ML Week 0x02 Linear Regression

Linear regression: the problem

1. $\boxed{\text{Problem } (\times 6)}$ We have a set of points $\{(x_i, y_i)\}$. Given a new x value, we'd like to predict $\hat{y}$.
2. **Linear model:** We'll assume there exists a linear relationship $y = \theta_0 + \theta_1 x$ that offers a good approximation to the data.
3. In the real world, there's always noise
4. Sometimes other effects, too
5. Talk about meaning of slope
6. Dangers of extrapolation. Example: global warming (a few data points in a few places at a few times)

$\boxed{\text{Residuals } (\times 6)}$

1. *résidu*
2. Goal: small residuals
3. Cost function: sum of squares of residuals
4. Residuals are what's left over after accounting for model fit.
5. A normal distribution of residuals is a good sign. And conversely.
6. Not rules: rule of thumb.
7. Time series (*une série temporelle*) often have important underlying structure. Correlation often doesn't model them well.

$\boxed{\text{Outliers } (\times 8)}$

1. Points that fall farther from the regression line have more effect. We call them *high leverage* points.
2. If the effect is noticeable on the regression, we call it an *influential point*.
3. If a point, omitted, would fall much further from the regression line, it is certainly influential.

4. If not enough data points, they might be all or mostly influential!
5. $\boxed{\text{Anscombe's quartet}}$ — summary statistics don't replace visualizing data
   - mean $x = 9$
   - variance = 11
   - mean $y = 7.50$
   - sample variance $\in (4.122, 4.127)$
   - $\mathbf{Corr}(x, y) = 0.816$
   - linear regression: $y = 3 + x/2$
6. Correlation does not imply $\boxed{\text{causation}}$—but it's a good hint

$\boxed{\text{Code lab: régression linéaire}}$

Linear regression

1. Univariate — 1 input, 1 continuous output
2. We think there's a linear model
3. Explanatory or predictor varaible
4. Response variable
5. $h_\theta(x) = \theta_0 + \theta_1 x$
6. $\boxed{\text{Cost function}}$: $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)^2$
7. Cost function = fonction objective?
8. y vs $\hat{y}$
9. $\boxed{\text{Gradient descent } (\times 3)}$ (algorithme du gradient)
10. Assignment is simultaneous
11. Outlier = *donnée aberrante*

Linear algebra (review)

1. Vector, $\boxed{\text{matrix}}$, transpose
2. addition, multiplication

3. vector space, basis vectors
4. linear transformation, $u = Av$, think about basis vectors
5. A, $A_{i,j}$

$\boxed{\text{Notation used in machine learning}}$

1. $x_j^{(i)}$ — value of feature j in training sample i
2. $x^{(i)}$ — training sample i
3. m = number of training samples
4. $n =| x^{(i)} |$ = number of features
5. $x_0 = 1$ (often called bias)

Multiple regression

1. Multiple explanatory variables, 1 continuous output
2. Fortunately, there are libraries to do this!

$\boxed{\text{Codelab}}$: We already did it! Diabetes example.

Other notes

- Overfitting
- Regularization (ridge regression, Tikhanov regularization): $-\lambda \sum$ params
- Polynomial regression
- Gradient descent variants
  - Batch gradient descent (all samples)
  - Stochastic gradient descent (single sample each iteration) (faster for very large sets)
  - Coordinate gradient descent (one component each iteration)
  - Note computational approximation if no derivative (and curse of dimensionality)
- When gradient descent doesn't work,
  - plot the cost function over iterations
  - if cost increasing or oscillating, reduce α
  - if leveled off, not much future gain