Announcements

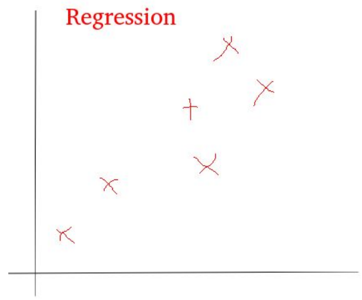
- Next week:
  - Tuesday self study
  - Wednesday tutorials
- Week after that:
  - Tuesday lecture via hangout
  - Wednesday tutorials
- There are homeworks on the webpage
  - Warning: template code is in python2, if this thing is a problem let me know

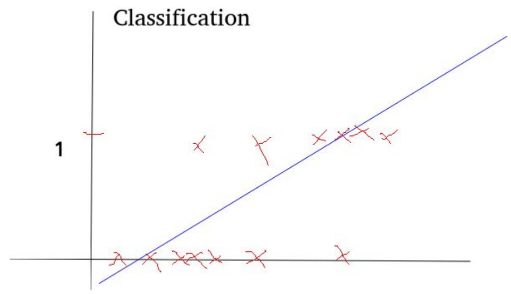
Regularization

- We have seen:
  - Pick the right polynomial for regression (x vs x^2 vs x^3)
  - Via holdout testing or k-fold cross validation
- Another idea:
  - Penalize huge weights
  - Instead $min_\theta \sum_i L(f(x_i), y_i)$, i.e. in regression $min_\theta \sum_i (x_i.\theta - y_i)^2$
  - Do $min_\theta \sum_i L(f(x_i), y_i) + C \sum_i \theta_i^2$ - L2 regularization or Ridge regression
  - Or $min_\theta \sum_i L(f(x_i), y_i) + C \sum_i |\theta_i|$ - L1 regularization or Lasso regression
  - L1 regularization leads to more zero weights -> sparse models
    - Useful when looking for relevant attributes
  - Ridge regression can be analytically solved
  - Lasso penalty is tricky to implement rather use some package (e.g. scikit-learn)
- How to choose C
  - Via holdout testing or k-fold cross validation

Binary classification

- 0/1 classification, e.g. spam / non spam, click / not click, safe / unsafe content, …
- Conversion to regression is not ideal





Prediction outside 0-1 range.
Penalty?
target 1 prediction 0.9 -> ok
target 1 prediction 0.1 -> bad
target 1 prediction -42 -> very bad
target 1 prediction 100 -> actually good, but very bad under quadratic penalty

- Let's change predictions. Force it into 0-1 range. Interpretation of prediction:
    - Probability of target being 1 (e.g. probability of spam)
    - Do regression and then process result z via sigmoid: $\sigma(z) = \frac{1}{1+e^{-z}}$
        - I.e our output for two attributes is:
            - $\sigma(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$
- How to fit.
    - Maximalize probability of data
        - If data point has target 1, I want to maximize probability
        - If data point had target 0, I want to minimize the probability
        - p_i - my prediction; $p_i^{y_i}(1 - p_i)^{1-y_i}$
    - Goal to optimize is product of all datapoints
        - $\prod_i p_i^{y_i}(1 - p_i)^{1-y_i}$
        - Where $p_i = \sigma(\theta.x)$
    - In practice we want sum (easier to differentiate). And also minimize something (just to be consistent with other stuff). Just logarithm and negate and goal would be to minimize
        - $-\sum_i \log(p_i^{y_i}(1 - p_i)^{1-y_i}) = -\sum_i y_i \log p_i + (1-y_i)\log(1-p_i)$
        - This is also called cross-entropy.
- More math:
    - Model: $P[y_i|x_i] = \sigma(\theta.x)$

    - We optimize likelihood of *y*-s
- What if we use L2 penalty $(p_i - y_i)^2$ instead of log? Think about gradient.
- Checkout how to calculate gradient for parameters.
    - $\frac{\partial J}{\partial \theta_j} = (y_i - p_i)x_j$
- This is also called Logistic regression

Softmax classification

- Generalization for multiple target categories (e.g. predict what is in the picture dog/cat/plane/house/…)
- Categories are fixed beforehand
- Predict probability for each category
    - E.g. $P[y_i = dog|x_i]$
- One parameter for each input-output combination (before only one parameter for each input).
- We should process outputs:
    - Each output is in 0-1 range
    - Sum of outputs should be one
- Via something called softmax:
    - $\sigma(z_1,\ z_2,\ ...,\ z_j)_i = \frac{e^{z_i}}{\sum_k e^{z_k}}$
- Model:
    - Input: x $(x_1,\ x_{2,\ ...,} x_m)$
    - k categories
    - Parameters θ_{ij} - matrix of size m x k: Θ
    - Intermediate output
        - $x\Theta$, j-th element: $z_j = \sum_i x_i \theta_{ij}$
    - Output probabilities
        - $p\ = \sigma(z)$
- Loss:
    - Negative log-likelihood of the data:
        - $-\sum_i \log(p_{iy_i})$
- Also called maximum-entropy classification

Probabilistic interpretation of regression

- Let's say that our output is $\theta_0 + \theta_1 x_1 + \theta_2 x_2\ +\ noise$
- What if noise has a normal distribution with mean 0 and variance V
- Or in other words: y is from normal distribution with mean $\theta_0 + \theta_1 x_1 + \theta_2 x_2$ and variance V

- Let's maximize the probability of the data (intentionally ignoring some terms to simplify the presentation):
  - $\prod_i e^{-(y_i - \theta_0 + \theta_1 x_1 + \theta_2 x_2)^2}$
  - Now do the log and negation (to get sum and minimization)
  - $-\sum_i -(y_i - \theta_0 + \theta_1 x_1 + \theta_2 x_2)^2 = \sum_i (y_i - \theta_0 + \theta_1 x_1 + \theta_2 x_2)^2$
  - Which is linear regression formulation!