Convex Regression

February 17, 2025

In this homework, we'll work through the idea and implementation of *convex regression*. We will focus on the one-dimensional case, although it extends very naturally to higher dimensions. Then we'll look into rates of convergence, comparing this new method to the stuff we've been using.

In my code, I'll be using a few libraries.

```
library(CVXR)
CVXR::add_to_solver_blacklist('OSQP')
# OSQP claims some feasible problems aren't
```

And some functions we've been using in labs.

```
invert.unique = function(x) {
  o = order(x)
  dup = duplicated(x[o])
  inverse = rep(NA, length(x))
  inverse[o] = cumsum(!dup)
  list(elements=o[!dup], inverse=inverse)
}

prediction.function = function(model) {
  function(x) { predict(model, data.frame(X=x)) }
}
```

1 Convexity

In some cases, we may believe that a curve we want to estimate is *convex*. A differentiable curve m is convex if its derivative is increasing.[1] More generally, a curve m is convex if all of its secants (line segments drawn from one point on the curve to another) lie above the curve, i.e., if for all points a and b it satisfies

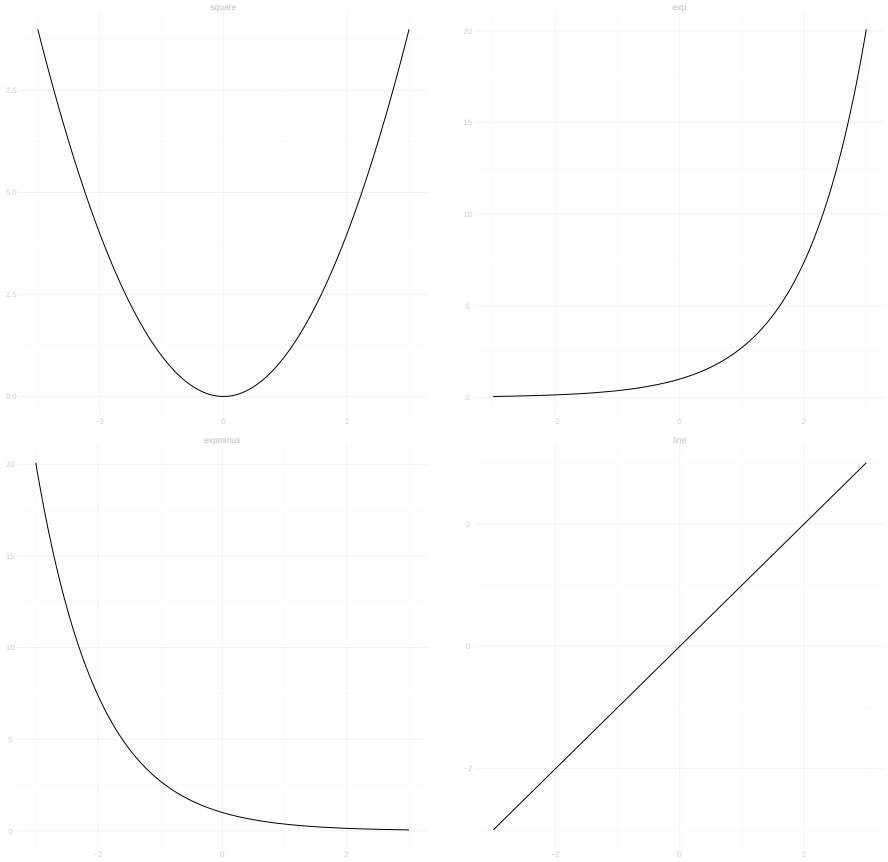
$$m\{(1-\lambda)a + \lambda b\} \leq (1-\lambda)m(a) + \lambda m(b) \quad \text{for all} \quad \lambda \in [0,1]. \tag{1}$$

This inequality is, in mathematical notation instead of visual language, exactly what we said about secants. The left side is the height of the curve at $x_\lambda = (1-\lambda)a + \lambda b$ and the right is the height of the secant connecting a to b at x_λ.

Characterizing points on secants. Keep in mind that any point x on the segment between a and b can be written in the form $x_\lambda = (1-\lambda)a + \lambda b$ for $\lambda \in [0,1]$. To do this, we simply solve the equation $x_\lambda = (1-\lambda)a + \lambda b$ for λ in terms of x, i.e., we take $\lambda = (x-a)/(b-a)$. This is pretty intuitive: λ is the fraction of the distance from a to b that we have to travel to get from a to x.

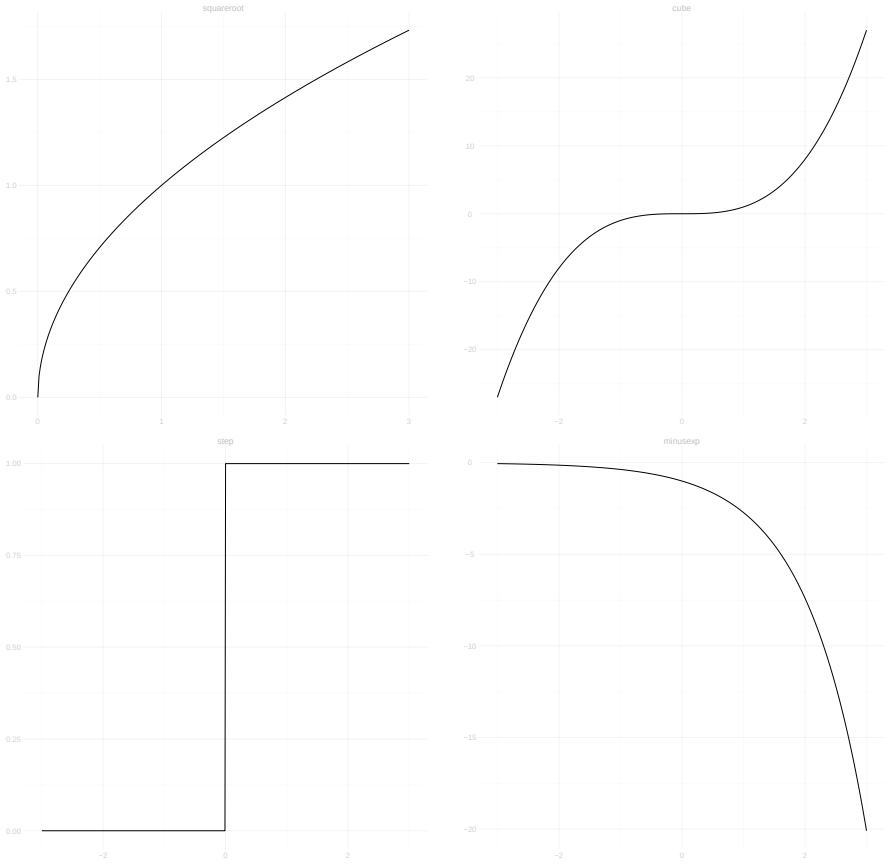
Examples. Here are some examples of convex curves.

1. $f(x) = x^2$
2. $f(x) = e^x$
3. $f(x) = e^{-x}$
4. $f(x) = x$



Here are a few curves that aren't convex.

1. $f(x) = \sqrt{x}$
2. $f(x) = (x - 1/2)^3$
3. $f(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$
4. $f(x) = -e^x$



Exercise 1 *On the eight plots above, draw a few secants. For the non-convex curves, make sure at least one is below the curve somewhere between the secant's endpoints.*

1.1 Differentiable Convex Functions

Now that we've got a sense of what's going on visually, let's argue that our more general definition based on (1) is consistent with the informal definition based

on derivatives I used in our first lecture.

Exercise 2 *Explain why, if a curve $m(x)$ is differentiable, it satisfies* (1) *if and only if its derivative $m'(x)$ is increasing.*

Hint. Here are two equivalent statements we can derive from (1) by taking $\lambda = (x-a)/(b-a)$ and $\lambda = 1-(x-b)/(b-a)$ respectively.

$$\begin{aligned} m(x) &\le m(a) + \frac{m(b)-m(a)}{b-a}(x-a) \quad \text{for all } x \in [a,b] \\ m(x) &\le m(b) + \frac{m(b)-m(a)}{b-a}(x-b) \quad \text{for all } x \in [a,b]. \end{aligned} \tag{2}$$

Rearranging, we get inequalities relating two slopes, one of which is the same in both cases.

$$\begin{aligned} \frac{m(x)-m(a)}{x-a} &\le \frac{m(b)-m(a)}{b-a} && \text{for all } x \in [a,b] \\ \frac{m(b)-m(a)}{b-a} &\le \frac{m(b)-m(x)}{b-x} = \frac{m(x)-m(b)}{x-b} && \text{for all } x \in [a,b]. \end{aligned} \tag{3}$$

What do these two equations together imply if we take $x \to a$ in the first and $x \to b$ in the second? This should help you show that convexity in the sense of (1) implies the increasingness of the derivative.

Another Hint. The mean value theorem tells us that, letting $x_\lambda = (1-\lambda)a + \lambda b$,

$$\begin{aligned} \frac{f(x_\lambda)-f(a)}{x_\lambda - a} &= f'(\tilde{a}) \quad \text{for some point} \quad \tilde{a} \in [a, x_\lambda] \\ \frac{f(b)-f(x_\lambda)}{b-x_\lambda} &= f'(\tilde{b}) \quad \text{for some point} \quad \tilde{b} \in [x_\lambda, b] \end{aligned} \tag{4}$$

If f' is increasing, how are these ratios related? And what, in terms of λ, a, and b, are their denominators? This should help you show that the increasingness of the derivative implies convexity in the sense of (1).

Solution 2 *Let's start with the only if part, i.e., showing that convexity implies increasingness of the derivative. We'll use the first hint. If f is convex,* (3) *is true. Taking the limit as $x \to a$ in the first inequality in* (3) *gives us the inequality $m'(a) \le \{m(b)-m(a)\}/(b-a)$. Taking the limit as $x \to b$ in the second gives us $\{m(b)-m(a)\}/(b-a) \le m'(b)$. It follows that $f'(a) \le f'(b)$.*

Now let's do the if part, i.e., showing that increasingness of the derivative implies convexity. We'll use the second hint. Because the derivative is increasing,

$$\frac{f(x_\lambda)-f(a)}{x_\lambda - a} = f'(\tilde{a}) \le f'(\tilde{b}) = \frac{f(b)-f(x_\lambda)}{b-x_\lambda}.$$

Here the denominators are $x_\lambda - a = \{(1-\lambda)a+\lambda b\} - a = \lambda(b-a)$ and $b - x_\lambda = b - \{(1-\lambda)a+\lambda b\} = (1-\lambda)(b-a)$, so dropping the common factor of $1/(b-a)$,

we can rephrase this inequality as $\{f(x_\lambda) - f(a)\}/\lambda \leq \{f(b) - f(x_\lambda)\}/(1-\lambda)$. *Multiplying by* $\lambda(1-\lambda)$, *this gives* $\{f(x_\lambda) - f(a)\}(1-\lambda) \leq \{f(b) - f(x_\lambda)\}\lambda$, *and adding* $(1-\lambda)f(a) + \lambda f(x_\lambda)$ *to both sides to rearrange, we get* $f(x_\lambda)\{1-\lambda+\lambda\} \leq (1-\lambda)f(a) + \lambda f(b)$.

1.2 Convex Sets

There's a related notion of a *convex set*. We won't be using this for convex regression part of this homework, but it'll come up in lecture soon.

A convex set is a set that contains all line segments between points in it. That is, a set $\mathcal{S}$ is convex if and only if, for all points $a, b \in \mathcal{S}$, $(1-\lambda)a + \lambda b \in \mathcal{S}$ for all $\lambda \in [0, 1]$. Here are a few examples.

In $1D$. A point, a line segment, or a line.

In $2D$. A filled-in triangle, square, or circle; the positive half-plane $\{(x, y) \in \mathbb{R}^2 : y \geq 0\}$; or the whole of $\mathbb{R}^2$.

Generally. A ball, the set $\{v : \rho(v) \leq r\}$, of any radius r in any seminorm ρ.

Here are a few sets that aren't convex.

In $1D$. Two points. Or the union of two disconnected intervals, e.g. $\{x : x \in [-1, 0] \text{ or } [1, 2]\}$.

In $2D$. A not-filled-in triangle, square, or circle.

In $3D$. A sphere, the set $\{v : \|v\| = r\}$, of any radius $r > 0$ in any norm.

Exercise 3 *Prove that a ball in a seminorm* ρ *is convex.*

Tip. Use the triangle inequality.

Solution 3 *If* a *and* b *are any points in a ball of seminorm of radius* r, *then so is* $(1-\lambda)a + \lambda b$ *for any* $\lambda \in [0, 1]$, *as*

$$\rho\{(1-\lambda)a + \lambda b\} \leq \rho\{(1-\lambda)a\} + \rho\{\lambda b\} = (1-\lambda)\rho(a) + \lambda\rho(b) \leq (1-\lambda)r + \lambda r = r.$$

Exercise 4 *Using the norms we discussed in our Vector Spaces Homework, explain why that implies that the filled-in square* $\{(x, y) : |x| \leq 1, |y| \leq 1\}$, *circle* $\{(x, y) : x^2 + y^2 \leq 1\}$, *and diamond* $\{(x, y) : |x| + |y| \leq 1\}$ *are convex.*

Solution 4 *They are the unit balls of the infinity-norm, two-norm, and one-norm respectively.*

Exercise 5 *Prove that a sphere of nonzero radius in any norm is not convex.*

Tip. Revisit the proof that seminorms are positive from the Vector Spaces Homework.

Solution 5 *If v is any point on such a sphere, so is $-v$, as $\|-v\| = |-1|\|v\| = \|v\|$. If the sphere were convex, this would imply that it contains the point $0 = (1/2)v + (1/2)(-v)$, but zero cannot be in a sphere of any nonzero radius as $\|0\| = 0$.*

Exercise 6 *Draw, in 2D, a non-convex set that isn't included in the examples above.*

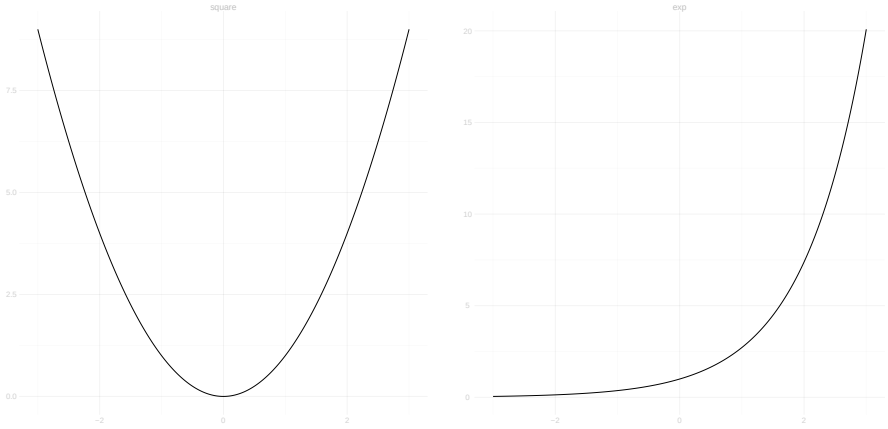
Exercise 7 *Explain why the intersection of two convex sets, i.e. the set of points that are in both of them, is a convex set.*

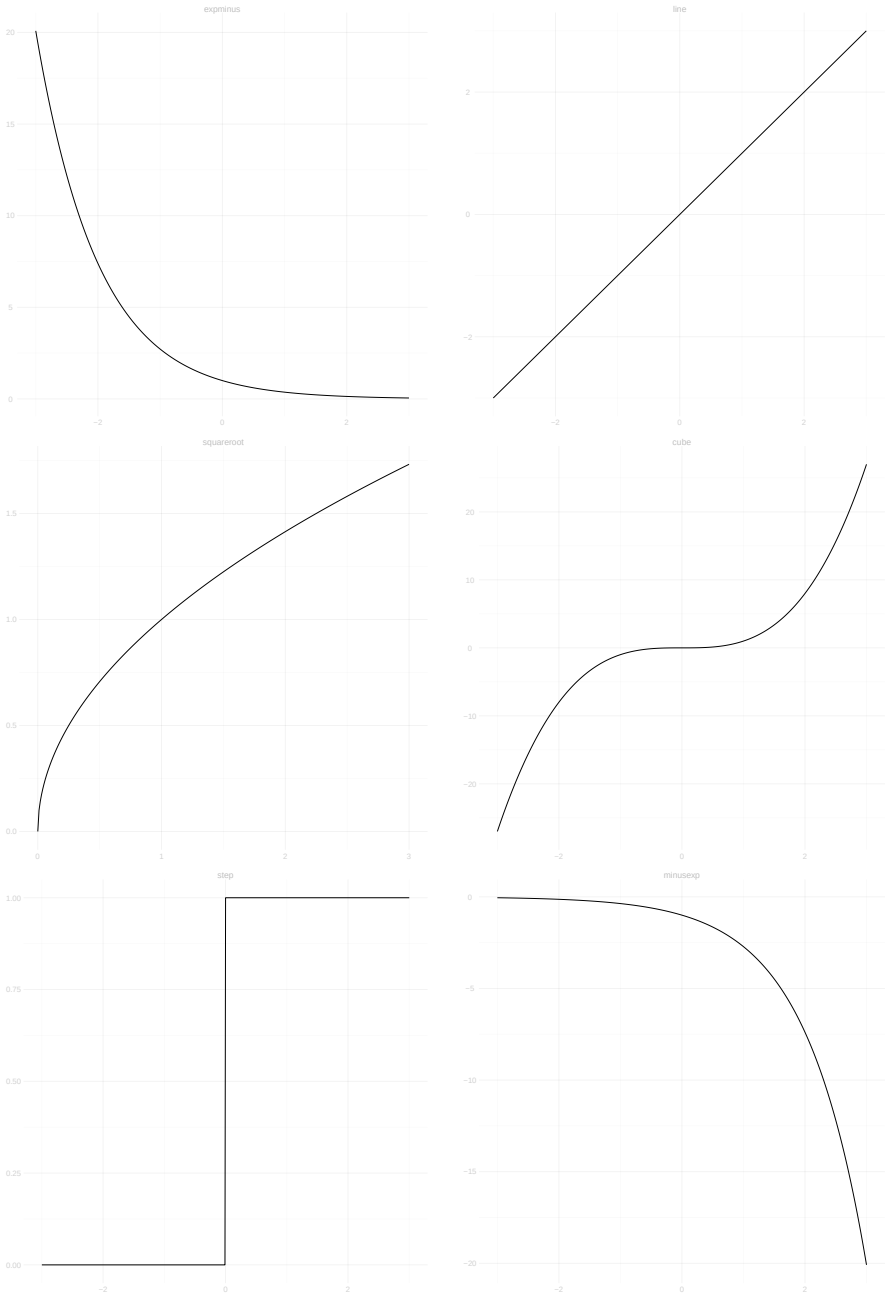
Solution 7 *The segment between any two points in the intersection is, as a result of the convexity of each set, is in both sets and therefore in the intersection.*

1.3 Convex functions have convex epigraphs

Here's another way of thinking about what convex functions look like. A function is convex if and only if its *epigraph*, the set of points on or above the curve, is convex. This is the definition of the epigraph of a function in mathematical notation. $\text{Epi}(f) = \{(x, y) : y \geq f(x)\}$.

Exercise 8 *On the plots below, fill in the epigraph.*





Exercise 9 *(Optional) Explain why this epigraph-based definition is equivalent to the secant-based definition above in* (1)*. You don't have to give a formal proof.*

Tip. To show these definitions are equivalent, show that the convexity of a function's epigraph implies the convexity of the function and that the convexity of a function implies convexity of its epigraph. The latter part is a little harder. For intuition, try drawing a segment in a convex function's epigraph and the secant below it.

Solution 9 *Let's start with how convexity of a function's epigraph implies convexity of the function. Because points on a curve are in its epigraph, a secant is the line segment between two points in the epigraph; convexity of the epigraph implies this segment is in the epigraph, i.e., above the curve itself.*

Now let's talk about why convexity of a function implies convexity of its epigraph. We'll rely on the fact that if we have two segments with matching x coordinates, and the y coordinates of one segment's endpoints lie above those of the other, then that segment lies above the other in its entirety, i.e.,

$$(1-\lambda)y_a + \lambda y_b \geq (1-\lambda)y'_a + \lambda y'_b \quad \text{if} \quad y_a \geq y'_a \;\; \text{and} \;\; y_b \geq y'_b.$$

Because any two points (a, y_a) and (b, y_b) in the function's epigraph lie (by definition) above the endpoints of the secant $(a, f(a))$ and $(b, f(b))$, and the secant lies above the curve and the segment between our first two points above the secant, it follows that the segment between the first two points lies above the curve, i.e., in the epigraph.

2 Convex Regression

Now that we've developed some intuition for what a convex function is, let's implement convex regression. That is, let's solve

$$\hat{\mu} = \underset{\text{convex } m:\mathbb{R}\to\mathbb{R}}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n}\{Y_i - m(X_i)\}^2. \tag{5}$$

For this, we'll follow the same steps we used in the Monotone Regression Lab. We'll first solve a version of this problem for convex functions on the sample $\mathcal{X} = \{X_1 \dots X_n\}$, then extend our solution to the real line.

3 Fitting the Convex Regression Model

What does it mean for a function on $\mathcal{X}$ to be convex? We'll start with the same 'secant' definition we use for functions on $\mathbb{R}$, then forget about points that aren't in $\mathcal{X}$. That is, we'll say that $m : \mathcal{X} \to \mathbb{R}$ is convex if and only if

$$\lambda m(a) + (1-\lambda)m(b) \leq m\{(1-\lambda)a + \lambda b\}$$

whenever a, b, and $x_\lambda = (1-\lambda)a + \lambda b$ with $\lambda \in [0,1]$ are all in $\mathcal{X}$.

And we'll solve the restricted problem.

$$\hat{\mu}_{|\mathcal{X}} = \underset{\text{convex } m:\mathcal{X}\to\mathbb{R}}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n}\{Y_i - m(X_i)\}^2. \tag{6}$$

Exercise 10 *Rewrite the optimization problem* (6) *more concretely in terms the values of* $X_1 \dots X_n$ *and* $m(X_1) \dots m(X_n)$.

Then translate it into a a constrained optimization over a vector $\vec{m}$*, so that, once you've solved for the optimal vector* $\vec{\mu}$*, you can express* $\hat{\mu}_{|\mathcal{X}}(X_1) \dots \hat{\mu}_{|\mathcal{X}}(X_n)$ *in terms of the elements of* $\vec{\mu}$ *(e.g. you might use* $\hat{\mu}_{|\mathcal{X}}(X_i) = \vec{\mu}_i$ *if* $X_1 \dots X_n$ *are distinct.) Try to do it so what you've written translates straightforwardly into CVXR code.*

Tips.

1. You should have a constraint for all triples (i, j, k) for which $X_i < X_j < X_k$. There is a smaller set of constraints that implies all of these, and we'll get there in Exercise 21, but it'll take some work. For now use the full set, like we did until the section 'Optional Exercise: Optimization' in the monotone regression lab.

2. If you want to keep things simple, go ahead and assume that $X_1 \dots X_n$ take on n distinct values, just like we did at the beginning of the monotone regression lab. If you want more generally applicable code, take a look at how we use `invert.unique` in the monotone regression lab to handle duplicate values.

Solution 10 *We'll get a constraint for all triples* (i, j, k) *for which* $X_i < X_j < X_k$*. This constraint arises from* (1) *with* $a = X_i$*,* $b = X_k$*, and* $\lambda = \frac{X_j - X_i}{X_k - X_i}$ *chosen so that* $X_j = (1 - \lambda)X_i + \lambda X_k$*. It's*

$$m(X_j) \leq \left(1 - \frac{X_j - X_i}{X_k - X_i}\right) m(X_i) + \frac{X_j - X_i}{X_k - X_i} m(X_k)$$

which we can rearrange to get the equivalent statement that the slope of the first secant is less than that of the second, i.e.,

$$\frac{m(X_j) - m(X_i)}{X_j - X_i} \leq \frac{m(X_k) - m(X_i)}{X_k - X_i}.$$

If we want an equivalent form without division, we can multiply by the product of these denominators, yielding

$$\{m(X_j) - m(X_i)\}(X_k - X_i) \leq \{m(X_k) - m(X_i)\}(X_j - X_i).$$

Exercise 11 *Implement that optimization in* **R**. *That is, write an* **R** *function* **convexreg** *analogous to* **monotonereg** *from the monotone regression lab that solves* (6)*. Then, from the eight distributions described below, sample* $n = 25$ *observations* $(X_1, Y_1) \dots (X_n, Y_n)$ *and use your code to calculate predictions* $\hat{\mu}(X_1) \dots \hat{\mu}(X_1)$ *based on the solution to* (6)*. Each time, plot your predictions on top of the data, i.e., make a single scatter plot showing both your predictions* $(X_i, \hat{\mu}(X_i))$ *and your observations* (X_i, Y_i)*. Turn in those eight plots, labeling each with the signal used, as your solution to this exercise.*

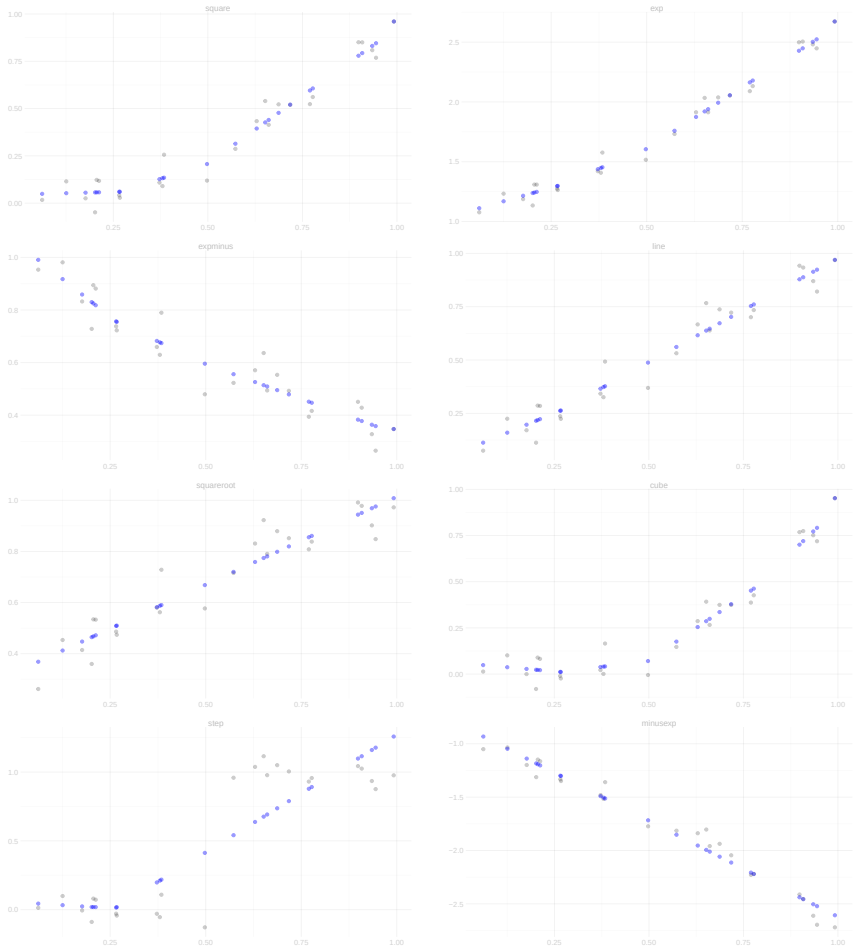
We'll use as our signals μ the eight examples of convex and non-convex functions in Section 1. For each, we'll work with independent and identically distributed observations $(X_1, Y_1) \dots (X_n, Y_n)$ where X_i is drawn from the uniform distribution on $[0, 1]$ and $Y_i = \mu(X_i) + \varepsilon_i$ for ε_i drawn from the normal distribution with mean zero and standard deviation $\sigma = 1/10$.

Tip. CVXR seems to be having some trouble with this one if we use division in our constraint, so don't. To write your constraint without division, observe that the following set of constraints are equivalent: (i) $a/b \leq a'/b'$ and (ii) $ab' \leq a'b$.

Solution 11

```
convexreg = function(X,Y) {
  input = list(X=X, Y=Y)
  n = length(X)
  m = Variable(n)
  mse = sum((Y - m)^2) / n

  grid = expand.grid(i=1:n, j=1:n, k=1:n)
  grid_ordered = grid[X[grid$i] <= X[grid$j] & X[grid$j] <= X[grid$k], ]
  ii = grid_ordered$i
  jj = grid_ordered$j
  kk = grid_ordered$k
  convex.constraint = list((m[jj] - m[ii]) * (X[kk] - X[ii]) <=
                           (m[kk] - m[ii]) * (X[jj] - X[ii]))

  # solve and ask for m that solves our minimization problem
  solved = solve(Problem(Minimize(mse), convex.constraint))
  mu.hat = solved$getValue(m)

  # now a little boilerplate to make it idiomatic R
  #  1. we record the input X and the solution mu.hat in a list
  #  2. we assign that list a class, so R knows predict should
  #     delegate to predict.convexreg
  #  3. we return the list
  model = list(X=X, mu.hat=mu.hat, input=input)
  attr(model, "class") = "convexreg"
  model
}

# save this for comparison to an optimized implementation below that we'll also call convexr
convexreg.slow = convexreg
```

```
mus = list(square = function(x) { x^2 },
           exp = function(x) { exp(x) },
           expminus = function(x) { exp(-x) },
           line = function(x) { x },
           squareroot = function(x) { sqrt(x) },
           cube = function(x) { x^3 },
           step = function(x) { 1*(x >= .5) },
           minusexp = function(x) { -exp(x) })

make.plot = function(mu,fit=convexreg, seed=1, n=25, line=FALSE, points=TRUE) {
     set.seed(seed)
     sigma = .1
     X = runif(n)
     Y = mu(X) + sigma*rnorm(n)

     model = fit(X,Y)

     p = ggplot() + geom_point(aes(x=X,y=Y), alpha=.2, data=data.frame(X=X,Y=Y))
     if(points) {
          p = p + geom_point(aes(x=X, y=mu.hat), alpha=.4, color='blue',
                     data=data.frame(X=model$X, mu.hat=model$mu.hat))
     }
     if(line) {
          x = seq(.001,1,by=.001)
          line.data = data.frame(x=x,
                                 mu=mu(x),
                                 mu.hat=predict(model, newdata=data.frame(X=x)))
          p=p + geom_line(aes(x=x, y=mu), data=line.data) +
                geom_line(aes(x=x, y=mu.hat), color='blue', data=line.data)
     }
     p
}
```



Exercise 12 *Revisit the curves $\hat{\mu}$ you fit in the last exercise. For each, answer these questions.*

1. *Does it fit the data?*
2. *If not, what other model we've talked about could we do to fit the data better?*

Then, if there is a better model, use it and include the resulting plot.

Solution 12 *The ones that fit well are the convex ones—the square, exp, expminus, and line—and the cube. The cube works because, while it's not convex as a function on $\mathbb{R}$, it is convex as a function on $[0, 1]$, and we're using a distribution for which $X_1 \dots X_n \in [0, 1]$.*[2]

The remaining ones—the square root, step, and minusexp—are all monotone with total variation one, so we'd be better off using monotone or bounded variation regression. On the left, I show the results of monotone regression, and on the right, I show the results of bounded variation regression with budget $B = 1$.

3.1 Filling in the gaps

At this point, you have an estimator $\hat{\mu}_{|\mathcal{X}}$ that minimizes squared error among the convex functions $m : \mathcal{X} \to \mathbb{R}$. This lets us plot some isolated points. But we want a convex curve $\hat{\mu}(x)$ for $x \in [0, 1]$ and we want it to be the best-fitting such curve, i.e., we want the solution to (5).

To do this, we'll use a *piecewise-linear extension* of $\hat{\mu}_{|\mathcal{X}}$. That is, having sorted X_i into increasing order, we will define $\hat{\mu}(x)$ everywhere on $[X_1, X_n]$ by drawing line segments between successive points $\{X_i, \hat{\mu}(X_i)\}$ and $\{X_{i+1}, \hat{\mu}(X_{i+1})\}$, and extend the leftmost and rightmost segment to fill the intervals $[0, X_1]$ and $[X_n, 1]$.[3] This gives us a piecewise-linear solution to (6). First, we'll implement it. Then we'll verify that it is, in fact, a solution to (6).

Exercise 13 *Briefly explain why piecewise-constant extension would not give us a solution to* (6)*. A sentence or a sketch should do.*

Tip. Think about the examples from Section 1.

Solution 13 *Piecewise-constant functions have steps and steps aren't convex.*

3.1.1 Implementation

Exercise 14 *Write out a formula for the piecewise-linear curve $\hat{\mu}(x)$ in terms of $\hat{\mu}(X_1) \dots \hat{\mu}(X_n)$. Then implement it and add the curve $\hat{\mu}(x)$ for $x \in [0, 1]$ to*

your plots from the Exercise 11.

Tip. For coding a piecewise linear function, try to modify the function `predict.piecewise.constant` from the bounded variation lab.

Solution 14

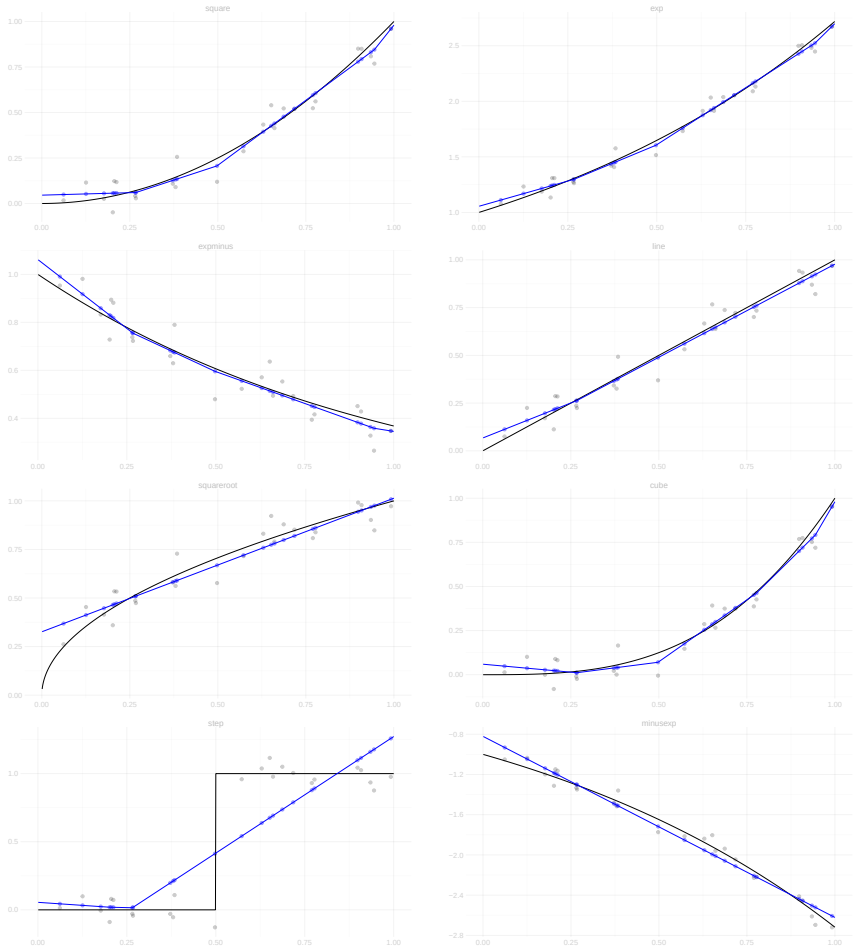
$$\hat{\mu}(x) = \hat{\mu}(X_i) + \frac{\hat{\mu}(X_{i+1}) - \hat{\mu}(X_i)}{X_{i+1} - X_i}(x - X_i) \quad \text{for } i = \max\{i \in 1 \dots n-1 : X_i \leq x\} \cup \{1\}. \tag{7}$$

Here's a little explanation. The formula for $\hat{\mu}(x)$ is the formula for the line through $\{X_i, \hat{\mu}(X_i)\}$ and $\{X_{i+1}, \hat{\mu}(X_i)\}$ where X_i for $i \in 1 \dots n-1$ is the largest X_i to the left of our query point x or, if there are none, X_1. This last caveat handles the case that x is to the left of X_1; to handle the case that it's to the right of X_n, we've restricted the range of our search to $1 \dots n-1$ because if we are to the right of X_n, we still want to use X_{n-1} as our segment's starting point.

Here's the implementation.

```
predict.piecewise.linear = function(model, newdata=data.frame(X=model$input$X)) {
  Y = model$mu.hat; X=model$X; x=newdata$X; n = length(X)
  # Sort the X/muhat pairs so that X[1] <= X[2] <= ... <= X[n]
  # This is usually handled by the fitting function, but to keep my slow implementation simp
  o = order(X)
  X = X[o]; Y = Y[o]

  # for each new data point x[k]
  # find the closest observed X[i[k]] left of x[k]
  # i.e., i[k] is the largest integer i for which X[i] <= x[k]
  i = findInterval(newdata$X, X)
  # If there is no X[i] < x[k], findInterval sets i[k]=0
  #  and we'll want to act as if we'd gotten 1 so we use the
  #  line through (X[1], Y[1])  and (X[2], Y[2])
  # If that k is n, we'll want to act as if we'd gotten n-1 so we use
  #  the line through (X[n-1], Y[n-1])  and (X[n], Y[n])
  i[i==0] = 1; i[i==n] = n-1
  # make a prediction using the formula y - y0 = (x-x0) * slope
  Y[i] + (x-X[i]) * (Y[i+1]-Y[i])/(X[i+1]-X[i])
}

predict.convexreg = predict.piecewise.linear
```

Here are the plots. The gray dots are observations (X_i, Y_i), the black line is $\mu(x)$, the blue dots are points $\{X_i, \hat{\mu}(X_i)\}$ on the fitted curves, and the blue line is $\hat{\mu}(x)$.



3.1.2 Verification

Exercise 15 *Consider any pair $x < x'$. Prove that for any piecewise-linear function m with breaks at $X_1 \dots X_n$, the secant slope $\{m(x') - m(x)\}/(x' - x)$ between these points is a* weighted average *of the slopes $\{m(X_{j+1}) - m(X_j)\}/(X_{j+1} - X_j)$ of the segments that lie between them. Briefly explain why this implies that our piecewise-linear extension of the solution to (6), $\hat{\mu} : \mathbb{R} \to \mathbb{R}$, solves the convex regression problem (5).*

Tips.

1. Break the 'explain' part of this down into feasibility and optimality, like we did in the bounded variation regression lab.

2. You can get an inequality equivalent to (1) by subtracting $m(a)$ from both sides of (1) and dividing the result by $\lambda(b - a)$. What does this have to

do with secant slopes?

Solution 15 ***The weighted average characterization.*** *What we're going to do is break down the secant slope we're interested in as a weighted average of segment slopes. Let $X_1 \dots X_n$ be sorted in increasing order and i and i' be chosen as in (7) for $x = x$ and $x = x'$ respectively, so x and x' are between X_i and X_{i+1} and $X_{i'}$ and $X_{i'+1}$ respectively.*[4] *We'll expand the secant slope's numerator, $m(x') - m(x)$, by adding zero written in a fancy way, using a telescoping sum: $0 = -m(X_{i'}) + \sum_{j=i}^{i'-1} m(X_{j+1}) - m(X_j) + m(X_i)$. What we get is a sum of differences in the piecewise-constant function m between points on the same segment. And we'll rewrite those differences as the product of the segment slope and the distance between the points, i.e., using the identity $m(b) - m(a) = \{b-a\} \times \{m(b) - m(a)\}/\{b-a\}$. That gives us a weighted average of slopes. Take a look.*

$$
\begin{aligned}
\frac{m(x') - m(x)}{x' - x} &= \frac{\{m(x') - m(X_{i'})\} + \left\{\sum_{j=i}^{i'-1} m(X_{j+1}) - m(X_j)\right\} - \{m(x) - m(X_i)\}}{x' - x} \\
&= \frac{m(x') - m(X_{i'})}{x' - X_{i'}} \left\{ \frac{x' - X_{i'}}{x' - x} \right\} \\
&+ \sum_{j=i}^{i'-1} \frac{m(X_{j+1}) - m(X_j)}{X_{j+1} - X_j} \left\{ \frac{X_{j+1} - X_j}{x' - x} \right\} \\
&+ \frac{x - m(X_i)}{x - X_i} \left\{ \frac{x - X_i}{x' - x} \right\} \\
&= \frac{m(X_{i'+1}) - m(X_{i'})}{X_{i'+1} - X_{i'}} \left\{ \frac{x' - X_{i'}}{x' - x} \right\} \\
&+ \sum_{j=i}^{i'-1} \frac{m(X_{j+1}) - m(X_j)}{X_{j+1} - X_j} \left\{ \frac{X_{j+1} - X_j}{x' - x} \right\} \\
&+ \frac{m(X_{i+1}) - m(X_i)}{X_{i+1} - X_i} \left\{ \frac{x - X_i}{x' - x} \right\}
\end{aligned}
$$

What's going on in the last equality? The red and blue slopes are the same in each expression. Because $\hat{\mu}$ is piecewise-linear, the slope of $\hat{\mu}$ between X_i and x is the same as the slope of $\hat{\mu}$ between X_i and X_{i+1}.

Now observe that the ratios in curly braces that multiply these slopes are non-negative weights that sum to one, so our secant slope is a weighted average of these segment slopes.

Feasibility. *What does this have to do with the convexity of piecewise-linear extensions of the solution to (6)? Let's reformulate our definition of convexity in terms of secant slopes. To do that, we'll start with the definition (1), subtract $m(a)$ from both sides, and divide the result by the 'run' $\{(1-\lambda)a + \lambda b\} - a =$*

$\lambda\{b-a\}$ *of the secant on the right side.*

$$\begin{aligned}
&(1-\lambda)m(a)+\lambda m(b) \geq m\{(1-\lambda)a+\lambda b\} && \iff\\
&\lambda\{m(b)-m(a)\} \geq m\{(1-\lambda)a+\lambda b\}-m(a) && \iff\\
&\frac{m(b)-m(a)}{b-a} \geq \frac{m\{(1-\lambda)a+\lambda b\}-m(a)}{\{(1-\lambda)a+\lambda b\}-a}
\end{aligned}$$

The result is an equivalent inequality that says that a curve is convex if and only ifthe slope of the secant from a *to* b *is at least as large as the slope of the secant from* a *to* $x_\lambda = \{(1-\lambda)a+\lambda b\} \in [a,b]$.

Here's where the weighted average characterization comes in. Let's compare the secant slope from a *to* b *with the secant slope from* a *to* x_λ.

$$\begin{aligned}
\frac{m(b)-m(a)}{b-a} &= \frac{m(X_{i'+1})-m(X_{i'})}{X_{i'+1}-X_{i'}}\left\{\frac{b-X_{i'}}{b-a}\right\}\\
&\quad+\sum_{j=i}^{i'-1}\frac{m(X_{j+1})-m(X_j)}{X_{j+1}-X_j}\left\{\frac{X_{j+1}-X_j}{b-a}\right\}\\
&\quad+\frac{m(X_{i+1})-m(X_i)}{X_{i+1}-X_i}\left\{\frac{a-X_i}{b-a}\right\}\\
\frac{m(x_\lambda)-m(a)}{x_\lambda-a} &= \frac{m(X_{i''+1})-m(X_{i''})}{X_{i''+1}-X_{i''}}\left\{\frac{x_\lambda-X_{i''}}{x_\lambda-a}\right\}\\
&\quad+\sum_{j=i}^{i''-1}\frac{m(X_{j+1})-m(X_j)}{X_{j+1}-X_j}\left\{\frac{X_{j+1}-X_j}{x_\lambda-a}\right\}\\
&\quad+\frac{m(X_{i+1})-m(X_i)}{X_{i+1}-X_i}\left\{\frac{a-X_i}{x_\lambda-a}\right\}
\end{aligned}$$

For each term except the first in the $a \to x_\lambda$ *secant slope, we have a corresponding term in the* a *to* b *secant slope with the same slope but a* smaller weight. *The rest of the weight for the* $a \to b$ *secant slope is on segments that are either that last segment on the* a *to* x_λ *secant slope or segments to the right of it, i.e. segments where, if* $m_{|\mathcal{X}}$ *is convex, the slope is at least as large as on any segment between* a *and* x_λ. *So for each term in the* $a \to x_\lambda$ *secant slope's expansion, we can think of the* $a \to b$ *secant slope as splitting the same weight between (i) the same segment (ii) a segment of slope at least as large [one between* x_λ *and* b*]. It follows that the* $a \to b$ *secant slope is at least as large as the* $a \to x_\lambda$ *secant slope, i.e., that the piecewise-linear extension of a convex function on* $\mathcal{X}$ *is convex on* $\mathbb{R}$. *This tells us* $\hat{\mu}$ *is feasible.*

Optimality. *We argue by contradiction. If we had any convex function* $m : \mathbb{R} \to \mathbb{R}$ *that had a smaller mean squared error than* $\hat{\mu}$*, then its restriction to the sample,* $m_{|\mathcal{X}}$*, would be convex have a smaller mean squared error than* $\hat{\mu}_{|\mathcal{X}}$*, so* $\hat{\mu}_{|\mathcal{X}}$ *could not be a solution to* (6).

3.2 Optimized Fitting

The downside of all this, from an implementation perspective, is that it involve *a lot* of constraints. The number of constraints is proportional to n^3. We can fix this. Ultimately, what we'll do is a lot like what we did to speed up monotone regression: we found a set of *local constraints* that determined whether a function was monotone. In particular, we found a way of establishing monotonicity by looking at pairs of neighboring observations instead of all pairs of observations.

3.3 Thinking locally about convexity

Let's think about whether we can use a *local properties* to determine whether a function is convex. By local property, I mean something you can check by looking only at small pieces of the function rather than the whole function all at once. For example, we know a function is increasing everywhere if it's increasing between n and $n+1$ for all integers n. This works more generally, if in place of the intervals $[n, n+1]$ we use any set of intervals that combine to cover the whole real line. And because a differentiable function is convex if and only if it has an increasing derivative, it follows that we can use this approach to determine whether a differentiable function is convex.

Let's try to generalize this. To start, it's worth observing that using exactly this approach won't work.

Exercise 16 *Describe a non-convex curve that is convex on the intervals* $[n, n+1]$ *for all integers* n*. Here, by convex on an interval, I mean that* (1) *holds for all points* a, b *in that interval.*

Tip. Look at the examples of non-convex curves above.

Solution 16

$$\textit{A step function.} \quad m(x) = \begin{cases} 0 & \textit{if } x < 0 \\ 1 & \textit{otherwise} \end{cases}$$

We can fix this by looking at *overlapping intervals* that cover the real line, for example, the intervals $[n-1, n+1]$. By overlapping, I mean that the endpoints of each interval are in the interior of (i.e. in but not endpoints of) some other interval. Our ultimate goal will be to show that a function is convex if it's convex on overlapping intervals that cover the real line. But to get the concepts down without messy arithmetic, let's start with something easier.

Exercise 17 *Show that if* $f(1) \leq \frac{1}{2}f(0) + \frac{1}{2}f(2)$ *and* $f(2) \leq \frac{1}{2}f(1) + \frac{1}{2}f(3)$*, then* $f(1) \leq \frac{2}{3}f(0) + \frac{1}{3}f(3)$*. Continue with this approach to show that* $f(1) \leq \frac{3}{4}f(0) + \frac{1}{4}f(4)$ *if, in addition,* $f(3) \leq \frac{1}{2}f(2) + \frac{1}{2}f(4)$*.*

Solution 17 *Substituting our second bound into our first gives us an inequality involving only $f(0)$, $f(1)$, and $f(3)$.*

$$\begin{aligned} f(1) &\leq \frac{1}{2}f(0) + \frac{1}{2}f(2) \\ &\leq \frac{1}{2}f(0) + \frac{1}{2}\left\{\frac{1}{2}f(1) + \frac{1}{2}f(3)\right\} \\ &= \frac{1}{2}f(0) + \frac{1}{4}f(1) + \frac{1}{4}f(3). \end{aligned}$$

Subtracting the right side's term $\frac{1}{4}f(1)$ from both sides, we get the bound $\frac{3}{4}f(1) \leq \frac{1}{2}f(0) + \frac{1}{4}f(3)$, and dividing both sides by $3/4$ gives the claimed bound on $f(1)$ in terms of $f(0)$ and $f(3)$. If $f(3) \leq \frac{1}{2}f(2) + \frac{1}{2}f(4)$, we can derive the bound $f(3) \leq \frac{1}{3}f(1) + \frac{2}{3}f(4)$ analogously from here:

$$f(3) \leq \frac{1}{2}f(2) + \frac{1}{2}f(4) \leq \frac{1}{2}\left\{\frac{1}{2}f(1) + \frac{1}{2}f(3)\right\} + \frac{1}{2}f(4).$$

Now let's derive the claimed bound in terms of $f(0)$ and $f(4)$. The argument is similar, combining these last two bounds.

$$\begin{aligned} f(1) &\leq \frac{2}{3}f(0) + \frac{1}{3}f(3) \\ &\leq \frac{2}{3}f(0) + \frac{1}{3}\left\{\frac{1}{3}f(1) + \frac{2}{3}f(4)\right\} \\ &= \frac{2}{3}f(0) + \frac{1}{9}f(1) + \frac{2}{9}f(4). \end{aligned}$$

Rearranging, we get $f(1) \leq \frac{9}{8}\{\frac{2}{3}f(0) + \frac{2}{9}f(4)\} = \frac{3}{4}f(0) + \frac{1}{4}f(4)$ as claimed.

It looks like there's a pattern here. If $f(n+1) \leq f(n) + f(n+2)$ for positive integers n, then $f(1) \leq \frac{n-1}{n}f(0) + \frac{1}{n}f(n)$. And because $1 = \frac{n-1}{n} \cdot 0 + \frac{1}{n} \cdot n$, this is an instance of our convexity-defining inequality (1) for $a = 0$ and $b = n$. If you're familiar with proof by induction, try the next exercise.

Exercise 18 *(Optional) Prove it! Use induction on n.*

Solution 18 *Suppose that, for $n = N$, we have the following bounds.*

$$\star \quad f(1) \leq \frac{n-1}{n}f(0) + \frac{1}{n}f(n) \quad \text{and} \quad f(n) \leq \frac{1}{n}f(1) + \frac{n-1}{n}f(n+1).$$

This is the case when $N = 1$. Then proceeding as in the last exercise,

$$f(1) \leq \frac{n-1}{n}f(0) + \frac{1}{n}f(n) \leq \frac{n-1}{n}f(0) + \frac{1}{n}\left\{\frac{1}{n}f(1) + \frac{n-1}{n}f(n+1)\right\}.$$

By rearranging terms, we get

$$\frac{n^2-1}{n^2}f(1) \leq \frac{n-1}{n}f(0) + \frac{n-1}{n^2}f(n+1)$$

and then by dividing by $\frac{n^2-1}{n^2} = \frac{(n+1)(n-1)}{n^2}$*, we get*

$$f(1) \leq \frac{n(n-1)}{n^2-1}f(0) + \frac{n-1}{n^2-1}f(n+1) = \frac{n}{n+1}f(0) + \frac{1}{n+1}f(n+1).$$

We can derive the bound $f(n) \leq \frac{1}{n+1}f(1) + \frac{n}{n+1}f(n+2)$ *analogously from here.*

$$f(n) \leq \frac{1}{n}f(1) + \frac{n-1}{n}f(n+1) \leq \frac{1}{n}\left\{\frac{n-1}{n}f(0) + \frac{1}{n}f(n)\right\} + \frac{n-1}{n}f(n+1).$$

Thus, we have the bounds $\star$ *for* $n = N+1$ *as well. It follows, by induction, that we have them for all* $n \geq 2$*.*

The general case. If, for some increasing sequence $x_1 < x_2 < x_3 < \ldots < x_n$, a function f is convex on the overlapping intervals $[x_1, x_3]$, $[x_2, x_4]$, ..., $[x_{n-2}, x_n]$, then it's convex on the interval $[x_1, x_n]$. This is what we'll want when we're implementing our faster version of convex regression.

Exercise 19 *(Optional) Prove it!*

Tip. Start by showing that if f is convex on two intervals $[a, b]$ and $[b, c]$ and satisfies $f(b) \leq (1-\lambda')f(a) + \lambda' f(c)$ for the value of $\lambda' \in [0, 1]$ for which $b = (1-\lambda')a + \lambda' c$, then f is convex on $[a, c]$. To do this, it helps to observe that we can write $x \in [a, b]$ as $(1-\lambda)a + \lambda b = (1-\lambda)a + \lambda\{(1-\lambda')a + \lambda' c\} = (1-\lambda\lambda')a + \lambda\lambda' c$ and do something analogous for $x \in [b, c]$.

Tip. At some point in your argument, you'll probably want to take $a = x_1$, $b = x_3$, and $c = x_4$. To show that $f(x_3) \leq (1-\lambda')f(x_1) + \lambda' f(x_4)$ for λ' such that $x_3 = (1-\lambda')x_1 + \lambda' x_4$, you'll want to use the properties that $f(x_3) \leq (1-\lambda'')f(x_2) + \lambda'' f(x_4)$ for λ'' such that $x_3 = (1-\lambda'')x_2 + \lambda'' x_4$ and $f(x_2) \leq (1-\lambda''')f(x_1) + \lambda''' f(x_3)$ for λ''' such that $x_2 = (1-\lambda''')x_1 + \lambda''' x_3$.

Tip. The basic idea here is the same as the last exercise, but it's a bit messy. At least the way I did it. If you do want to try it, I recommend that you skip it on you first pass and come back to it when you've worked through the others.

Solution 19
First tip. *Our premise is that* f *is convex on two intervals* $[a, b]$ *and* $[b, c]$ *and satisfies* $f(b) \leq (1-\lambda')f(a) + \lambda' f(c)$ *for the value of* $\lambda' \in [0, 1]$ *for which* $b = (1-\lambda')a + \lambda' c$*. We'll show this implies* f *is convex on* $[a, c]$*.*

Any point in this interval is in at least one of the intervals $[a, b]$ *or* $[b, c]$*, so we'll deal with each case. If* $x \in [a, b]$*, we can write* $x = (1-\lambda)a + \lambda b$ *for* $\lambda \in [0, 1]$*. It follows, given our definition of* λ'*, that*

$$x = (1-\lambda)a + \lambda\{(1-\lambda')a + \lambda' c\} = (1-\lambda\lambda')a + \lambda\lambda' c.$$

And it follows from this and our premise that

$$\begin{aligned} f\{(1-\lambda\lambda')a+\lambda\lambda' c\} &= f\{(1-\lambda)a+\lambda b\} \\ &\leq (1-\lambda)f(a)+\lambda f(b) \\ &\leq (1-\lambda)f(a)+\lambda\{(1-\lambda')f(a)+\lambda' f(c)\} \\ &= (1-\lambda\lambda')f(a)+\lambda\lambda' f(c). \end{aligned}$$

This holds for all λ from 0 to 1 and therefore for all $\lambda'' = \lambda\lambda' \in [0, \lambda']$.

If instead $x \in [b, c]$, we can write $x = (1-\lambda)b + \lambda c$ for $\lambda \in [0, 1]$. It follows, given our definition of λ', that

$$x = (1-\lambda)\{(1-\lambda')a+\lambda' c\} + \lambda c = (1-\lambda-\lambda'+\lambda\lambda')a + (\lambda+\lambda'-\lambda\lambda')c.$$

And it follows from this and our premise that

$$\begin{aligned} f\{(1-\lambda-\lambda'+\lambda\lambda')a+(\lambda+\lambda'-\lambda\lambda')c\} &= f\{(1-\lambda)b+\lambda c\} \\ &\leq (1-\lambda)f(b)+\lambda f(c) \\ &\leq (1-\lambda)\{(1-\lambda')f(a)+\lambda' f(c)\}+\lambda f(c) \\ &= (1-\lambda-\lambda'+\lambda\lambda')f(a)+(\lambda+\lambda'-\lambda\lambda')f(c). \end{aligned}$$

This holds for all λ from 0 to 1 and therefore for all $\lambda'' = \lambda+\lambda'-\lambda\lambda' \in [\lambda', 1]$.

These two results together imply that $f\{(1-\lambda'')a+\lambda'' c\} \leq (1-\lambda'')f(a) + \lambda'' f(c)$ for all $\lambda'' \in [0, 1]$, i.e., that f is convex on $[a, c]$.

Second tip. *Using what we've just shown for $a = x_1$, $b = x_2$, and $c = x_4$, we see that because f is convex on $[x_1, x_2]$ and $[x_2, x_4]$, it's convex on $[x_1, x_4]$ if $f(x_2) \leq (1-\lambda')f(x_1)+\lambda' f(x_4)$ for the value of λ' for which $x_2 = (1-\lambda')x_1+\lambda' x_4$. Let's show that. Because f is convex on $[x_1, x_3]$ and $[x_2, x_4]$, we know that*

$$\begin{aligned} &f(x_3) \leq (1-\lambda'')f(x_2)+\lambda'' f(x_4) \quad \text{for } \lambda'' \text{ such that } x_3 = (1-\lambda'')x_2+\lambda'' x_4 \\ &f(x_2) \leq (1-\lambda''')f(x_1)+\lambda''' f(x_3) \quad \text{for } \lambda''' \text{ such that } x_2 = (1-\lambda''')x_1+\lambda''' x_3. \end{aligned}$$

We can eliminate x_3 from out system of two 'such that' equations to get an expression for x_2 in terms of x_1 and x_4. Solving both (via addition/multiplation) for $\lambda''' x_3$, we find that

$$\lambda'''(1-\lambda'')x_2 + \lambda'''\lambda'' x_4 = \lambda''' x_3 = x_2 + (\lambda'''-1)x_1$$

and therefore

$$x_2 = \frac{1-\lambda'''}{1-\lambda'''(1-\lambda'')}x_1 + \frac{\lambda''}{1-\lambda'''(1-\lambda'')}x_4 = (1-\lambda')x_1+\lambda' x_4 \quad \text{for} \quad \lambda' = \frac{\lambda''}{1-\lambda'''(1-\lambda'')}.$$

Furthermore, as a consequence of convexity on $[x_1, x_3]$ and $[x_2, x_4]$,

$$f(x_2) \leq (1-\lambda''')f(x_1)+\lambda''' f(x_3) \leq (1-\lambda''')f(x_1)+\lambda'''\{(1-\lambda'')f(x_2)+\lambda'' f(x_4)\}$$

and therefore, solving for $f(x_2)$, that

$$f(x_2) \leq \frac{1-\lambda'''}{1-\lambda'''(1-\lambda'')} f(x_1) + \frac{\lambda''}{1-\lambda'''(1-\lambda'')} = (1-\lambda')f(x_1) + \lambda' f(x_4).$$

This is what was required to show convexity on $[x_1, x_4]$.

Now we know that f is convex on each of the overlapping intervals $[x_1, x_4]$, $[x_3, x_5]$, …. Let's rename them. Letting $\tilde{x}_1 = x_1$ and $\tilde{x}_i = x_{i+1}$ for $i \in 2 \dots n-1$, what we know is that f is convex on each of the intervals $[\tilde{x}_1, \tilde{x}_3], [\tilde{x}_2, \tilde{x}_4], \dots$. And it follows that we can apply the argument above to this set of intervals to show that f is convex on each of the intervals $[\tilde{x}_1, \tilde{x}_4], [\tilde{x}_3, \tilde{x}_5], \dots$, or using our original names, on each of the intervals $[x_1, x_5], [x_4, x_6], \dots$. We can repeat this, each time merging our first two intervals, until we ultimately get the result that f is convex on the single interval $[x_1, x_n]$.

3.4 Implementation

If our observations $X_1 \dots X_n$ are distinct and sorted in increasing order, then $I_1 = [-\infty, X_2], I_2 = [X_1, X_3], I_3 = [X_2, X_4], \dots, I_{n-1} = [X_{n-2}, X_n], I_n = [X_{n-1}, \infty]$ are overlapping intervals that cover the real line. A function $m : \mathbb{R} \to \mathbb{R}$ is convex if and only if it's convex on all of these intervals, i.e. if the restriction $m_{|I_j}$ is convex for all intervals I_j. And a function $m_{|\mathcal{X}} : \mathcal{X} \to \mathbb{R}$ is convex if and only the restriction to *the observations X_i in each interval*, i.e. the intersection $\mathcal{X}_j = \mathcal{X} \cap I_j$, is convex for all intervals I_j. What's cool about this is that there are either 2 or 3 observations in each of these sets $\mathcal{X}_j$[5], so — convexity on a set $\mathcal{X}_j$ being a property involving all *triples* in $\mathcal{X}_j$ — we get either 0 or 1 for each of these sets $\mathcal{X}_1 \dots \mathcal{X}_n$. That means that, in total, we get no more than n constraints—in fact, we'll get $n-2$.

Exercise 20 *Rewrite the optimization problem* (6) *more concretely in terms the values of $X_1 \dots X_n$ and $m(X_1) \dots m(X_n)$, this time using the $n-2$ 'convexity on $\mathcal{X}_j$' constraints rather than the $\approx n^3$ constraints we used in Exercise 10.*

Then translate it into a a constrained optimization over a vector $\vec{m}$, so that, once you've solved for the optimal vector $\vec{\mu}$, you can express $\hat{\mu}_{|\mathcal{X}}(X_1) \dots \hat{\mu}_{|\mathcal{X}}(X_n)$ in terms of the elements of $\vec{\mu}$ (e.g. you might use $\hat{\mu}_{|\mathcal{X}}(X_i) = \vec{\mu}_i$ if $X_1 \dots X_n$ are distinct.) Try to do it so what you've written translates straightforwardly into CVXR code.

Tip. If you want to keep things simple, go ahead and assume that $X_1 \dots X_n$ take on n distinct values, just like we did at the beginning of the monotone regression lab. If you want more generally applicable code, take a look at how we use `invert.unique` in the monotone regression lab to handle duplicate values.

Solution 20 *See constraints in the code below.*

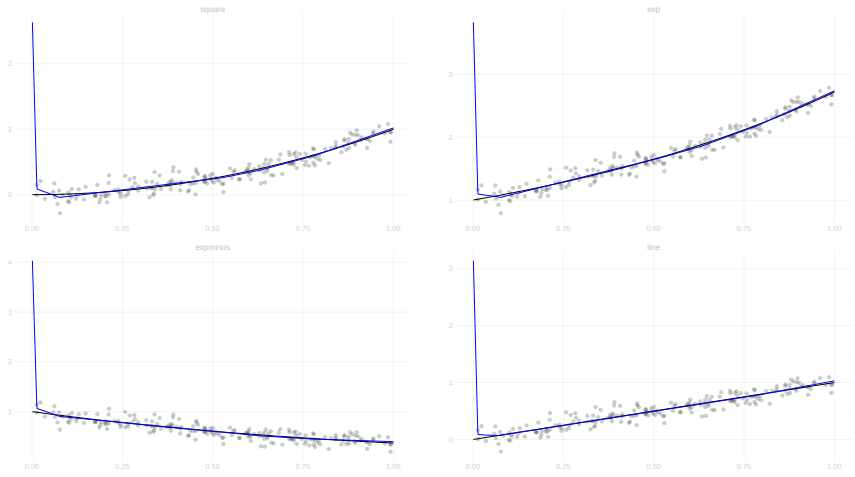
Exercise 21 *Write a new version of* `convexreg` *that uses the constraints from Exercise 20. Then implement it and check that your solution agrees with the one you got using the all-triples constraints in Exercise 11. No need to turn in code, but you'll want this faster implementation for this next part.*

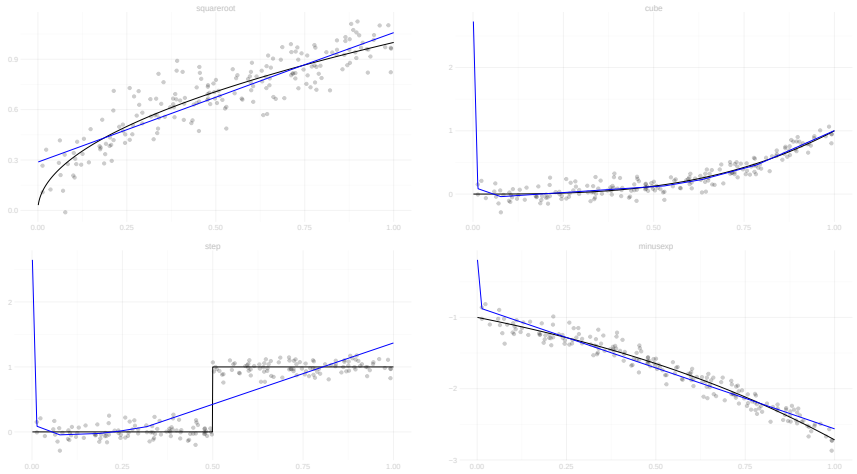
Repeat the fitting-and-plotting exercise from Exercise 11, but using sample size $n = 200$ *instead of* $n = 25$ *and plotting your solution's piecewise-linear extension* $\hat{\mu}$ *as a curve rather than the point predictions* $\hat{\mu}(X_1) \dots \hat{\mu}(X_n)$*. That is, for the eight distributions described below Exercise 11, sample* $n = 200$ *observations* $(X_1, Y_1) \dots (X_n, Y_n)$*, use your new version of* `convexreg` *together with your code from Exercise 14 to solve the convex regression problem* (5)*. Each time, plot the solution* $\hat{\mu}$ *(as a curve) on top of a scatter plot of the observations* (X_i, Y_i)*. Turn in those eight plots, labeling each with the signal used, as your solution to this exercise.*

Solution 21
```
convexreg = function(X, Y, concave = FALSE, monotone = FALSE) {
    # Step 0.
    # We check that the inputs satisfy our assumptions.
    stopifnot(length(X) == length(Y))
    input = list(X=X, Y=Y)
    n = length(X)
    # and find the unique elements of X and the inverse mapping
    unique.X = invert.unique(X)

    # Step 1.
    # We tell CVXR we're thinking about a vector of unknowns m in R^p.
    m = Variable(length(unique.X$elements))
    # and permute and duplicate these into a vector mX with n elements in correspondence w
    mX = m[unique.X$inverse]

    # Step 2.
    # We tell CVXR that we're interested in mean squared error.
    mse = sum((Y - mX)^2 / n)

    # Step 3.
    # We specify our constraints.
    # Interpretation (rearrange): secant slopes are increasing
    uX = X[unique.X$elements]
    ii = 1:(n-2)
    constraints =
      list(((m[ii+1]-m[ii])   * (uX[ii+2]-uX[ii+1]) -
            (m[ii+2]-m[ii+1]) * (uX[ii+1]-uX[ii])) * (-1)^concave <= 0)

    ## If you want to fit monotone convex curves, you can add this constraint.
    if(monotone) {
        decreasing = monotone == 'decreasing'
```

```
            constraints = c(constraints, (-1)^(decreasing) * diff(m) >= 0)
    }
    # Step 4.
    # We ask CVXR to minimize mean squared error subject to our constraints.
    # And we ask for vector mu.hat that does it.
    solved = solve(Problem(Minimize(mse), constraints))
    mu.hat = solved$getValue(m)

    # Step 5: a little boilerplate to make it idiomatic R.
    # 1. we record the unique levels of X and mu.hat, in correspondence and sorted in incre
    # 2. we assign that list a class, so R knows predict should delegate to predict.convex
    # 3. we return the list
    model = list(X = X[unique.X$elements], mu.hat = mu.hat, input = input)
    attr(model, "class") = "convexreg"
    model
}
```

```
X = runif(20)
Y = X + rnorm(20)
model1 = convexreg(X,Y)
model2 = convexreg.slow(X,Y)
max(abs(predict(model1, newdata=data.frame(X=X)) - predict(model2, newdata=data.frame(X=X))

## [1] 5.578616e-06
```





4 Rates of Convergence

Now we've got three nonparametric regression models: monotone curves, bounded variation curves, and convex curves. To keep things simple, we'll be working with data sampled around one signal: $\mu(x) = x$. That is, we'll work with independent and identically distributed observations $(X_1, Y_1) \dots (X_n, Y_n)$ where X_i is drawn from uniform distribution on $[0, 1]$ and $Y_i = \mu(X_i) + \varepsilon_i$ for ε_i drawn independently from the normal distribution with mean zero and standard deviation $\sigma = .5$.

Tip. What we're doing here is taking what we did at the end of the convergence rates lab, simplifying it by using only one signal instead of four, and then adding two new regression models. Use the lab's solution as a starting point.

Exercise 22 *Draw a sample of size $N = 1600$ from this distribution. To get samples of sizes $n = \{25, 50, 100, 200, 400, 800, 1600\}$, use the first 25, 50, etc. observations.*

At all of these sample sizes, fit a line, an increasing curve, a bounded variation curve with budget $B = 1$, and a convex curve. Calculate sample MSE $\|\hat{\mu} - \mu\|^2_{L_2(Pn)}$ and population MSE $\|\hat{\mu} - \mu\|^2_{L_2(P)}$ for each. Repeat this ten times and average the results to get estimates of expected sample MSE and expected population MSE at each sample size n. Include plots of these as a function of n as your solution.

Tip. This can be slow for larger samples. Try it out for samples of size $25 \dots 400$ before adding in $n = 800$ and $n = 1600$.

Solution 22

Let's try to summarize these plots by rates of convergence.

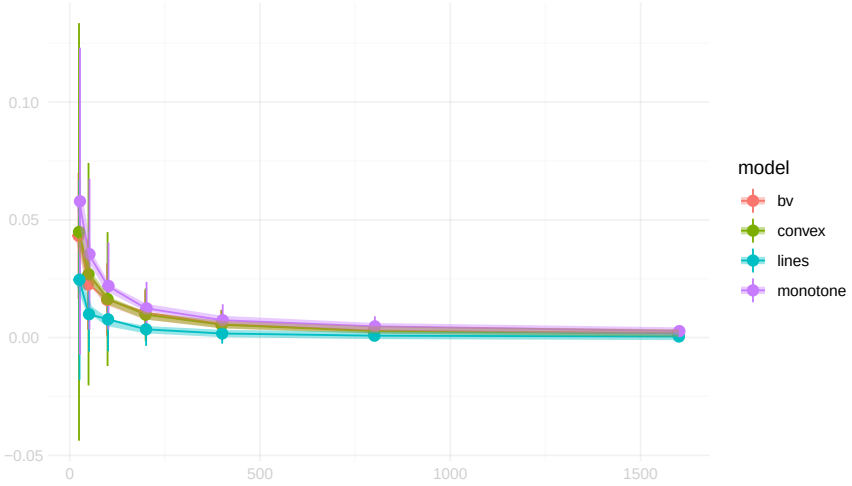


Figure 1: Expected sample MSE (thin lines) and a predictions (thick lines) based on our rate estimates

Exercise 23 *For each of your four regression models, use* `nls` *to fit a curve of the form* $m(n) = \alpha n^{-\beta}$ *to* $RMSE = \sqrt{MSE}$ *where MSE is your estimate of expected population mean squared error from the last exercise. Repeat for expected sample mean squared error.*

Plot the resulting predictions of MSE, $\hat{m}(n)^2$*, on top of your actual MSE curves from the the previous exercise to check their accuracy. Include these plots and report these rates of convergence* $\hat{\beta}$ *as your solution. Briefly comment on what you see, too.*

error.measure	model	a	b
population	bv	0.68	0.37
population	convex	1.35	0.43
population	lines	0.79	0.49
population	monotone	0.79	0.36
sample	bv	0.66	0.36
sample	convex	0.73	0.38
sample	lines	0.69	0.47
sample	monotone	0.78	0.37

Solution 23 *For both sample and population RMSE, the rates I'm estimating are about* $n^{-1/2}$ *for lines,* $n^{-1/3}$ *for monotone and bounded variation, and* $n^{-2/5}$ *for convex. Maybe that's cheating a little, since those are the rates we'll prove later in the semester, but it is there in the table to some extent.*

- *The* $n^{-1/2}$ *rate for lines is something you may have seen in a previous class. If you haven't, you've probably seen it for* horizontal lines, *since the*

least squares prediction $\hat{\mu}(x)$ in that model is the constant $\bar{Y}$ which has standard deviation $\sigma/\sqrt{n}$.

- *Later in the semester, we'll prove that the rates for monotone and bounded variation regression are, in fact,$n^{-1/3}$ or better. If we have time, we'll prove that the rate for convex regression is $n^{-2/5}$ or better, too.*

Our actual error curves do agree well with the predictions we get based on these rates.

At most sample sizes the errors follow the pattern

$$\textit{monotone} > \textit{bounded variation} > \textit{convex} > \textit{lines}.$$

I'd expect some of those comparisons. In a sense, there are fewer lines than monotone curves and fewer monotone curves than convex curves, even though none of these models are contained in the others.

Why fewer convex curves than monotone curves? Think about what the monotonicity constraint tells you: once your curve has hit a certain point, it has to either stay flat or trend up. What the convexity constraint tells you is that once your curve has hit a certain point, it has to keep trending up at least as fast as it has been. The extra flexibility in the convex model relative to the monotone one is that they can trend down before they trend up, but that isn't really enough to make up for the fact that you can't ever flatten out.

Later on in the semester, we'll see some theory that'll tell us what's going on here pretty clearly.