smola, Alexander. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773, 2012.

Kim, Seung-Jean, Koh, Kwangmoo, Boyd, Stephen, and Gorinevsky, Dimitry. ℓ_1 trend filtering. *SIAM Review*, 51(2):339–360, 2009.

Mammen, Enno and van de Geer, Sara. Locally apadtive regression splines. *Annals of Statistics*, 25(1):387–413, 1997.

Nussbaum, Michael. Spline smoothing in regression models and asymptotic efficiency in L_2. *Annals of Statistics*, 13(3):984–997, 1985.

Scholz, Fritz and Stephens, Michael. K-sample Anderson-Darling tests. *Journal of the American Statistical Association*, 82(399):918–924, 1987.

Tibshirani, Ryan J. Adaptive piecewise polynomial estimation via trend filtering. *Annals of Statistics*, 42(1): 285–323, 2014.

Wahba, Grace. *Spline Models for Observational Data*. Society for Industrial and Applied Mathematics, Philadelphia, 1990.

Wilcoxon, Frank. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.