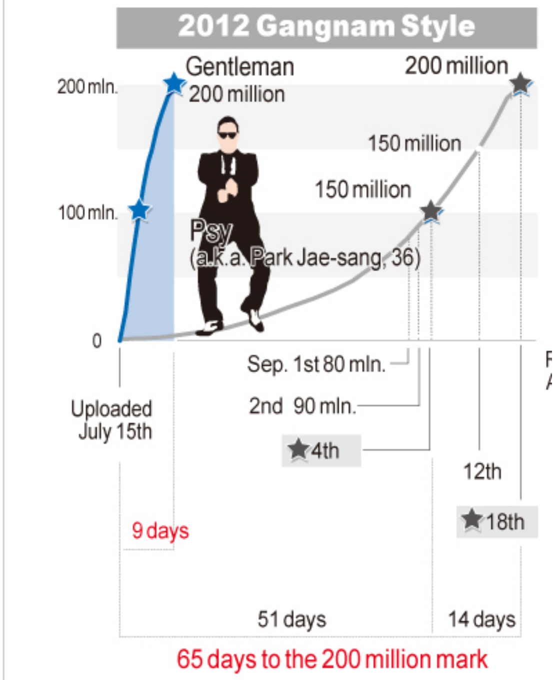
Assignment #5

This assignment is due on April 23rd one hour before class via email to mailto:wallraven@korea.ac.kr. Please observe the directory structure in this zip-file: solutions for part1 should go into the part1 directory, etc.

If you are done with the assignment, make one zip-file of the assignment5 directory and call this <LASTNAME_FIRSTNAME_A5.zip> (e.g.: HONG_GILDONG_A5.zip).

Please make sure to comment the code, so that I can understand what it does. Uncommented code will reduce your points!

Part1 Finding zeros (30 points):



Here is a picture showing the views for Gentleman versus Gangnam Style. Use your ruler-skills or any other method to derive several data points on each curve. Make sure to measure the x-value (in days) and the y-value (in million views). **Try to measure at least 7-8 points for each curve.**

a) Similarly to the censusgui example from class (the code of which you can find in the NCM book!!), fit your data for Gentleman and Gangnam Style with polynomials of order 1 to n (where n is the number of data points you measured).

Given **a good model** for both curves, how many days after upload will Gentleman or Gangnam Style hit 300 mln viewers, how many days for 500 mln, and how many days for 1000 mln?

Make sure to insert all observations and interpretations as comments into your script! Please make sure to justify your choice of model very well!!!

Part2 Curve fitting 1 (15 points):

erf is the so-called Error Function, which is defined as

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt.$$

The function is thus a simple integral of a Gaussian. This integral, however, cannot be described as a simple, closed function, but can be represented as a series of terms. This function has important applications in statistics, and many other fields. Here, we will play a little around with this function using the Matlab built-in function `erf`.

In a script called `errorerf.m`, generate 11 datapoints, $t_k = (k-1)/10$, $y_k = \operatorname{erf}(t_k)$, $k=1,\ldots,11$.

a) Fit the data in a least-squares sense with polynomials of degrees 1 through 10. (Hint: You want to create a matrix of coefficients with the Vandermonde matrix as shown in class and solve those with the backslash command – **please do NOT use polyfit!!!**) Compare the fitted polynomial with erf(t) for values of t between the data points. How does the maximum error (in a least squares sense) depend on the polynomial degree? For this, plot the error you make in the fit in a figure.

b) Because erf(t) is an odd function of t, that is, erf(x) = −erf(−x), it is reasonable to fit the data by a linear combination of odd powers of t:

$\operatorname{erf}(t) \approx c_1 t + c_2 t^3 + \cdots + c_n t^{2n-1}$.

Again, see how the error between data points depends on n, that is, plot these errors into the plot as well. Save the whole plot as `errorerf.png` in the `part1` directory.

Insert all observations as comments into `errorerf.m`

Part3 Curve fitting 2 (15 points):

Here are 25 observations, y_k, taken at equally spaced values of t.

t = 1:25;

y = [5.0291 6.5099 5.3666 4.1272 4.2948 6.1261 12.5140 10.0502 9.1614 7.5677 7.2920 10.0357 11.0708 13.4045 12.8415 11.9666 11.0765 11.7774 14.5701 17.0440 17.0398 15.9069 15.4850 15.5112 17.6572];

Write a script called `datafit.m` that creates these datapoints.

a) Fit the data with a straight line, $y(t) = \beta_1 + \beta_2 t$ (**again, do NOT use polyfit!!!**), and plot the residuals, $y(t_k) - y_k$. You should observe that one of the data points has a much larger residual than the others. This is probably an outlier.

b) Discard the outlier, and fit the data again by a straight line. Plot the residuals again. Do you see any pattern in the residuals? What kind of pattern could be missing?

c) Fit the data, with the outlier excluded, by a model of the form $y(t) = \beta_1 + \beta_2 t + \beta_3 f(t)$, where $f(t)$ is the pattern you found in b).

d) Evaluate the fit from c) on a finer grid over the interval [0,26]. Plot the fitted curve, using line style '-', together with the original data, using line style 'o'. Include the outlier, using a different marker, '*'.

Be sure to include all commands and observations into `datafit.m`