Linear Regression

Fraida Fund

Contents

In this lecture

- Simple (univariate) linear regression
- Multiple linear regression
- Linear basis function regression
- OLS solution for simple regression
- Interpretation
- OLS solution for multiple/LBF regression

Regression

Regression - quick review

The output variable y is continuously valued.

We need a function f to map each input vector $\mathbf{x_i}$ to a prediction,

$$\hat{y}_i = f(\mathbf{x_i})$$

where (we hope!) $\hat{y}_i \approx y_i$.

Prediction by mean

Last week, we imagined a simple model that predicts the mean of target variable in training data:

$$\hat{y}_i = w_0$$

$\forall i$, where $w_0 = \frac{1}{n} \sum_{i=1}^{n} y_i = \bar{y}$.

Prediction by mean, illustration
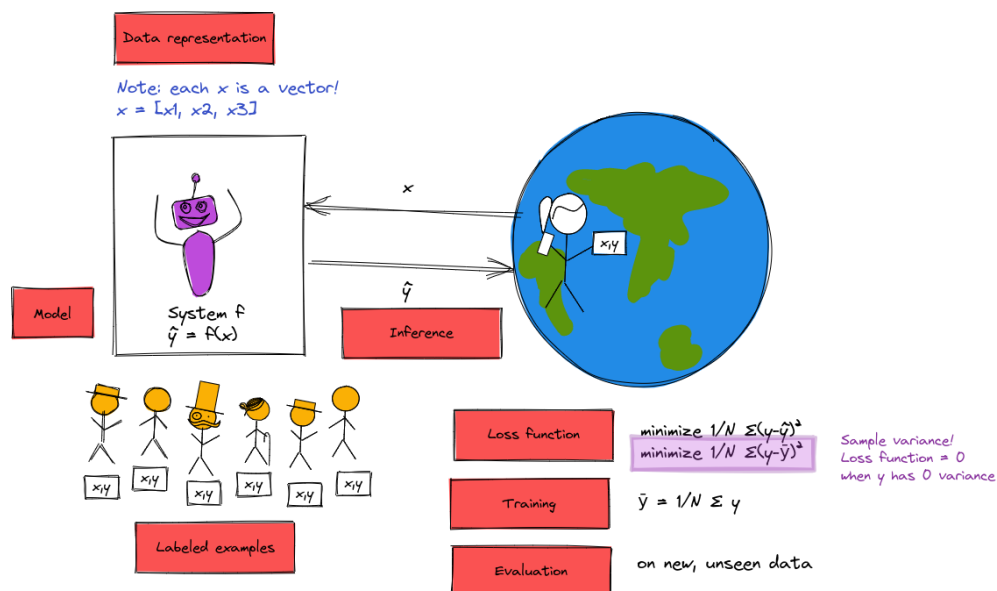
Figure 1: A "recipe" for our simple ML system.

Note that the loss function we defined for this problem - sum of squared differences between the true value and predicted value - is the variance of y.

Under what conditions will that loss function be very small (or even zero)?
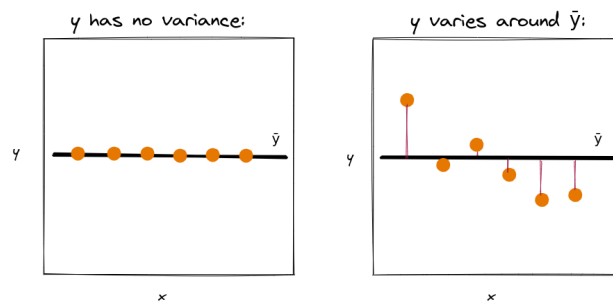
Figure 2: Prediction by mean is a good model if there is no *variance* in y. But, if there *is* variance in y, a good model should *explain* some/all of that variance.

Sample mean, variance - definitions

Sample mean and variance:

$$\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i, \quad \sigma_y^2 = \frac{1}{n} \sum_{i=1}^{n} (y_i - \bar{y})^2$$

Simple linear regression

A "simple" linear regression is a linear regression with only one feature.

Regression with one feature

For simple linear regression, we have feature-label pairs:

$$(x_i, y_i), i = 1, 2, \cdots, n$$

(we'll often drop the index i when it's convenient.)

Simple linear regression model

Assume a linear relationship:

$$\hat{y}_i = w_0 + w_1 x_i$$

where $\mathbf{w} = [w_0, w_1]$, the intercept and slope, are model **parameters** that we *fit* in training.

Residual term (1)

There is variance in y among the data:

- some of it is "explained" by $f(x) = w_0 + w_1 x$
- some of the variance in y is *not* explained by $f(x)$
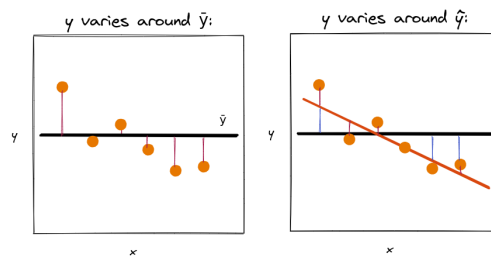
Figure 3: Some (but) not necessarily all of variance of y is explained by the linear model.

Maybe y varies with some other function of x, maybe part of the variance in y is explained by other features not in x, maybe it is truly "random"...

Residual term (2)

The *residual* term captures everything that isn't in the model:

$$y_i = w_0 + w_1 x_i + e_i$$

where $e_i = y_i - \hat{y}_i$.

Example: Intro ML grades (1)
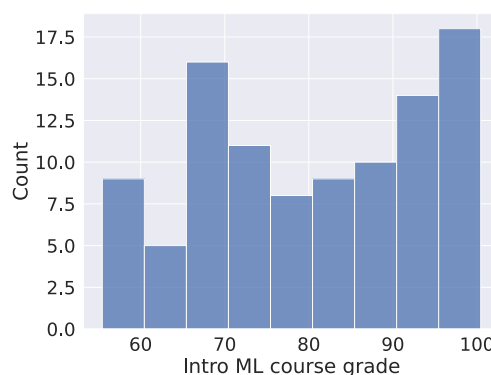
Figure 4: Histogram of previous students' grades in Intro ML.

Note: this is a fictional example with completely invented numbers.

Suppose students in Intro ML have the following distribution of course grades. We want to develop a model that can predict whether you will be a student in the "95-100" bin or a student in the "75-80" bin.

Example: Intro ML grades (2)
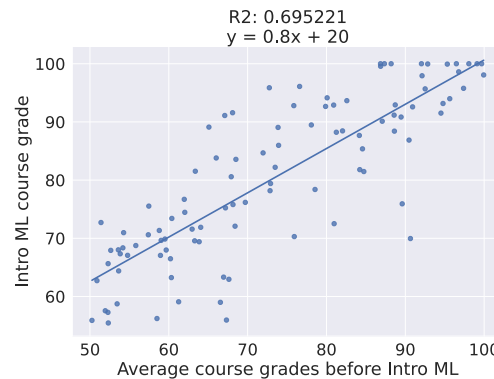
Figure 5: Predicting students' grades in Intro ML using regression on previous coursework.

To some extent, a student's average grades on previous coursework "explains" their grade in Intro ML.

- The predicted value for each student, $\hat{y}$, is along the diagonal line. Draw a vertical line from each student's point (y) to the corresponding point on the line ($\hat{y}$). This is the residual $e = y - \hat{y}$.
- Some students fall right on the line - these are examples that are explained "well" by the model.
- Some students are far from the line. The magnitude of the *residual* is greater for these examples.
- The difference between the "true" value y and the predicted value $\hat{y}$ may be due to all kinds of differences between the "well-explained example" and the "not-well-explained-example" - not everything about Intro ML course grade can be explained by performance in previous coursework! This is what the residual captures.

Interpreting the linear regression: If slope w_1 is 0.8 points in Intro ML per point average in previous coursework, we can say that

- a 1-point increase in score on previous coursework is, on average, associated with a 0.8 point increase in Intro ML course rgade.

What can we say about possible explanations? We can't say much using this method - anything is possible:

- statistical fluke (we haven't done any test for significance)
- causal - students who did well in previous coursework are better prepared
- confounding variable - students who did well in previous coursework might have more time to study because they don't have any other jobs or obligations, and they are likely to do well in Intro ML for the same reason.

This method doesn't tell us *why* this association is observed, only that it is. (There are other methods in statistics for determining whether it is a statistical fluke, or for determining whether it is a causal relationship.)

(Also note that the 0.8 point increase according to the regression model is only an *estimate* of the "true" relationship.)

Multiple linear regression

Matrix representation of data

Represent data as a **matrix**, with n samples and d features; one sample per row and one feature per column:

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,d} \\ \vdots & \ddots & \vdots \\ x_{n,1} & \cdots & x_{n,d} \end{bmatrix}, \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

$x_{i,j}$ is jth feature of ith sample.

Note: by convention, we use capital letter for matrix, bold lowercase letter for vector.

Linear model

For a given sample (row), assume a linear relationship between feature vector $\mathbf{x_i} = [x_{i,1}, \cdots, x_{i,d}]$ and scalar target variable y_i:

$$\hat{y}_i = w_0 + w_1 x_{i,1} + \cdots + w_d x_{i,d}$$

Model has $d + 1$ **parameters**.

- Samples are vector-label pairs: $(\mathbf{x_i}, y_i), i = 1, 2, \cdots, n$
- Each sample has a feature vector $\mathbf{x_i} = [x_i, 1, \cdots, x_i, d]$ and scalar target y_i
- Predicted value for ith sample will be $\hat{y}_i = w_0 + w_1 x_{i,1} + \cdots + w_d x_{i,d}$

It's a little awkward to carry around that w_0 separately, if we roll it in to the rest of the weights we can use a matrix representation...

Matrix representation of linear regression (1)

Define a new **design matrix** and **weight vector**:

$$\mathbf{A} = \begin{bmatrix} 1 & x_{1,1} & \cdots & x_{1,d} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n,1} & \cdots & x_{n,d} \end{bmatrix}, \mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{bmatrix}$$

Matrix representation of linear regression (2)

Then, $\hat{\mathbf{y}} = \mathbf{A}\mathbf{w}$.

And given a new sample with feature vector $\mathbf{x_i}$, predicted value is $\hat{y}_i = [1, \mathbf{x_i}^T]\mathbf{w}$.

Here is an example showing the computation:
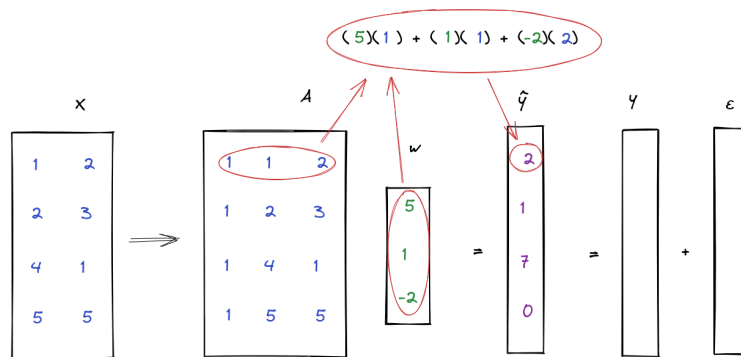
Figure 6: Example of a multiple linear regression.

What does the residual look like in the multivariate case?

Illustration – residual with two features
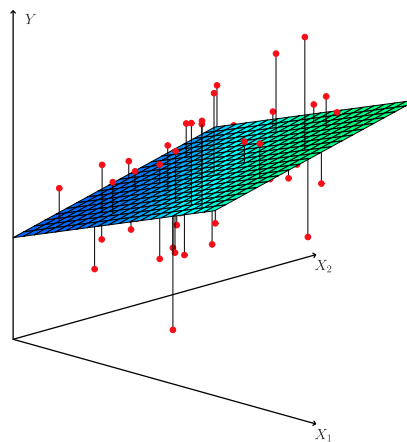
Figure 7: In 2D, the least squares regression is now a plane. In higher d, it's a hyperplane. (ISLR)

Linear basis function regression

The assumption that the output is a linear function of the input features is very restrictive. Instead, what if we consider *linear combinations* of *fixed non-linear* functions?

Basis functions

A function

$$\phi_j(\mathbf{x}) = \phi_j(x_1, \cdots, x_d)$$

is called a **basis function**.

Linear basis function model for regression

Standard linear model:

$$\hat{y}_i = w_0 + w_1 x_{i,1} + \cdots + w_d x_{i,d}$$

Linear basis function model:

$$\hat{y}_i = w_0 \phi_0(\mathbf{x_i}) + \cdots + w_p \phi_p(\mathbf{x_i})$$

Some notes:

- The 1s column we added to the design matrix is easily represented as a basis function ($\phi_0(\mathbf{x}) = 1$).
- There is not necessarily a one-to-one correspondence between the columns of X and the basis functions ($p \neq d$ is OK!). You can have more/fewer basis functions than columns of X.
- Each basis function can accept as input the entire vector $\mathbf{x_i}$.
- The model has $p + 1$ parameters.

Vector form of linear basis function model

The prediction of this model expressed in vector form is:

$$\hat{y}_i = \langle \boldsymbol{\phi}(\mathbf{x_i}), \mathbf{w} \rangle = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x_i})$$

where

$$\boldsymbol{\phi}(\mathbf{x_i}) = [\phi_0(\mathbf{x_i}), \cdots, \phi_p(\mathbf{x_i})], \mathbf{w} = [w_0, \cdots, w_p]$$

(The angle brackets denote a dot product.)

Important note: although the model can be non-linear in $\mathbf{x}$, it is still linear in the parameters $\mathbf{w}$ (note that $\mathbf{w}$ appears *outside* $\phi(\cdot)$!) That's what makes it a *linear model*.

Some basis functions have their own parameters that appear inside the basis function, i.e. we might have a model
$$\hat{y}_i = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x_i}, \boldsymbol{\theta})$$
where $\boldsymbol{\theta}$ are the parameters of the basis function. The model is *non-linear* in those parameters, and they need to be fixed before training.

Matrix form of linear basis function model

Given data $(\mathbf{x_i}, y_i), i = 1, \cdots, n$:

$$\Phi = \begin{bmatrix} \phi_0(\mathbf{x_1}) & \phi_1(\mathbf{x_1}) & \cdots & \phi_p(\mathbf{x_1}) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x_n}) & \phi_1(\mathbf{x_n}) & \cdots & \phi_p(\mathbf{x_n}) \end{bmatrix}$$

and $\hat{\mathbf{y}} = \Phi \mathbf{w}$.

"Recipe" for linear regression (???)

1. Get **data**: $(\mathbf{x_i}, y_i), i = 1, 2, \cdots, n$
2. Choose a **model**: $\hat{y}_i = \langle \phi(\mathbf{x_i}), \mathbf{w} \rangle$
3. Choose a **loss function**: **???**
4. Find model **parameters** that minimize loss: **???**
5. Use model to **predict** $\hat{y}$ for new, unlabeled samples
6. Evaluate model performance on new, unseen data

Now that we have described some more flexible versions of the linear regression model, we will turn to the problem of finding the weight parameters, starting with the simple linear regression. (The simple linear regression solution will highlight some interesting statistical relationships.)

Ordinary least squares solution for simple linear regression

Mean squared error

We will use the *mean squared error* (MSE) loss function:

$$L(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \frac{1}{n} \sum_{i=1}^{n} (e_i)^2$$

a variation on the *residual sum of squares* (RSS):

$$\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} (e_i)^2$$

"Least squares" solution: find values of $\mathbf{w}$ to minimize MSE.

"Recipe" for linear regression

1. Get **data**: $(\mathbf{x_i}, y_i), i = 1, 2, \cdots, n$
2. Choose a **model**: $\hat{y}_i = \langle \phi(\mathbf{x_i}), \mathbf{w} \rangle$
3. Choose a **loss function**: $L(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$
4. Find model **parameters** that minimize loss: $\mathbf{w}^*$
5. Use model to **predict** $\hat{y}$ for new, unlabeled samples
6. Evaluate model performance on new, unseen data

How to find $\mathbf{w}^*$?

The loss function is convex, so to find $\mathbf{w}^*$ where $L(\mathbf{w})$ is minimized, we:

- take the partial derivative of $L(\mathbf{w})$ with respect to each entry of $\mathbf{w}$
- set each partial derivative to zero

Optimizing $\mathbf{w}$ - simple linear regression (1)

Given

$$MSE(w_0, w_1) = \frac{1}{n} \sum_{i=1}^{n} [y_i - (w_0 + w_1 x_i)]^2$$

we take

$$\frac{\partial MSE}{\partial w_0} = 0, \frac{\partial MSE}{\partial w_1} = 0$$

Optimizing $\mathbf{w}$ - simple linear regression (2)

First, the intercept:

$$MSE(w_0, w_1) = \frac{1}{n} \sum_{i=1}^{n} [y_i - (w_0 + w_1 x_i)]^2$$

$$\frac{\partial MSE}{\partial w_0} = -\frac{2}{n} \sum_{i=1}^{n} [y_i - (w_0 + w_1 x_i)]$$

using chain rule, power rule.

(We can then drop the -2 constant factor when we set this expression equal to 0.)

Optimizing $\mathbf{w}$ - simple linear regression (3)

Set this equal to 0, "distribute" the sum, and we can see

$$\frac{1}{n} \sum_{i=1}^{n} [y_i - (w_0 + w_1 x_i)] = 0$$

$$\implies w_0^* = \bar{y} - w_1^* \bar{x}$$

where $\bar{x}, \bar{y}$ are the means of x, y.

Optimizing $\mathbf{w}$ - simple linear regression (4)

Now, the slope coefficient:

$$MSE(w_0, w_1) = \frac{1}{n} \sum_{i=1}^{n} [y_i - (w_0 + w_1 x_i)]^2$$

$$\frac{\partial MSE}{\partial w_1} = \frac{1}{n} \sum_{i=1}^{n} 2(y_i - w_0 - w_1 x_i)(-x_i)$$

Optimizing $\mathbf{w}$ - simple linear regression (5)

$$\implies -\frac{2}{n}\sum_{i=1}^{n} x_i(y_i - w_0 - w_1 x_i) = 0$$

Solve for w_1^*:

$$w_1^* = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

Note: some algebra is omitted here, but refer to the secondary notes for details.

Optimizing $\mathbf{w}$ - simple linear regression (6)

The slope coefficient is the ratio of *sample covariance* σ_{xy} to *sample variance* σ_x^2:

$$\frac{\sigma_{xy}}{\sigma_x^2}$$

where $\sigma_{xy} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$ and $\sigma_x^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$

Optimizing $\mathbf{w}$ - simple linear regression (7)

We can also express it as

$$\frac{r_{xy}\sigma_y}{\sigma_x}$$

where sample correlation coefficient $r_{xy} = \frac{\sigma_{xy}}{\sigma_x \sigma_y}$.

(Note: from Cauchy-Schwartz law, $|\sigma_{xy}| < \sigma_x \sigma_y$, we know $r_{xy} \in [-1, 1]$)

MSE for optimal simple linear regression

$$MSE(w_0^*, w_1^*) = \sigma_y^2 - \frac{\sigma_{xy}^2}{\sigma_x^2}$$

$$\implies \frac{MSE(w_0^*, w_1^*)}{\sigma_y^2} = 1 - \frac{\sigma_{xy}^2}{\sigma_x^2 \sigma_y^2}$$

- the ratio on the left is the *fraction of unexplained variance*: of all the variance in y, how much is still "left" unexplained after our model explains some of it? (best case: 0)
- the ratio on the right is the *coefficient of determination*, R2 (best case: 1).

Regression metrics

Many numbers:

- Correlation coefficient R (r_{xy})
- Slope coefficient w_1 and intercept w_0
- RSS, MSE, R2

Which of these depend only on the data, and which depend on the model too?

Which of these tell us something about the "goodness" of our model?

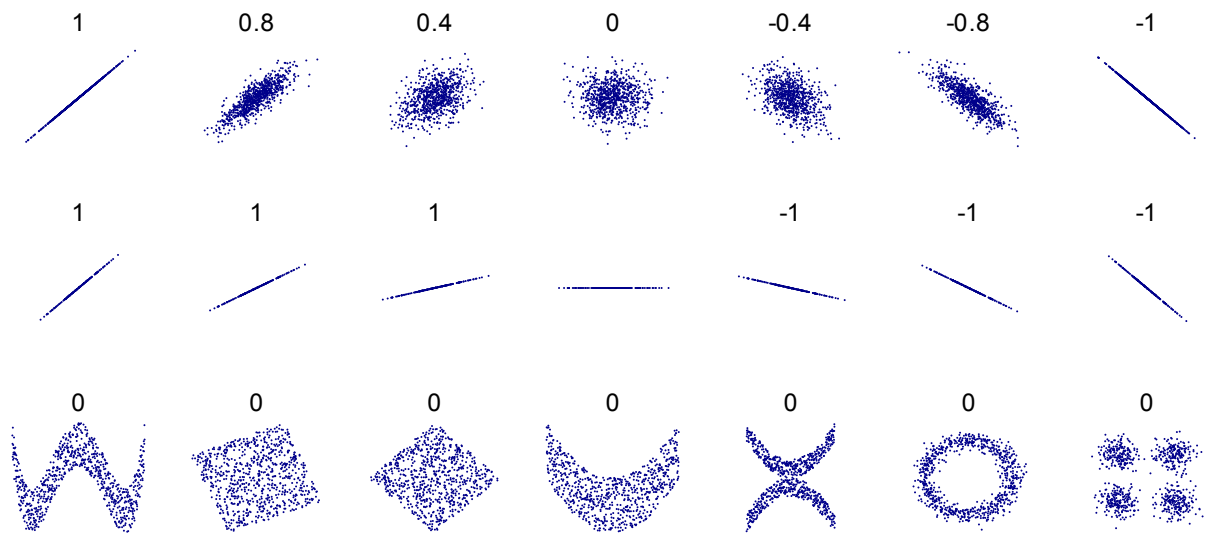## Interpreting R: correlation coefficient

Figure 8: Several sets of (x, y) points, with r_{xy} for each. Image via Wikipedia.

Interpreting R2: coefficient of determination (1)

$$R2 = 1 - \frac{MSE}{\sigma_y^2} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y_i})^2}$$

For linear regression: What proportion of the variance in y is "explained" by our model?

- $R^2 \approx 1$ - model "explains" all the variance in y
- $R^2 \approx 0$ - model doesn't "explain" any of the variance in y

Interpreting R2: coefficient of determination (2)

Alternatively: what is the ratio of error of our model, to error of prediction by mean?

$$R2 = 1 - \frac{MSE}{\sigma_y^2} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y_i})^2}$$
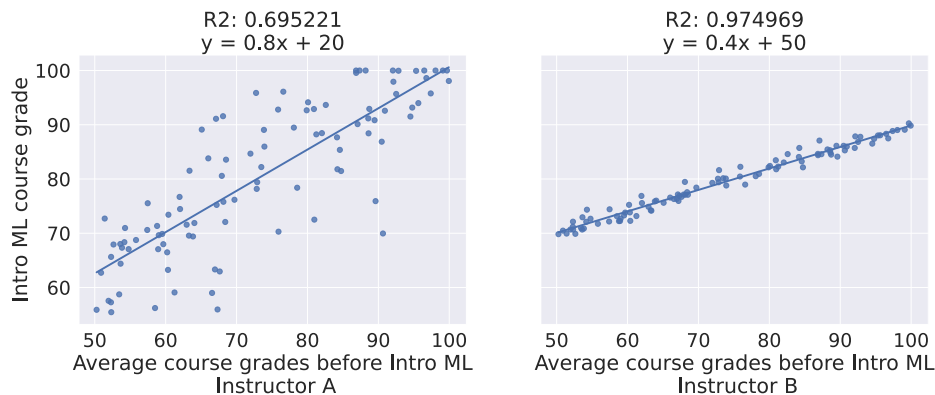
Example: Intro ML grades (3)

Figure 9: Predicting students' grades in Intro ML, for two different sections.

In Instructor A's section, a change in average overall course grades is associated with a bigger change in Intro ML course grade than in Instructor B's section; but in Instructor B's section, more of the variance among students is explained by the linear regression on previous overall grades.

Ordinary least squares solution for multiple/linear basis function regression

Setup: L2 norm

Definition: L2 norm of a vector $\mathbf{x} = (x_1, \cdots, x_n)$:

$$||\mathbf{x}|| = \sqrt{x_1^2 + \cdots + x_n^2}$$

We will want to minimize the L2 norm of the residual.

Setup: Gradient vector

To minimize a multivariate function $f(\mathbf{x}) = f(x_1, \cdots, x_n)$, we find places where the **gradient** is zero, i.e. each entry must be zero:

$$\nabla f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} \\ \vdots \\ \frac{\partial f(\mathbf{x})}{\partial x_n} \end{bmatrix}$$

The gradient is the vector of partial derivatives.

MSE for multiple/LBF regresion

Given a vector $\mathbf{y}$ and matrix Φ (with d columns, n rows),

$$L(\mathbf{w}) = \frac{1}{2}\|\mathbf{y} - \Phi\mathbf{w}\|^2$$

where the norm above is the L2 norm.

(we defined it with a $\frac{1}{2}$ constant factor for convenience.)

Gradient of MSE

$$L(\mathbf{w}) = \frac{1}{2}\|\mathbf{y} - \Phi\mathbf{w}\|^2$$

gives us the gradient

$$\nabla L(\mathbf{w}) = -\Phi^T(\mathbf{y} - \Phi\mathbf{w})$$

Solving for $\mathbf{w}$

$$\begin{aligned}
\nabla L(\mathbf{w}) &= 0, \\
-\Phi^T(\mathbf{y} - \Phi\mathbf{w}) &= 0, \\
\Phi^T\Phi\mathbf{w} &= \Phi^T\mathbf{y}, \quad \text{or} \\
\mathbf{w} &= (\Phi^T\Phi)^{-1}\Phi^T\mathbf{y}.
\end{aligned}$$

Solving a set of linear equations

If $\Phi^T\Phi$ is full rank (usually: if $n \geq d$), then a unique solution is given by

$$\mathbf{w}^* = \left(\Phi^T\Phi\right)^{-1}\Phi^T\mathbf{y}$$

This expression:

$$\Phi^T\Phi\mathbf{w} = \Phi^T\mathbf{y}$$

represents a set of d equations in d unknowns, called the *normal equations*.

We can solve this as we would any set of linear equations (see supplementary notebook on computing regression coefficients by hand.)